Model Theory and AI: Results in Query Learning of Automata and Weighted Model Counting

by

Kevin Huan Zhou
B.S., Carnegie Mellon University, 2018

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mathematics
in the Graduate College of the
University of Illinois at Chicago, 2024

Chicago, Illinois

Defense Committee:
James Freitag, Chair and Advisor
John Baldwin
Joel (Ronnie) Nagloo
György Turán
Caroline Terry, The Ohio State University

Soli Deo gloria

ACKNOWLEDGMENT

First and foremost, I would like to thank my advisor, James Freitag, for his support and guidance throughout my years as a graduate student. The ideas for the projects contained in this thesis originated from him, and he has also provided me with valuable advice and guidance on many non-mathematical aspects of academic life, such as suggestions for conferences and workshops to attend and tips for the paper submission process. I also want to thank him for providing travel support on several occasions and for his recommendations to be supported by the NSF RTG in Algebraic and Arithmetic Geometry at UIC as well as the NSF TRIPODS Institute at UIC.

Thanks are also due to the many people with whom I've had the chance to discuss my work and related topics, whether at length or just in passing, including but not limited to John Baldwin, Artem Chernikov, Matthew Harrison-Trainor, Ronnie Nagloo, Lev Reyzin, Caroline Terry, Gyoŕgy Turán, and Guy van den Broeck. In particular, special thanks goes to John Baldwin, Ronnie Nagloo, Caroline Terry, and György Turán for serving on my thesis defense committee.

Math is best done in the company of others, and I must thank many of my fellow graduate students for their friendship throughout the years. Of particular importance is the wonderful community of logic graduate students at UIC and around the world—thank you for being kind, welcoming, collaborative, and a lot of fun to crack logic jokes with.

ACKNOWLEDGMENT (Continued)

Special thanks is due to my community at Church of the Beloved. I would not have been able to make it through these six years without their constant fellowship and prayers. Of chief note are Pastor Abe and Suzette Lee, for being steadfast sources of leadership and stability through many uncertain times, and Eugenia Kang, for keeping the church running through so many ups and downs and for providing heaps of support and guidance during my time on staff.

An acknowledgements section would not be complete without thanks to family. To Mom, Dad, and Jason, thank you for your constant and unwavering love and support in all areas of life.

KHZ

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

SUMMARY

In this thesis, we study two connections between model theory and AI. These two areas may at first seem fairly unrelated—model theory generally concerns itself with studying infinite (and often uncountable) structures, which tends to lie outside the realm of tractable computation. On the other hand, AI is highly concerned with tractable and efficient computation, especially in the now popular area of machine learning. However, a major recent trend of understanding *dividing lines* in model theory have led to many fruitful applications outside of model theory.

The first connection we study is in computational learning theory, specifically in *query learning* of various forms of automata, which is the content of Chapter III. Query learning is a setting of machine learning in which a learner interactively submits queries to an oracle, and uses the answers to those queries to attempt to identify some unknown target function. The query learning setting is particulary well suited for learning automata, since a common appliation is to understand the behavior of a real-world black-box system that is modeled by some automaton. Queries can be simulated by interacting with the system and observing the outputs. The study of query learning of automata was initated by Angluin in 1987 with the introduction of the L^* algorithm that learns DFAs (2), which has since been extended to many other generalizations of DFAs. More recently, Chase and Freitag introduced an alternative general-purpose method for proving bounds on the number of queries needed to learn via various combinatorial measures of complexity, and apply this method to the setting of DFAs to obtain qualitatively different results compared to Angluin's algorithm (19). Our work applies this method to two generalizations

of DFAs, *advice DFAs* and *nominal DFAs*. We give the first known query learning bounds for advice DFAs, and give qualitatively different results for nominal DFAs compared to previous results.

The second connection we study is in *weighted model counting*, which is the content of Chapter IV. In particular, we study the weighted first-order model counting problem, which is to determine the weighted sum of all models of a given first-order sentence or theory in a finite relational language over a fixed domain. Weighted first-order model counting has applications in statistical relational learning and probabilistic databases, and can also be seen as a generalization of the SAT problem from computational complexity theory. Prior work has mainly focused on developing computational complexity guarantees on calculating the weighted model count of various restricted classes of sentences. A related problem that has been extensively studied in combinatorics is that of counting the number of structures in *hereditary properties* of various combinatorial objects. A hereditary property is a class of combinatorial objects which is defined by a universal sentence or theory. The main goal in this line of research is to classify the possible asymptotic growth rates of hereditary properties, and Laskowski and Terry prove a classification into four possible asymptotic growth rates for hereditary properties of $\mathcal{L}$-structures, where $\mathcal{L}$ is a finite relational language (36). Our work focuses on understanding how these two distinct approaches to similar problems interact with each other.

CHAPTER I

INTRODUCTION

Mathematical logic and artificial intelligence have a long history of interplay dating back to the earliest efforts to formalize computation by Church and Turing. In this thesis, we investigate two areas in which modern model theory interacts with AI. The first interaction is in computational learning theory, specifically in query learning of automata. We study this interaction in Chapter III. It is adapted from a paper that will appear in *ATVA* 2024. The second interaction is in weighted model counting, a problem with applications in statistical relational learning and probabilistic databases. We study this interaction in Chapter IV. It is based on work for a paper currently in preparation.

I.1 Query learning of automata

Learning various forms of automata with queries is a long-studied field with many applications, including in automatic verification and model checking. The basic question in the area is the following: given some black-box system that is modeled by an automaton, can we deduce the underlying model simply by interacting with the system and observing the input-output behavior? These interactions are formalized using *queries* that are posed to an oracle; two common queries that are considered are *equivalence* and *membership* queries. In essence, an equivalence query is a hypothesis which the learner tests against the system being learned, and if the hypothesis is incorrect, the learner observes a counterexample (i.e. an input for which the

hypothesis and the actual system give differing outputs). On the other hand, a membership query is an input that is fed to the system for which the output is observed. The amount of interactivity available when attempting to learn real-world systems makes query learning a natural setting for learning automata.

This field was initiated by Angluin in 1987 with the introduction of the L^* algorithm that learns deterministic finite automata (DFAs) using a polynomially bounded number of equivalence and membership queries (2). Historically, the study of query learning of automata has centered around adapting Angluin's L^* algorithm to different settings, such as such as tree automata (51), nondeterministic finite automata (14), ω-automata (4), symbolic automata (23), and fully-ordered lattice automata (25). More recently, Chase and Freitag introduced an alternative general-purpose method for proving bounds on the number of queries needed to learn. Their approach involves computing the *Littlestone dimension* and *consistency dimension* of the concept classes in question. These notions of dimension are combinatorial complexity measures studied in computational learning theory, but are also closely related to important notions in model theory. We include a discussion of these connections in Section I.3.

Chase and Freitag's method of obtaining query learning bounds via Littlestone and consistency dimension is especially effective for automata since many types of automata exhibit a version of the Myhill-Nerode theorem, a characterization of regular languages by a syntatic property of the language. The conditions imposed by the Myhill-Nerode characterization turn out to be useful in finding a bound for the consistency dimension. For example, Chase and Freitag apply this method to regular languages and regular ω-languages and obtain qualitatively

different results from prior work. Our work applies this method to two generalizations of DFAs: advice DFAs and nominal DFAs.

Advice DFAs were studied as early as 1968 by Salomaa (52), though we follow the notation of Kruckman et al. (32). Advice DFAs generalize classical DFAs by allowing the automata access to an additional advice string that it reads concurrently with the input. This makes them useful in modeling situations where the transition behavior can vary over time. They also have connections to logic: it is a classical result that DFAs correspond to weak monadic second-order formulas over the structure of natural numbers with the successor operation; advice DFAs correspond to formulas over expansions of this structure by unary predicates, a frequently studied setting; see e.g. (11; 17; 24; 48). Another motivating factor for advice DFAs comes from the study of *automatic structures*, which are structures whose domain and atomic relations are recognized by DFAs. It turns out some natural structures are not automatic or even isomorphic to an automatic structure, such as $(\mathbb{Q}, +)$, the additive group of the rationals (54). However, $(\mathbb{Q}, +)$ *is* isomorphic to a structure whose domain and atomic relations are recognized by advice DFAs (32; 45). For advice DFAs, we give the first known bounds on query complexity.

Nominal DFAs, introduced by Bojańczyk, Klin, and Lasota (13), are a generalization DFAs to infinite alphabets. Such a generalization can be useful when there are infinitely many options for data values, such as in XML documents (where arbitrary strings can appear as attribute values) or in software verification (in order to deal with pointers or arbitrary function parameters). Generalizing automata theory to infinite alphabets is not as simple as replacing instances

of "finite" with "infinite", since without any restrictions, the fact that there are uncountably many subsets of any infinite set makes computation intractable. However, in most reasonable applications, there is additional structure that can be leveraged to make computation reasonable. Nominal DFAs utilize the notion of *nominal sets* (first introduced by Gabbay & Pitts (26)) to formalize this idea. Nominal sets use group actions to capture the idea that in most cases, data values can be compared with each other, such as for equality or for the ordering in an underlying total order on the data values. A further discussion of this intution can be found in Subsection III.3.1. Aside from the development of the theory of automata over infinite alphabets, nominal sets have also found much use in the concurrency and semantics communities as a formalism for modeling name binding (see e.g. (43; 46)). For nominal DFAs, we give qualitatively different results from prior work. A more detailed discussion of our results can be found in Section III.1.

I.2 Weighted model counting

The Boolean satisfiability problem, often abbreviated SAT, is a classic problem in computer science of determining whether or not there exists an assignment of Boolean variables that satisfies a given Boolean formula. It is a canonical NP-complete problem with wide-ranging applications. The counting version of this problem, #SAT, asks how many distinct assignments satisfy the formula, and is the starting point for various model counting problems. Generalizations of the problem can be found by changing the underlying logic from propositional logic, such as to first-order logic, or by moving to the weighted setting, in which an assignment or structure is given a weight and the goal is to find the weighted sum of all satisfying assign-

ments/models instead of just the count. Our work is focused on the *weighted first-order model counting* problem, i.e., computing the weighted sum of all models of some first-order theory over a fixed domain.

Weighted model counting is closely related to problems in *statistical relational learning* (27; 49), in which one aims to model and learn probabilistic relationships between objects which are generally not identically and independently distributed (a common assumption in statistical learning), but rather have rich interconnected relational structure. Such problems crop in real-world scenarios where large knowledge bases contain millions or billions of rows of uncertain relational data. As an example, Google's Knowledge Vault (21) contains triples in the form of `(subject, predicate, object)`, such as `(Barack Obama, place of birth, Honolulu)`. Associated to each triple is a confidence score which represents how likely Knowledge Vault believes the statement to be true. One may then desire to carry out *probabilistic inference*, that is, to determine the probability of some statement being true given the information in the knowledge base.

A standard formalization of this problem is via *Markov logic networks* (MLNs) (50). An MLN consists a finite collection of pairs (w_i, φ_i), where $w_i \in [0, \infty]$ and each φ_i is a first-order formula. Each pair is called a *constraint*, and is meant to represent the fact that the formula φ_i is true with probability proportional to the weight w_i. If $w_i = \infty$, then (w_i, φ_i) is called a *hard*

constraint, representing that φ_i must hold. For example, consider the Markov logic network consisting of the following two constraints:

$$2.0 \qquad (\texttt{smokes}(x) \wedge \texttt{influences}(x, y)) \rightarrow \texttt{smokes}(y)$$

$$0.5 \qquad \texttt{stress}(x) \rightarrow \texttt{smokes}(x)$$

which encodes that people who are influenced by smokers are more likely smoke and that people who are stressed are also more likely to smoke, while the effect of being influenced by a smoker is greater than the effect of being stressed. Given a fixed domain, every first-order structure over the domain is then assigned weight $\exp(\sum_i w_i n_i)$, where i ranges over all finite-weight constraints and n_i is the number of tuples in the structure that satisfy constraint φ_i. Structures that fail to satisfy a hard contstraint are assigned weight 0. This induces a probability distribution over all structures, where each structure is given probability proportional to its weight. The task of probabilistic inference in this case is to determine the probability that a given first-order sentence. For example, one could ask what the probability that the statement $\exists x(\texttt{smokes}(x) \wedge \forall y(\texttt{stress}(y) \rightarrow \neg\texttt{influences}(y, x)))$ holds, which expresses the likelihood that there is a smoker who is not influenced by anyone who is stressed. Following (58), this problem can be recast as a weighted first-order model counting problem in the following way. Suppose we have an MLN $\{(w_i, \varphi_i(\overline{x}_i)) \mid i \in I\} \cup \{(\infty, \varphi_j(\overline{x}_j)) \mid j \in J\}$, where w_i is finite for all $i \in I$. For each $i \in I$, introduce a new relation symbol $R_i(\overline{x}_i)$. Let T be the theory $\{\forall \overline{x}_i \; R_i(\overline{x}_i) \leftrightarrow \varphi_i(\overline{x}_i) \mid i \in I\} \cup \{\forall \overline{x}_j \; \varphi_j(\overline{x}_j) \mid j \in J\}$. Assign $w(R_i) = e^{w_i}$ for each $i \in I$, and

$w(P) = 1$ for all other relation symbols P. Then determining $\Pr(\psi)$ according to the MLN over the domain $[n]$ is equivalent to computing $\frac{\mathsf{WFOMC}(T \cup \{\psi\}, n, w)}{\mathsf{WFOMC}(T, n, w)}$.

Since the desire is to utilize weighted first-order model counting for computational tasks, a major goal is to prove computational complexity results for WFOMC. In general, the weighted first-order model counting problem is computationally hard: in particular, there is a sentence whose weighted model counting problem is known to be $\#P_1$-hard[1] (12, Theorem 3.1). On the other hand, for certain nice fragments of first-order logic, the weighted model counting problem can be computed in polynomial time, such as for sentences involving only two logical variables (57; 58; 12). This was later extended to allow for counting quantifiers (33) and a linear order axiom (55).

Independently of the work on weighted model counting, the unweighted version of the first-order model counting problem has been extensively studied in the combinatorics literature. In particular, much work has focused on understanding the asymptotic growth rates of *hereditary properties* of various combinatorial objects, which are classes of objects that are defined by a universal theory. A typical result proves a "jump" in the possible asymptotic growth rates, that is, showing that the number of objects of size n must either be at most $f(n)$ or at least $g(n)$, where $f(n)$ has strictly slower asymptotic growth rate than $g(n)$. The problem was first studied for graphs by Scheinerman and Zito (53), eventually culminating in a series of papers by Balogh, Bollobás, and Weinrich which together give a classification of the possible

---

[1]the complexity class $\#P_1$ is the class of functions that count the number of accepting computations for a nondeterministic Turing machine with a unary input alphabet (56)

growth rates into four discrete classes (8; 9; 10). Other settings that have been studied include hypergraphs (22), tournaments (7), and posets (6). Generalizing all of the previously listed examples, Laskowski and Terry give a complete classification of the jumps in growth rates of hereditary properties of $\mathcal{L}$-structures for any finite relational language $\mathcal{L}$ (36).

Since the results on hereditary properties often involve strong structural characterizations of classes falling into the various growth rates, one may hope that this can shed some light on the computational aspects of the weighted model counting problem. In the other direction, restricting to the fragments of first-order logic studied in weighted model counting may yield stronger classification results for unweighted model counting. One immediate obstacle to interactions between these two areas is the need to restrict to *universal* theories in the unweighted case. The usual Skolemization procedure is not suitable for model counting problems, since it adds function symbols to the language. However, an advantage of the weighted setting is the existence of an efficient Skolemization procedure that preserves the weighted model count without adding any function symbols (58). With this procedure in hand, studying any weighted model counting problem is equivalent to studying the weighted model counting problem for a universal sentence, at which point one may hope to utilize the classification results proven for the unweighted counting problem to provide further insight into the weighted case. Understanding this connection is the main motivation for our work.

I.3 The model theoretic perspective

At first glance, the relationship between first-order model theory and the subfields of AI that we study may not be so obvious. Model theory generally concerns itself with studying

infinite (and often uncountable) structures, which tends to lie outside the realm of tractable computation, an important requirement for applications in AI. However, a major recent trend of understanding *dividing lines* in model theory has led to many fruiful applications outside of model theory.

At its core, first-order model theory is about developing a theory of *definable sets*, i.e., subsets of a structure which are defined by a first-order formula. This parallels a wide range of other areas in math and computer science. For example, a central motivating theme in algebraic geometry is to develop a theory of varieties, which in the classical sense are sets definable in algebraically closed fields by polynomial equations. In a very different area, the main focus of computational complexity theory is to develop the theory of complexity classes, which (for decision problems) are subsets of finite strings defined by the computational resources needed to compute them.

One of the first roadblocks to model theory is that a general theory of definable sets is, in a formal sense, impossible, such as incompleteness phenomena (e.g., Gödel's incompleteness theorems) or undecidability phenomena (e.g., the MRDP theorem). One one might tackle these roadblocks by studying the degrees of incompleteness or undecidability, which (roughly speaking) are the routes taken by modern set theory and computability theory. On the other hand, since many objects studied in mathematics are known to be well-behaved in particular ways, model theory takes the approach of imposing strong tameness assumptions and studying the consequences of such assumptions. This has led to the development of dividing lines which

separate theories into distinct classes for which the structure theory can be established to varying degrees.

One strength of this approach is that many of the dividing lines studied in model theory have equivalent combinatorial characterizations, which opens up interactions with other areas of math and computer science that utilizing the same or similar conditions. The earliest instance of this interaction was when Laskowski observed that a formula being NIP corresponds to its corresponding class of definable sets having finite VC-dimension (34). VC-dimension is a combinatorial notion of complexity that found much use in the computational learning theory literature as a characterization of concept classes that are probably approximately correct (PAC) learnable. Outside of model theory and computational learning theory, VC-dimension has proven to be tremendously useful in combinatorics, such as giving improvements to Szemerédi's celebrated regularity lemma for graphs of bounded VC-dimension (38) and a proof of the Erdös-Hajnal conjecture for graphs with bounded VC-dimension (44).

More recently, Chase and Freitag noticed that model-theoretic stability corresponds with finite Littlestone dimension (18). Littlestone dimension is another combinatorial notion of complexity that was originally introduced by Littlestone as a way to characterize online learnability (37). On the other hand, stability is the most studied tameness assumption in model theory, with decades of work going into developing a structure theory for stable theories. Stability also has applications in combinatorics, with Malliaris and Shelah proving another strengthening of the Szemerédi regularity lemma for stable graphs (41). Since Chase and Freitag's observation, a flurry of interplay between model theory and computational learning theory in the setting

of finite Littlestone dimension has occured, leading to results such as the equivalence of online learnability and private PAC learnability (1) and several other equivalent characterizations of online learning (40).

In his PhD thesis, Chase also notes that the model-theoretic notion of nfcp (not the finite cover property) corresponds to finite strong consistency dimension (20), which is a strengthening of the consistency dimension which we use to derive our bounds on query learning of various forms of automata. Strong consistency dimension is also known as the *dual Helly number*, a notion arising from discrete geometry which has also found other applications in learning theory (e.g. (16)).

In the area of model counting, early work in studying the asymptotic growth rates of hereditary properties relied on mainly combinatorial methods. However, Laskowski and Terry utilized several model-theoretic ideas and techniques to prove their general result, such as the notion of *mutual algebraicity*, which was first introduced in a purely model-theoretic context by Laskowski (35). The problem of finding finitely many distinct jumps in the growth rates of hereditary properties also mirrors one of the original goals of model-theoretic classification theory, which was to understand the number of non-isomorphic models of a complete theory in various (infinite) cardinalities.

CHAPTER II

PRELIMINARIES

II.1 Basic Notation

Given a natural number $\ell \in \mathbb{N}$ and a set X, let $[\ell] := \{1, \dots, \ell\}$,

$$X^{(\ell)} := \{(x_1, \dots, x_\ell) \in X^\ell \mid x_i \neq x_j \text{ for all } i \neq j\}, \qquad \text{and} \qquad \binom{X}{\ell} := \{Y \subseteq X \mid |Y| = \ell\}$$

We will frequently use overlined letters to denote tuples of elements or variables, and bolded letters to denote tuples of natural numbers. If $\overline{x}$ is a tuple of variables (or elements), then x_i will denote the i-th element of the tuple, and similarly k_i will denote the i-th element of $\boldsymbol{k} \in \mathbb{N}^\ell$. Additionally, given a vector $\boldsymbol{k} \in \mathbb{N}^\ell$ of natural numbers, a *$\boldsymbol{k}$-partition of X* is an ordered partition of X into sets $X_1, \dots, X_\ell$ such that $|X_i| = k_i$ for each $1 \leq i \leq \ell$.

Given a set of symbols A, let A^* denote the set of all finite strings over A. Furthermore, let A^ω denote the set of all length-ω strings over A (equivalently, the set of all functions $f : \mathbb{N} \to A$).

II.2 First-order logic

In this section, we set some notation for various concepts in first-order logic, including some non-standardized terminology. We assume familiarity with the basics of first-order model theory, including languages, structures, satisfaction, theories, and types.

Let $\mathcal{L}$ be a language. We will use calligraphic letters (e.g., $\mathcal{M}, \mathcal{N}$) for $\mathcal{L}$-structures, and plain letters (e.g., M, N) to denote the domain of the corresponding structure. Given an $\mathcal{L}$-formula $\varphi(\overline{x})$, let $\varphi(\overline{x})^1$ denote $\varphi(\overline{x})$ and $\varphi(\overline{x})^0$ denote $\neg\varphi(\overline{x})$. Given an $\mathcal{L}$-structure $\mathcal{M}$ and a formula $\varphi(x_1, \ldots, x_s)$, define

$$\varphi(\mathcal{M}) := \{(a_1, \ldots, a_s) \in M^s \mid \mathcal{M} \models \varphi(a_1, \ldots, a_s)\}.$$

For $n \geq 1$, an *n-type* $p(x_1, \ldots, x_n)$ is a consistent set of $\mathcal{L}$-formulas each with free variables $x_1, \ldots, x_n$, and a type is *complete* if it is maximally consistent. Furthermore, given a tuple $\overline{a} \in \mathcal{M}$, and a set $B \subseteq M$, let $\mathrm{tp}^{\mathcal{M}}(\overline{a}/B)$ denote the type of $\overline{a}$ over B, i.e., the set of all $\mathcal{L}$-formulas with parameters from B that are satisfied by $\overline{a}$. If B is empty, we will omit it, and if $\mathcal{M}$ is clear from context, we will also omit it. Similarly, let $\mathrm{qftp}^{\mathcal{M}}(\overline{a}/B)$ denote the quantifier-free type of $\overline{a}$, i.e., the set of all quantifier-free $\mathcal{L}$-formulas that are satisfied by $\overline{a}$. Given a type $p(x, y)$ in two variables, let $p^{opp}(x, y)$ denote type $p(y, x)$ obtained by interchanging x and y.

An *atom* is a formula of the form $R(\overline{x})$ or $x_1 = x_2$. A *literal* is an atom or its negation.

The following definition is not standard, but will be useful for our purposes. We define the *strict* n-type of $(a_1, \ldots, a_n)$, denoted by $\mathrm{tp}^{\mathcal{M}}_{str}(a_1, \ldots, a_n)$ to be

$$\begin{aligned}\mathrm{tp}^{\mathcal{M}}_{str}(a_1, \ldots, a_n) =& \{\varphi(x_1, \ldots, x_n) \mid \varphi \text{ is a literal involving all variables, and } \mathcal{M} \models \varphi\} \\ & \cup \{x_i \neq x_j \mid 1 \leq i < j \leq n\},\end{aligned}$$

and we say that an n-type $p(x_1, \ldots, x_n)$ is *strict* if $x_i \neq x_j \in p$ for all $1 \leq i < j \leq n$, and all other elements of p are literals involving all variables $x_1, \ldots, x_n$.

A formula is in *disjunctive normal form (DNF)* if it is in the form

$$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m_i} \psi_{ij}(\overline{x}),$$

where $n, m_1, \ldots, m_n$ are positive integers and ψ_{ij} are literals involving the variables in $\overline{x}$. A formula is a *full DNF* if in every one of the inner conjuncts, each atom or its negation appears once, and each conjunct appears at most once (up to the order of the literals in the conjunct). As an example, if the language consists of one unary relation symbol P and one binary relation symbol R,

$$\begin{aligned}\phi(x_1, x_2) = &(P(x_1) \wedge P(x_2) \wedge R(x_1, x_1) \wedge R(x_2, x_2) \wedge R(x_1, x_2) \wedge R(x_2, x_1)) \\ &\vee (P(x_1) \wedge \neg P(x_2) \wedge R(x_1, x_2) \wedge \neg R(x_1, x_1) \wedge R(x_2, x_2) \wedge \neg R(x_2, x_1))\end{aligned}$$

is a full DNF. A formula is in *prenex normal form* if it is of the form $Q_1 x_1 \cdots Q_n x_n \ \psi(\overline{x}, \overline{y})$, where each Q_i is a quantifier and ψ is quantifier-free. $Q_1 x_1 \cdots Q_n x_n$ is called the *prefix* and ψ is called the *matrix*. A formula is in *Skolem normal form* if it is in prenex normal form and uses only universal quantifiers. Every quantifier-free formula is equivalent to a full DNF, and every formula is equivalent to one written in prenex normal form.

Definition II.2.1. For $n \in \mathbb{N}$, let $\mathbf{FO}^n$ denote the fragment of first-order logic consisting of formulas where at most n distinct logical variables are used.

A *counting quantifier* is a quantifier of the form $\exists_{\leq k}$, $\exists_{=k}$, or $\exists_{\geq k}$. The intepretation of $\exists_{\leq k}$ is given by: $\mathcal{M} \models \exists_{\leq k} x\ \psi(x, \bar{b})$ if and only $|\{a \in M \mid \mathcal{M} \models \psi(a, \bar{b})\}| \leq k$. The interpretations of $\exists_{=k}$ and $\exists_{\geq k}$ are given by the same condition, replacing $\leq$ with $=$ and $\geq$, respectively. Note that $\exists_{\leq k}$, $\exists_{=k}$, and $\exists_{\geq k}$ can be defined using the standard existential and universal quantifiers with $k+1$, $k+1$, and k distinct logical variables, respectively. For $n \in \mathbb{N}$, let $\mathbf{C}^n$ denote the fragment of first-order logic consisting of formulas that can use counting quantifiers where at most n distinct logical variables are used.

Given a binary relation symbol R, let $LO(R)$ denote the axioms stating that R is a linear order.

II.3 Query learning

Let X be a set (the *instance space*). A *concept* is a function $C : X \to \{0, 1\}$, and a *concept class* $\mathcal{C}$ on X is a nonempty set of concepts. We note that a concept is sometimes equivalently defined as a subset of X, but for our purposes it will be easier to reason about functions. Fix a concept $C \in \mathcal{C}$, which we call the *target*, and another concept class $\mathcal{H} \supseteq \mathcal{C}$, which we call the *hypothesis class*.

- An *equivalence query* (EQ) consists of a hypothesis $H \in \mathcal{H}$, to which the oracle answers *yes* if $H = C$, or with a counterexample $x \in X$ for which $H(x) \neq C(x)$.
- A *membership query* (MQ) consists of an element $x \in X$, to which the oracle responds with the value of $C(x)$.

The query learning procedure consists of consecutive rounds: in each round, the learner poses a query to the oracle, and the oracle responds with the corresponding answer. The learner

is allowed to choose queries based on the responses to previous queries, and succeeds if they submit the target as an equivalence query. In *EQ-learning*, the learner is only allowed to submit equivalence queries, while in *(EQ+MQ)-learning*, the learner can use both equivalence and membership queries.

Definition II.3.1 (Query Complexity)**.** Let $\mathcal{C} \subseteq \mathcal{H}$ be two concept classes. The *EQ-query complexity of* $\mathcal{C}$ *with queries from* $\mathcal{H}$ is defined to be the least n such that there is an algorithm for the learner to submit equivalence queries from $\mathcal{H}$ with the property that for any $C \in \mathcal{C}$, the learner can identify C within at most n queries, or ∞ if no such n exists.

The *(EQ+MQ)-query complexity of* $\mathcal{C}$ *with queries from* $\mathcal{H}$ is defined in the same way, except that the learner is allowed to also use membership queries.

We note that we make no assumptions on how the oracle chooses counterexamples for equivalence queries. That is to say, we study the worst-case bounds. Other work has studied the case where counterexamples are chosen from some known distribution, e.g. (3).

A significant stream of prior work has studied the relationship between query learning and combinatorial complexity measures of $\mathcal{C}$ and $\mathcal{H}$ (31; 5; 19; 30), in particular the *Littlestone dimension*, *consistency dimension*, and *strong consistency dimension* (also known as the *dual Helly number*), which we define now.

A *binary element tree* is a complete binary tree whose internal nodes are labeled by elements of X. A binary element tree T is *shattered* by $\mathcal{C}$ if there is a way to label all the leaves of T with elements of $\mathcal{C}$ such that the following condition holds: given a leaf node labeled by $A \in \mathcal{C}$, for each internal node above A labeled by $x \in X$, $A(x) = 1$ if and only if the (unique) path

from the root to A goes through the left child of x. An example of a (labelled) binary element tree of height 2 is given in Figure 1.

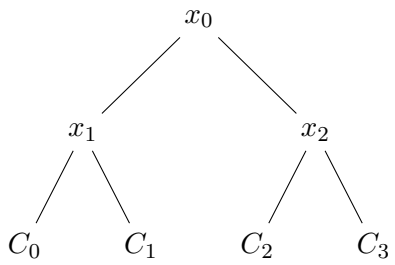

Figure 1: A binary element tree of height 2. Here, $x_0, x_1, x_2 \in X$, and $C_0, C_1, C_2, C_3 \in \mathcal{C}$. The tree is shattered exactly when $C_0(x_0) = C_0(x_1) = 1$, $C_1(x_0) = 1$ but $C_1(x_1) = 0$, $C_2(x_0) = 0$ but $C_2(x_2) = 1$, and $C_3(x_0) = C_3(x_2) = 0$

Definition II.3.2 (Littlestone dimension)**.** The *Littlestone dimension* of a concept class $\mathcal{C}$, denoted $\mathrm{Ldim}(\mathcal{C})$, is the maximum n such that there exists a binary element tree T of height n which is shattered by $\mathcal{C}$. If no such n exists, we say that $\mathrm{Ldim}(\mathcal{C}) = \infty$.

Remark II.3.3. A straightforward bound for the Littlestone dimension of a finite class $\mathcal{C}$ is $\mathrm{Ldim}(\mathcal{C}) \leq \log |\mathcal{C}|$. To see this, note that a binary element tree T of height $> \log |\mathcal{C}|$ has more than $|\mathcal{C}|$ leaves, so there must be two leaves with the same label $C \in \mathcal{C}$. Consider the internal node where the paths leading to these two leaves first differ; suppose it is labeled by x. These two paths are such that they both end at a leaf labeled by C, and one goes through the left

child of an internal node labeled by x and the other goes the right child of an internal node labeled by x. This implies that $x \in C$ and $x \notin C$, which is contradictory information, and so T cannot be shattered.

We now define (strong) consistency dimension. For the following bulleted definitions, let A, B be partial functions from X to $\{0,1\}$.

- $\mathrm{dom}(A)$ denotes domain of A.
- The *size* of A refers to the cardinality of $\mathrm{dom}(A)$.
- For a set $Y \subseteq \mathrm{dom}(A)$, the *restriction* of A to Y is the partial function $A|_Y$ defined by $A|_Y(x) = A(x)$ for $x \in Y$ and undefined outside of Y.
- We say that B *extends* A if $\mathrm{dom}(A) \subseteq \mathrm{dom}(B)$ and $B|_{\mathrm{dom}(A)} = A$. On the other hand, we say that A is a *restriction* of B.
- Given a concept class $\mathcal{C}$, A is *n-consistent with $\mathcal{C}$* if every size n restriction of A has an extension in $\mathcal{C}$. Otherwise, A is *n-inconsistent*.

Definition II.3.4 (Consistency Dimension)**.** The *consistency dimension* of $\mathcal{C}$ with respect to $\mathcal{H}$, denoted $\mathrm{Cdim}(\mathcal{C}, \mathcal{H})$, is the least n such that for every concept $A : X \to \{0,1\}$ that is n-consistent with $\mathcal{C}$, we have that $A \in \mathcal{H}$. If no such n exists, we say that $\mathrm{Cdim}(\mathcal{C}, \mathcal{H}) = \infty$. In the case that $\mathcal{H} = \mathcal{C}$, we will write $\mathrm{Cdim}(\mathcal{C})$ to denote $\mathrm{Cdim}(\mathcal{C}, \mathcal{C})$.

Remark II.3.5. By contrapositive, $\mathrm{Cdim}(\mathcal{C}, \mathcal{H}) \leq n$ if for every concept A such that $A \notin \mathcal{H}$, there is a subset $Y \subseteq X$ of size n such that $A|_Y$ cannot be extended to anything in $\mathcal{C}$. It is often

easier to work concretely with the contrapositive characterization of consistency dimension, so our proofs of bounds on consistency dimension will go through this direction.

Definition II.3.6 (Strong Consistency Dimension)**.** The *strong consistency dimension* of $\mathcal{C}$ with respect to $\mathcal{H}$, denoted $\mathrm{SCdim}(\mathcal{C}, \mathcal{H})$, is the least integer n such that for every partial function $A : X \to \{0, 1\}$ that is n-consistent with $\mathcal{C}$, there is an extension of A that is in $\mathcal{H}$. If no such n exists, we say that $\mathrm{SCdim}(\mathcal{C}, \mathcal{H}) = \infty$. In the case that $\mathcal{H} = \mathcal{C}$, we will write $\mathrm{SCdim}(\mathcal{C})$ to denote $\mathrm{SCdim}(\mathcal{C}, \mathcal{C})$.

While the definitions of consistency dimension and strong consistency dimension appear similar, the two quantities can differ significantly—even for DFAs, SCdim is much larger than Cdim. Concretely, let $\mathcal{L}_{n,m}$ denote the class of binary regular languages on strings of length at most m specified by a DFA with at most n states. Chase and Freitag showed that $\mathrm{Cdim}(\mathcal{L}_{n,m}) = O(n^2)$, but $\mathrm{SCdim}(\mathcal{L}_{n,m})$ cannot be polynomial in n and m (see the discussion following Theorem 3.3 of (19)). Furthermore, for any given $c, d < \infty$, they give an explicit example of a concept class $\mathcal{C}_{c,d}$ such that $\mathrm{Ldim}(\mathcal{C}) = d$ and $\mathrm{Cdim}(\mathcal{C}) = c + 1$, but $\mathrm{SCdim}(\mathcal{C}) = c^d$ (19, Examples 2.15 and 2.19).

We now state the best known bounds on query complexity in terms of the Littlestone and (strong) consistency dimensions.

Theorem II.3.7. *Let $\mathcal{C} \subseteq \mathcal{H}$ be two concept classes on a set X, and set $c = \mathrm{Cdim}(\mathcal{C}, \mathcal{H})$, $d = \mathrm{Ldim}(\mathcal{C})$, and $k = \mathrm{SCdim}(\mathcal{C}, \mathcal{H})$. Then:*

1. *(19, Theorem 2.24) The (EQ+MQ)-query complexity of $\mathcal{C}$ with queries from $\mathcal{H}$ is $O(cd)$.*

2. *(19, Theorem 2.6) The EQ-query complexity of $\mathcal{C}$ with queries from $\mathcal{H}$ is $O(c^d)$.*

3. *(30, Theorem 1) The EQ-query complexity of $\mathcal{C}$ with queries from $\mathcal{H}$ is $O(dk \log k)$.*

Note that item 3 seems to improve the dependence of EQ-query complexity on Littlestone dimension from exponential to linear. However, as seen above, the move from Cdim to SCdim can reintroduce an exponential dependence on Ldim, so it is not guaranteed to be a stronger bound.

II.4 Automata theory

In this section, we set notation and terminology for deterministic finite automata and regular languages, and review the Myhill-Nerode theorem.

Definition II.4.1 (Deterministic Finite Automata)**.** Let Σ be a finite set (the *input alphabet*). A *deterministic finite automaton (DFA) over* Σ consists of the following data:

- a finite set Q (the set of *states*);
- a function $\delta : Q \times \Sigma \to Q$ (the *transition function*);
- a state $q_0 \in Q$ (the *initial state*); and
- a set of states $F \subseteq Q$ (the set of *accepting states*).

Given an input string $x = x_1 x_2 \cdots x_n \in \Sigma^*$, and a DFA M, define the *run* of M on x to be the sequence of states $\alpha_0, \ldots, \alpha_n \in Q$ such that $\alpha_0 = q_0$, and for $1 \leq i \leq n$, we have that $\delta(\alpha_{i-1}, x_i) = \alpha_i$. A string $x \in \Sigma^*$ is *accepted* by M if the last state appearing in the run of M on x is in F. A *language* is a function $L : \Sigma^* \to \{0, 1\}$. We note that languages are often

defined as subsets of Σ^*, but here we use functions in order to align with our definition of a concept. A language L is *recognized* by a DFA M if M accepts x for all $x \in L$. We say that a language L is *regular* if it is recognized by some DFA.

The Myhill-Nerode theorem provides a useful syntactic characterization of regular languages. Given a language $L : \Sigma^* \to \{0, 1\}$, define an equivalence relation $\equiv_L$ on Σ^* as follows: for strings $x, y \in \Sigma^*$, we say that $x \equiv_L y$ iff $L(xz) = L(yz)$ for all $z \in \Sigma^*$.

Theorem II.4.2 (Myhill-Nerode)**.** *A language L is regular if and only if $\equiv_L$ has finitely many equivalence classes. Moreover, $\equiv_L$ has exactly n classes if and only if the minimal DFA recognizing L has exactly n states.*

II.5 Weighted Model Counting

A *weighted language* is a pair $(\mathcal{L}, w)$, where $\mathcal{L}$ is a finite relational language and $w : \mathcal{L} \to \mathbb{R}$. Let $p = \{R_i(\overline{x}) \mid i \in I\} \cup \{\neg R_j(\overline{x}) \mid j \in J\}$, where I and J are finite and $R_i, R_j \in \mathcal{L}$. That is, p is a finite type consisting only of literals. We will extend w to such types by setting $w(p) = \prod_{i \in I} w(R_i)$.

Let $\mathcal{M}$ be a finite $\mathcal{L}$-structure. The *weight* of $\mathcal{M}$ is

$$w(\mathcal{M}) := \prod_{R \in \mathcal{L}} w(R)^{|R(\mathcal{M})|}.$$

Note: we work in what is called the *symmetric* setting, in which every instance of a relation holding is given the same weight. For example, in the setting of graphs, the weight of a given graph only depends on the number of edges present, not on the actual pairs of vertices for which

there is an edge. The asymmetric setting, in which weight assigned to an instance of a relation also depends on the actual tuple for which the relation holds, is known to be strictly harder than the symmetric setting (29).

Let A be a set and $\mathcal{C}$ be collection of structures. Define

$$\mathsf{WFOMC}(\mathcal{C}, A, w) := \sum_{\substack{\mathcal{M} \in \mathcal{C} \\ M = A}} w(\mathcal{M})$$

$$\mathsf{WFOMC}(\mathcal{C}, n, w) := \mathsf{WFOMC}(\mathcal{C}, [n], w)$$

If T is a first-order theory, let $\mathcal{C}_T := \{\mathcal{M} \mid \mathcal{M} \models T\}$ denote the class of models of T and define

$$\mathsf{WFOMC}(T, A, w) := \mathsf{WFOMC}(\mathcal{C}_T, A, w), \text{ and similarly,}$$

$$\mathsf{WFOMC}(T, n, w) := \mathsf{WFOMC}(T, [n], w).$$

If the weight of every relation is 1, then $\mathsf{WFOMC}(T, n, w)$ is simply the number of models of T with domain $[n]$. We will refer to this value as $\mathsf{FOMC}(T, n)$ (the unweighted model count).

We note that in most literature on weighted model counting, the weight of a structure is defined in terms of two weight functions $w, \overline{w} : \mathcal{L} \to \mathbb{R}$, with $\overline{w}$ representing the weight of the negation of a relation. That is, the weight of a model is instead defined as

$$\mathsf{weight}_{w,\overline{w}}(\mathcal{M}) := \prod_{R \in \mathcal{L}} w(R)^{|R(\mathcal{M})|} \; \overline{w}(R)^{|\neg R(\mathcal{M})|},$$

and the weighted model count is defined accordingly. The two setups are equivalent up to rescaling the weights, so for simplicity of notation we only use one weight function. The equivalence is as follows:

Let $w, \overline{w} : \mathcal{L} \to \mathbb{R}$. We may assume that $\overline{w}$ is always non-zero—if $\overline{w}(R) = 0$ for some R, then any model for which R failed on any tuple would have weight 0, so we can assume that R always holds and ignore it. Thus, we may let $w' : \mathcal{L} \to \mathbb{R}$ be defined by $w'(R) = w(R)/\overline{w}(R)$. If $r(R)$ denotes the arity of the relation R, then

$$\begin{aligned}
\mathsf{weight}_{w,\overline{w}}(\mathcal{M}) &= \prod_{R \in \mathcal{L}} w(R)^{|R(\mathcal{M})|} \ \overline{w}(R)^{|\neg R(\mathcal{M})|} \\
&= \prod_{R \in \mathcal{L}} \left(\frac{w(R)}{\overline{w}(R)} \right)^{|R(\mathcal{M})|} \ \overline{w}(R)^{|R(\mathcal{M})|} \ \overline{w}(R)^{|\neg R(\mathcal{M})|} \\
&= \prod_{R \in \mathcal{L}} \left(w'(R)^{|R(\mathcal{M})|} \right) \left(\overline{w}(R)^{|M|^{r(R)}} \right) \\
&= w'(\mathcal{M}) \prod_{R \in \mathcal{L}} \overline{w}(R)^{|M|^{r(R)}}.
\end{aligned}$$

In particular, if $\mathsf{WFOMC}(T, n, w, \overline{w})$ denotes the weighted model count using both weight functions, we have that

$$\begin{aligned}\mathsf{WFOMC}(T, n, w, \overline{w}) &= \sum_{\substack{\mathcal{M} \models T \\ M = [n]}} \mathsf{weight}_{w,\overline{w}}(\mathcal{M}) \\ &= \sum_{\substack{\mathcal{M} \models T \\ M = [n]}} \left(w'(\mathcal{M}) \prod_{R \in \mathcal{L}} \overline{w}(R)^{n^r(R)} \right) \\ &= \mathsf{WFOMC}(T, n, w') \cdot \prod_{R \in \mathcal{L}} \overline{w}(R)^{n^{r(R)}}.\end{aligned}$$

The final factor can be computed in polynomial time with respect to n, so we may convert between the weighted model counts in both settings efficiently.

We conclude this subsection by giving the statement of the Skolemization procedure for weighted first-order model counting.

Theorem II.5.1. *(58, Theorem 3) Let $(\mathcal{L}, w)$ be a weighted language and φ be an $\mathcal{L}$-sentence. Then there is another weighted language $(\mathcal{L}', w')$ expanding $(\mathcal{L}, w)$ and an $\mathcal{L}'$-sentence φ such that for any $\mathcal{L}$-sentence ψ,*

$$\mathit{WFOMC}(\varphi \wedge \psi, n, w) = \mathit{WFOMC}(\varphi' \wedge \psi, n, w').$$

In particular, if we take $\psi = \top$, then $\mathit{WFOMC}(\varphi, n, w) = \mathit{WFOMC}(\varphi', n, w)$. Moreover, this procedure runs in time polynomial in the size of φ (and independent of n).

II.6 Hereditary properties

Definition II.6.1. Let $\mathcal{L}$ be a finite relational language. A *hereditary $\mathcal{L}$-property* is a class of $\mathcal{L}$-structures that is closed under isomorphism and substructure.

Fact II.6.2. Let $\mathcal{H}$ be a class of $\mathcal{L}$-structures. The following are equivalent:

(i) $\mathcal{H}$ is a hereditary $\mathcal{L}$-property.

(ii) There is a class $\mathcal{F}$ of $\mathcal{L}$-structures such that $\mathcal{H}$ is exactly the class of $\mathcal{L}$-structures that do not contain any substructure that is isomorphic to a structure from $\mathcal{F}$.

(iii) There is a universal theory T such that $\mathcal{H}$ is exactly the class of models of T.

Given a hereditary property $\mathcal{H}$, let $T_{\mathcal{H}}$ denote the universal theory that defines it. On the other hand, given a universal theory T, let $\mathcal{H}_T$ denote the hereditary property that it defines.

For a hereditary property $\mathcal{H}$, let $\mathcal{H}_n := \{\mathcal{M} \in \mathcal{H} \mid \mathcal{M} \text{ has domain } [n]\}$. The *speed* of $\mathcal{H}$ is the function $n \mapsto |\mathcal{H}_n|$. Note that $\mathsf{FOMC}(\mathcal{H}, n) = |\mathcal{H}_n|$.

Theorem II.6.3. *(36, Theorem 1.4) Suppose $\mathcal{H}$ is a hereditary $\mathcal{L}$-property, where $\mathcal{L}$ is a finite relational language with maximum arity r. Then one of the following holds:*

1. *There are $k \in \mathbb{N}$ and rational polynomials $p_1(x), \ldots, p_k(x)$ such that for sufficiently large n, $|\mathcal{H}_n| = \sum_{i=1}^{k} p_i(n) i^n$.*
2. *There is an integer $k \geq 2$ such that $|\mathcal{H}_n| = n^{n(1-\frac{1}{k}-o(1))}$.*
3. *There is $\epsilon > 0$ such that $n^{n(1-o(1))} \leq |\mathcal{H}_n| \leq 2^{n^{r-\epsilon}}$.*
4. *There is a constant $C > 0$ such that $|\mathcal{H}_n| = 2^{Cn^r + o(n^r)}$.*

CHAPTER III

QUERY LEARNING OF ADVICE AND NOMINAL AUTOMATA

III.1 Introduction

In this chapter, we prove query learning bounds for two variants of DFAs: advice DFAs and nominal DFAs. A more detailed discussion of the area of query learning of automata can be found in Section I.1.

For advice DFAs, we give the first known bound for the query complexity of advice DFAs. In particular, our result for advice DFAs is as follows: let $\mathcal{L}_k^{\text{adv}}(n, m)$ be the set of languages over an alphabet of size k recognized by an advice DFA on at most n states, restricted to strings of length at most m (the precise definition of an advice DFA is given in Section III.2).

Theorem III.2.6. *The (EQ+MQ)-query complexity of $\mathcal{L}_k^{adv}(n, m)$ with queries from $\mathcal{L}_k^{adv}(2n, m)$ is $O(n^3 mk \log n)$.*

The restriction on the length of the strings in $\mathcal{L}_k^{\text{adv}}(n, m)$ is necessary to make the problem tractable and can be thought of as giving a dependence on the length of the longest counterexample for equivalence queries. It may be of interest to see if there is an analogous version of the L^* algorithm for advice DFAs, and if so, how the bounds with that approach compare to ours.

For nominal DFAs, our result is as follows: given a G-alphabet A, let $\mathcal{L}_A^{\mathrm{nom}}(n,k)$ denote the set of G-languages recognized by a nominal DFA whose state set has at most n orbits and has dimension at most k (precise definitions of these terms are given in Section III.3).

Theorem III.3.45. *For a fixed G-alphabet A, the (EQ+MQ)-query complexity of $\mathcal{L}_A^{nom}(n,k)$ with queries from $\mathcal{L}_A^{nom}(n,k)$ is at most $\frac{n^{O(k)}}{k^k}$.*

Query learning of nominal DFAs was previously studied by Moerman et al. (42), who develop a generalization of Angluin's L^* algorithm for nominal DFAs. The bound that they derive is a complicated quantity (see (42, Corollary 1)), but in particular we observe the following: if the target automaton has n orbits and nominal dimension k, and p is the nominal dimension of the (fixed) alphabet, then their bound is lower bounded by both

(a) $\min\left(\left(\frac{nk}{e}\right)^m, \left(\frac{m}{e}\right)^{nk}\right)$ and

(b) $(k^n n!)^p$.

These are not explicitlly stated (42), but follow with some additional work (see Remark III.3.29).

Our result improves on (a) in the sense that our bound does not depend on the length of the longest counterexample, and improves the asymptotic dependence on n compared to (b) (polynomial in n with respect to k, instead of factorial in n). This is especially important in light of Corollary III.3.38, which says that in this setting, k is at most a constant multiple of n. However, we do not give any algorithmic complexity guarantees.

III.2 Learning advice DFAs

III.2.1 Overview of advice DFAs

In this subsection, we give an introduction to automata with advice. For a more comprehensive overview, see (32).

Definition III.2.1 (Advice DFA)**.** Let Σ and Γ be finite sets (the *input* and *advice alphabets*, respectively). An *advice DFA M over Σ with advice from Γ* consists of the following data:

- a finite set Q (the set of *states*);
- a length-ω string $A \in \Gamma^\omega$ over the advice alphabet (the *advice string*);
- a function $\delta : Q \times \Sigma \times \Gamma \to Q$ (the *transition function*);
- a state $q_0 \in Q$ (the *initial state*); and
- a set of states $F \subseteq Q$ (the set of *accepting states*).

Given an input string $x = x_1 x_2 \cdots x_n \in \Sigma^*$ and advice DFA M, define the *run* of M on x to be the sequence of states $\alpha_0, \ldots, \alpha_n \in Q$ such that $\alpha_0 = q_0$, and for each $i \in [n]$, we have that $\delta(\alpha_{i-1}, x_i, A_i) = \alpha_i$. A string $x \in \Sigma^*$ is *accepted* by M if the last state appearing in the run of M on x is in F. A language L is *recognized* by aa advice DFA M if M accepts x for all $x \in L$. We say that a language L is *regular with advice* if it is recognized by some advice DFA.

Advice DFAs extend classical DFAs with the addition of the advice string A (which is fixed beforehand as part of the automaton). The advice string is read in parallel with the input string; i.e., when M reads the n^{th} character of the input, it also has access to the n^{th} character

of the advice when deciding which transition to make. One way to think about the advice string is that it allows the transition function to vary at each step of the computation (although at a fixed step i, the transition behavior is the same regardless of the input string).

Advice DFAs satisfy a Myhill-Nerode characterization, under a variant of the $\equiv_L$ relation. Define an equivalence relation $\equiv_{L,m}$ on Σ^m by $x \equiv_{L,m} y$ iff $xz \in L \iff yz \in L$ for all $z \in \Sigma^*$. Notice that $\equiv_{L,m}$ is simply $\equiv_L$ restricted to strings of length m.

Theorem III.2.2. *(Myhill-Nerode for advice DFAs, cf. (32, Theorem 4)) Let L be a language.*

(i) Suppose L is accepted by an advice DFA that has n states. Then $\equiv_{L,m}$ has at most n classes for all $m \in \mathbb{N}$.

(ii) Suppose $\equiv_{L,m}$ has at most n classes for all $m \in \mathbb{N}$. Then there is an advice DFA on $2n$ states that recognizes L.

Note that this statement is more precise than the original version in (32); in particular, the relationship between the number of states and the number of $\equiv_{L,m}$-classes does not appear in the original theorem. However, this relationship is easily derived from its proof. We also note that in general, the bound of $2n$ states in the second item is tight, as witnessed by the following example.

Example III.2.3. Define $L : \{0,1\}^* \to \{0,1\}$ by

$$L(w) = \begin{cases} 1 & \text{if } (w \text{ has an even number of 0's} \wedge |w| \neq 2) \text{ or } (|w| = 3) \\ 0 & \text{otherwise} \end{cases}$$

Notice that $\equiv_{L,m}$ has at most two classes for every m—one for strings with an even number of 0's and one for strings with an odd number of 0's. However, it is not recognized by any DFA M with advice with 3 states. To see this, let M accept L. The runs of strings 00 and 01 both end in reject states of M, but since they are in separate $\equiv_{L,2}$-classes, they must end in distinct states. So there are at least two reject states. Similarly, the runs of 000 and 001 both end in distinct accept states. Thus there must be at least 4 states in M.

Replacing "w has an even number of 0's" with any regular language that has at most k equivalence classes in the Myhill-Nerode relation (for example, "the number of 0's in w is divisible by k"), we obtain a language L_k such that $\equiv_{L_k,m}$ has at most k classes for every m, but any advice DFA accepting L_k must have at least $2k$ states.

III.2.2 Learning bound for advice DFAs

For the remainder of the section, fix $k \in \mathbb{N}$ and let Σ be an alphabet of size k. We will not fix the advice alphabet Γ; however, notice that if we are constructing an automaton with at most n states, we may take Γ to have size n^{nk}—the advice string can be thought of as coding the transition function at a given step, and there are n^{nk} possible transition functions (functions from $Q \times \Sigma \to Q$).

Note that there are uncountably many languages that are regular with advice—for example, for every length-ω string A, the language consisting of the finite prefixes A is regular with advice. Thus we cannot have any finitary representation of arbitrary languages that are regular

with advice. Hence, we consider the case where we restrict to strings of bounded length. Let $\mathcal{L}_k^{\text{adv}}(n,m)$ denote the set

$$\mathcal{L}_k^{\text{adv}}(n,m) := \{L \subseteq \Sigma^{\leq m} \mid L \text{ is recognized by an advice DFA with at most } n \text{ states}\},$$

and let $\mathcal{E}_k(n,m)$ denote the set

$$\mathcal{E}_k(n,m) := \{L \subseteq \Sigma^{\leq m} \mid \equiv_{L,\ell} \text{ has at most } n \text{ classes for all } \ell \leq m\}.$$

By Theorem III.2.2, $\mathcal{L}_k^{\text{adv}}(n,m) \subseteq \mathcal{E}_k(n,m) \subseteq \mathcal{L}_k^{\text{adv}}(2n,m)$ for any $n, m \in \mathbb{N}$. Because of this, it is more convenient to compute the query complexity of $\mathcal{L}_k^{\text{adv}}(n,m)$ with queries $\mathcal{L}_k^{\text{adv}}(2n,m)$.

Proposition III.2.4. *The consistency dimension of $\mathcal{L}_k^{adv}(n,m)$ with respect to $\mathcal{L}_k^{adv}(2n,m)$ is at most $n(n+1)$.*

Proof. Let $L : \Sigma^{\leq m} \to \{0,1\}$, and suppose that $L \notin \mathcal{L}_k^{\text{adv}}(2n,m)$. Since $\mathcal{E}_k(n,m) \subseteq \mathcal{L}_k^{\text{adv}}(2n,m)$, we have that $L \notin \mathcal{E}_k(n,m)$. This means that there is $\ell \in [m]$ and strings $x_0, \ldots, x_n \in \Sigma^\ell$ which are pairwise $\equiv_{L,\ell}$-inequivalent. For each $0 \leq i < j \leq n$, let $z_{ij} \in \Sigma^*$ such that $L(x_i z_{ij}) \neq L(x_j z_{ij})$ (i.e., z_{ij} distinguishes x_i and x_j according to $\equiv_{L,\ell}$). Consider the set $B = \{x_k z_{ij} \mid 0 \leq i < j \leq n,\ k = i,j\}$, which has size $2\binom{n+1}{2} = n(n+1)$. Let $L' : \Sigma^{\leq m} \to \{0,1\}$ extend $L|_B$. Since L' agrees with L on B, $x_0, \ldots, x_n$ must be $\equiv_{L',\ell}$-inequivalent. Therefore, L' has at least $n+1$ $\equiv_{L',\ell}$-classes. Hence $L' \notin \mathcal{E}_k(n,m)$, so in particular $L' \notin \mathcal{L}_k^{\text{adv}}(n,m)$. So B is a set of size $n(n+1)$ which witnesses the fact that L is n-inconsistent with $\mathcal{L}_k^{\text{adv}}(n,m)$, and thus the consistency dimension of $\mathcal{L}_k^{\text{adv}}(n,m)$ w.r.t. $\mathcal{L}_k^{\text{adv}}(2n,m)$ is at most $n(n+1)$. □

Proposition III.2.5. *The Littlestone dimension of $\mathcal{L}_k^{adv}(n,m)$ is $O(nmk \log n)$.*

Proof. We can bound the Littlestone dimension of $\mathcal{L}_k^{\text{adv}}(n,m)$ by bounding its size and applying Remark II.3.3. As noted earlier, we may interpret Γ as coding all possible transition functions, so the advice string A simply determines the transition function at each step.

Consider an advice DFA such that $Q = [n]$ and $q_0 = 1$. To fully specify the behavior of the automaton on strings of length m, it is enough to choose one transition function for each step, as well as the set of accepting states. There are $(n^{nk})^m = n^{nmk}$ ways to choose the transition functions, and 2^n ways to choose the set of accepting states. So there are $n^{nmk}2^n$ total possible advice DFAs of this form.

Every language in $\mathcal{L}_k^{\text{adv}}(n,m)$ is accepted by an advice DFA of this form, so $|\mathcal{L}_k^{\text{adv}}(n,m)|$ is upper bounded by $n^{nmk}2^n$. Therefore

$$\begin{aligned}
\text{Ldim}(\mathcal{L}_k^{\text{adv}}(n,m)) &\leq \log |\mathcal{L}_k^{\text{adv}}(n,m)| \\
&= \log(n^{nmk}2^n) \\
&= nmk \log n + n \\
&= O(nmk \log n).
\end{aligned}$$

□

Combining Propositions III.2.4 and III.2.5 with Theorem II.3.7(1), we obtain:

Theorem III.2.6. *The (EQ+MQ)-query complexity of $\mathcal{L}_k^{adv}(n,m)$ with queries from $\mathcal{L}_k^{adv}(2n,m)$ is $O(n^3mk \log n)$.*

III.3 Learning nominal DFAs

III.3.1 Overview of nominal sets and DFAs

In this subsection, we define nominal sets and nominal DFAs. For a more comprehensive treatment, see (13). We will also state and prove some useful facts about nominal sets in Subsection III.3.2.

As mentioned in the introduction, one must leverage some underlying structure to properly generalize automata theory to infinite alphabets. For example, consider the alphabet $A = \mathbb{N}$, and define $L : A^* \to \{0, 1\}$ by $L(w) = 1$ if and only if $w = aa$ for some $a \in A$. An equivalently defined language over a finite alphabet is easily seen to be regular, and so we expect this example to be "regular" as well. An infinite automaton that recognizes L is shown in Figure 2.

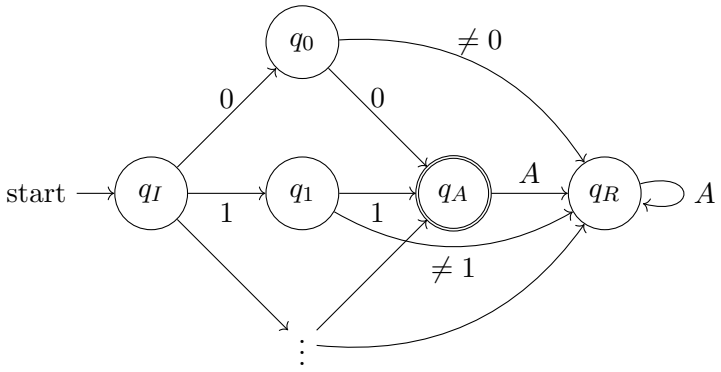

Figure 2: An infinite automaton that recognizes L. In some cases, we have combined infinitely many transitions into a single arrow, by assuming that we can compare values for equality.

This can be condensed further into an actually finite diagram, as shown in Figure 3.

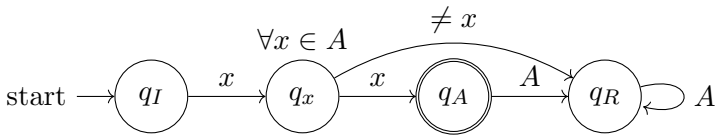

Figure 3: A finitary representation of Figure 2. We can compress the infinitely many distinct states for each character from A into a single "state" because it is enough to be able to compare whether or not the first and second characters read are equal or not.

In order to formalize these intuitions about representing infinitary data with finitary representations, Bojańczyk, Klin, and Lasota use the notion of *nominal sets*. Note that we do not work in the most general setting of (13)—we will only focus on what they call the *equality symmetry*. However, it is reasonable to expect that our results can generalize to other well-behaved symmetries.

Given a set A, let $\mathrm{Sym}(A)$ denote the group of permutations on A, i.e., the set of bijections from A to A with the operation of function composition. For the rest of the paper, let G denote $\mathrm{Sym}(\mathbb{N})$. Given a set X, a *(left) action* of G on X is an operation $\cdot : G \times X \to X$ such that:

1) for all $x \in X$, $e \cdot x = x$, where e is the identity function on $\mathbb{N}$, and

2) for all $\pi, \pi' \in G$ and $x \in X$, $(\pi \circ \pi') \cdot x = \pi \cdot (\pi' \cdot x)$.

Example III.3.1. Let $X = \mathbb{N}^2$. Define an action of G on X by $\pi \cdot (n, m) = (\pi(n), \pi(m))$. Observe that there is a permutation π for which $\pi \cdot (n_1, m_1) = (n_2, m_2)$ if and only if either (1) $n_1 = m_1$ and $n_2 = m_2$, or (2) $n_1 \neq m_1$ and $n_2 \neq m_2$. This example illustrates how the action of $\mathrm{Sym}(\mathbb{N})$ can be used to formalize the idea of being able to compare data values for equality.

Definition III.3.2. Fix X and an action of G on X. Given a subset $D \subseteq \mathbb{N}$ and an element $x \in X$, we say that D *supports* x if for every $\pi \in G$ for which $\pi|_D$ is the identity function on D, we have that $\pi \cdot x = x$. Furthermore, a finite set $D \subseteq \mathbb{N}$ is called the *least support of* x if

(1) D supports x,

(2) no proper subset of D supports x, and

(3) no other finite subset of $\mathbb{N}$ has properties (1) and (2).

We will use $supp(x)$ to denote the least support of x.

An equivalent characterization of an element having least support is that the intersection of any two finite supports of x also supports x.

Definition III.3.3. A *nominal set* is a set X along with an action of G on X such that every element of X has a least support.

For example, given any $k \in \mathbb{N}$, $\mathbb{N}^k$ with the action $\pi \cdot (a_1, \ldots, a_k) = (\pi(a_1), \ldots, \pi(a_k))$ is a nominal set, since every element $(a_1, \ldots, a_k)$ is supported by $\{a_1, \ldots, a_k\}$. Similarly, $\mathbb{N}^{(k)}$ with the action $\pi \cdot \{a_1, \ldots, a_k\} = \{\pi(a_1), \ldots, \pi(a_k)\}$ is also a nominal set.

Lemma III.3.4. *(13, Lemma 4.9) Let X be a nominal set and let $x \in X$. If D supports x, then for any $\pi \in G$, $\pi(D)$ supports $\pi \cdot x$.*

As a corollary, $|supp(x)| = |supp(\pi \cdot x)|$ for any $\pi \in G$.

Definition III.3.5. Let X be a nominal set. The *nominal dimension* of X is $\sup_{x \in X} |supp(x)|$ (i.e., the largest size of a least support).

In the literature on nominal sets, this is simply referred to as the dimension of X. However, in order to prevent confusion with other notions of dimension used in this paper, we will use the term nominal dimension.

Definition III.3.6. Let X be a nominal set. The *orbit* of an element $x \in X$, denoted $G \cdot x$, is the set

$$G \cdot x = \{\pi \cdot x \mid \pi \in G\}.$$

Every nominal set is partitioned into the disjoint union of its orbits. We say that X is *orbit-finite* if X is the union of only finitely many orbits.

By Lemma III.3.4, elements in the same orbit have the same size of least support, so orbit-finite nominal sets have finite nominal dimension.

Example III.3.7. Let $X = \mathbb{N}^2$, and define the action of G on X as in Example III.3.1. The orbits of X are $G \cdot (n, n)$ and $G \cdot (n, m)$, where $n, m \in \mathbb{N}$ and $n \neq m$. These are the sets of pairs whose coordinate are either equal or unequal, respectively. Thus X is orbit-finite with two orbits. Additionally, every element (n, m) has least support $\{n, m\}$, and so the nominal dimension of X is 2.

For $i = 1, \ldots, n$, let X_i be a set with an action of G on X_i. The *pointwise action* of G on $\prod_{i=1}^{n} X_i$ is the action given by $\pi \cdot (x_1, \ldots, x_n) = (\pi \cdot x_1, \ldots, \pi \cdot x_n)$.

Definition III.3.8 (Equivariance)**.** Let X be a nominal set. A subset $Y \subseteq X$ is *equivariant* if for any $y \in Y$ and $\pi \in G$, $\pi \cdot y \in Y$.

A relation R on $\prod_{i=1}^{n} X_i$, where each X_i is a nominal set, is *equivariant* if it is equivariant when considered as a subset of the product equipped with the pointwise action.

In particular, a function $f : X \to Y$ is equivariant exactly when

$$f(\pi \cdot x) = \pi \cdot f(x).$$

Remark III.3.9. Y is an equivariant subset of X if and only if Y is a union of orbits.

We are now ready to define nominal DFAs and state the nominal Myhill-Nerode theorem. Fix an orbit-finite nominal set A, which we will call the *G-alphabet*. The action of G on A naturally extends to A^*: given a string $w \in A^*$, $\pi \cdot w$ is the string obtained by letting π act on each individual character of w.

Definition III.3.10 (G-language)**.** A *G-language* is a function $L : A^* \to \{0, 1\}$ such that the set $\{x \in A^* \mid L(x) = 1\}$ is an equivariant subset of A^*.

Definition III.3.11 (Nominal DFA)**.** Let A be an orbit-finite nominal set (the *input alphabet*). A *nominal DFA M over A* consists of the following data:

- an orbit-finite nominal set Q (the set of *states*);

- an equivariant function $\delta : Q \times A \to Q$ (the *transition function*)
- a state $q_0 \in Q$ such that the orbit of q_0 is $\{q_0\}$ (the *initial state*)
- an equivariant subset $F \subseteq Q$ (the set of *accepting states*)

As a shorthand, we say that a nominal DFA M has n orbits or nominal dimension k when the state set has n orbits or nominal dimension k.

Given an input string $x = x_1 x_2 \cdots x_n \in A^*$, define the *run* of M on x to be the sequence of states $\alpha_0, \ldots, \alpha_n \in Q$ such that $\alpha_0 = q_0$, and for $1 \leq i \leq n$, we have that $\delta(\alpha_{i-1}, x_i) = \alpha_i$. A string $x \in \Sigma^*$ is *accepted* by M if the last state appearing in the run of M on x is in F. A language L is *recognized* by a nominal DFA M if M accepts x for all $x \in L$, and the language recognized by a nominal DFA must be a G-language (13, Definition 3.1). We say that a G-language L is *nominal regular* if it is recognized by some nominal DFA.

Example III.3.12. Let $A = \mathbb{N}$, and let $L : A \to \{0, 1\}$ be the example language from earlier defined by $L(w) = 1$ if and only if $w = aa$ for some $a \in A$. This language is a G-language. Additionally, in the automaton that recognizes L, we can define an action of G on the set of states by $\sigma \cdot q_i = q_{\sigma(i)}$ for any $\sigma \in G$ and $i \in \mathbb{N}$, while $\sigma \cdot q = q$ for all $\sigma \in G$ for the states q_I, q_A, and q_R. This automaton is a nominal DFA, and hence L is nominal regular.

Let L be a G-language, and define the usual relation $\equiv_L$ on A^* by: $x \equiv_L y$ if and only if for all $z \in A^*$, we have $L(xz) = L(yz)$. This relation is equivariant (13, Lemma 3.4), and hence by Lemma III.3.21, $A^*/\equiv_L$ is a nominal set. We will write $[x]_L$ to denote the $\equiv_L$-equivalence class of a string $x \in A^*$.

Theorem III.3.13. *(Myhill-Nerode for nominal DFAs (13, Theorem 5.2)) Let A be an orbit-finite nominal set, and let L be a G-language. Then the following are equivalent:*

1. *$A^*/\equiv_L$ has at most n orbits and has nominal dimension at most k;*
2. *L is nominal regular, and in particular is recognized by a nominal DFA with at most n orbits and nominal dimension at most k.*

Note that this statement is more precise than the original version in (13); in particular, the conditions on the number of orbits and the nominal dimension do not appear in the original theorem. For completeness, we give the proof of Theorem III.3.13, including the derivation of the bounds on the number of orbits and the nominal dimension.

Definition III.3.14 (Reachable Nominal DFA)**.** A nominal DFA M is said to be *reachable* if for every state q in M, there is $x \in A^*$ such that the run of M on x ends in state q.

Definition III.3.15 (Syntactic Automaton)**.** Fix an orbit-finite nominal alphabet A, and let $L : A^* \to \{0,1\}$ be a G-language. The *syntactic automaton* of L, denoted M_L is specified as follows:

- the state set is the set $A^*/\equiv_L$;
- the transition function is $\delta_L : A^*/\equiv_L \times A \to A^*/\equiv_L$ defined by

$$\delta_L([x]_L, a) = [xa]_L,$$

- the initial state is $[\epsilon]_L$;

- the set of accepting states is $\{[x]_L \mid x \in L\}$.

Lemma III.3.16. *(13, Lemma 3.6; Proposition 5.1) The syntactic automaton of a G-language is a reachable nominal DFA.*

By (13, Lemma 3.7), a G-language is always recognized by its syntactic automaton.

Definition III.3.17 (Automaton Homomorphism)**.** Let $M = (Q, \delta, q_0, F)$ and $M' = (Q', \delta', q_0', F')$ be two nominal DFAs over the same alphabet A. An *automaton homomorphism* from M to M' is an equivariant function $f : Q \to Q'$ such that:

- $f(q_0) = q_0'$;
- $q \in F \iff f(q) \in F'$ for every $q \in Q$; and
- $f(\delta(q, a)) = \delta'(f(q), a)$ for every $q \in Q, a \in A$.

If there exists an automaton homomorphism from M to M', then M and M' recognize the same language: for any string x, the run of M on x ends in state q if and only if the run of M' on x ends in the state $f(q)$, and $q \in F$ if and only if $f(q) \in F'$, so M accepts x if and only if M' accepts x.

Lemma III.3.18. *(13, Lemma 3.7) Let L be a G-language. For any reachable nominal DFA M that recognizes L, there is a surjective automaton homomorphism f from M to M_L.*

We can now prove bounds in the nominal Myhill-Nerode theorem:

Proof of Theorem III.3.13. (1. $\Rightarrow$ 2.) Suppose that $A^*/\equiv_L$ has at most n orbits and nominal dimension at most k. Then M_L is a nominal DFA that recognizes L, and its state set is $A^*/\equiv_L$ which by assumption has at most n orbits and nominal dimension at most k.

(2. $\Rightarrow$ 1.) Suppose that L is recognized by a nominal DFA M with at most n orbits and nominal dimension at most k. We may assume that M is reachable, so by Lemma III.3.18, there is a surjective automaton homomorphism f from M to the syntactic automaton M_L. By Lemma III.3.19 and Lemma III.3.20, the state set of M_L has at most n orbits and nominal dimension k. But the state set of M_L is exactly $A^*/\equiv_L$, and so we obtain the desired bounds.

□

III.3.2 Auxiliary results on nominal sets and G-languages

In this subsection, we state and prove several useful facts about nominal sets and G-languages. Some of these results do not appear in any previously published literature and may also be of independent interest.

We first give some lemmas that demonstrate how equivariant functions behave nicely with orbits and supports.

Lemma III.3.19. *Let $f : X \to Y$ be an equivariant function, and let $x \in X$. The image of the orbit of x (in X) under f is equal to the orbit of $f(x)$ (in Y).*

As a corollary, if f is surjective, then Y has at most as many orbits as X.

Proof. The orbit of $f(x)$ is the set $\{\pi \cdot f(x) \mid \pi \in G\}$. By equivariance of f, this is equal to $\{f(\pi \cdot x) \mid \pi \in G\} = f(\{\pi \cdot x \mid \pi \in G\})$, which is the image under f of the orbit of x. □

Lemma III.3.20. *(13, Lemma 4.8) Let $f : X \to Y$ be an equivariant function, $x \in X$, and $D \subseteq \mathbb{N}$. If D supports x, then D supports $f(x)$.*

As a corollary, $supp(f(x)) \subseteq supp(x)$, and hence if f is surjective, the nominal dimension of Y is at most the nominal dimension of X.

We will also work extensively with quotients by equivariant equivalence relations. Recall that if R is an equivalence relation on X, X/R denotes the set of equivalence classes of R, and is called the *quotient* of X by R.

Lemma III.3.21. *(13, Lemma 3.5) Let X be a nominal set and $R \subseteq X \times X$ be an equivariant equivalence relation. Then the quotient X/R is a nominal set, under the action $\pi \cdot [x]_R = [\pi \cdot x]_R$, and the quotient map $x \mapsto [x]_R$ is a surjective equivariant function.*

Lemma III.3.22. *Let X be a nominal set and $R \subseteq X \times X$ be an equivariant equivalence relation. Then $supp([x]_R) \subseteq supp(x)$ for every $x \in X$, and the nominal dimension of X/R is at most the nominal dimension of X.*

Proof. This follows immediately from Lemma III.3.20 and Lemma III.3.21. □

Lemma III.3.23. *Let X, Y be nominal sets. Given an equivariant function $F : X \to Y$ and equivariant equivalence relation $\equiv_Y$ on Y, there are induced equivariant equivalence relation $\equiv_X$ on X and equivariant function $f : X/\equiv_X \to Y/\equiv_Y$.*

Furthermore, f is injective, and if F is surjective, then f is also surjective.

Proof. Define $\equiv_X$ by $x_1 \equiv_X x_2$ if and only if $F(x_1) \equiv_Y F(x_2)$. A standard argument confirms that this is indeed an equivalence relation, equivariance follows from the fact that F is equivariant.

Next, define $f : X/\equiv_X \to \equiv_Y$ by $f([x]_{\equiv_X}) = [F(x)]_{\equiv_Y}$. This is well-defined: if $x_1 \equiv_X x_2$, then

$$\begin{aligned} f([x_1]_{\equiv_X}) &= [F(x_1)]_{\equiv_Y} \\ &= [F(x_2)]_{\equiv_Y} && \text{since } x_1 \equiv_X x_2 \Rightarrow F(x_1) \equiv_Y F(x_2) \\ &= f([x_2]_{\equiv_X}) \end{aligned}$$

f is equivariant since if $\pi \in G$, then $f(\pi \cdot [x]_{\equiv_X}) = f([\pi \cdot x]_{\equiv_X}) = [F(\pi \cdot x)]_{\equiv_Y} = [\pi \cdot F(x)]_{\equiv_Y} = \pi \cdot [F(x)]_{\equiv_Y} = \pi \cdot f([x]_{\equiv_X})$. f is injective since for $x_1, x_2 \in X$,

$$\begin{aligned} f([x_1]_{\equiv_X}) &= f([x_2]_{\equiv_X}) \\ \Rightarrow [F(x_1)]_{\equiv_Y} &= [F(x_2)]_{\equiv_Y} \\ \Rightarrow F(x_1) &\equiv_Y F(x_2) \\ \Rightarrow x_1 &\equiv_X x_2 \\ \Rightarrow [x_1]_{\equiv_X} &= [x_2]_{\equiv_X}. \end{aligned}$$

Finally, if F is surjective, then given $[y]_{\equiv_Y} \in Y/\equiv_Y$, there is $x \in X$ such that $F(x) = y$ and hence $f([x]_{\equiv_X}) = [y]_{\equiv_Y}$, so f is surjective. □

Since automata involve transition functions, we will need to understand products of nominal sets. Nominality is easily seen to be preserved under Cartesian products:

Proposition III.3.24. *The product of two nominal sets is nominal. In particular, the nominal dimension of the product of two nominal sets $X \times Y$ is at most the sum of the nominal dimensions of X and Y.*

Proof. Suppose X and Y are nominal with nominal dimensions k and ℓ, respectively. Let $(x, y) \in X \times Y$. Notice that $supp(x) \cup supp(y)$ has size at most $k + \ell$ and supports (x, y). □

On the other hand, in the most general setting, products of orbit-finite nominal sets need not be orbit-finite (13, Example 2.5). However, in our context of $\mathrm{Sym}(\mathbb{N})$, as well as some other nicely behaved symmetries, products of orbit-finite sets are known to be orbit-finite (13, Section 10). For our purposes, we will need an explicit bound on the number of orbits of products of orbit-finite sets.

Recall that $\mathbb{N}^{(k)} := \{(a_1, \ldots, a_k) \mid a_i \neq a_j \text{ for } i \neq j\}$. When equipped with the pointwise action of G, $\mathbb{N}^{(k)}$ is a single-orbit nominal set.

Lemma III.3.25. *(13, Lemma 4.13) Given a nominal set X of nominal dimension k that has exactly one orbit, there is an equivariant surjection $f_X : \mathbb{N}^{(k)} \to X$.*

Definition III.3.26. (cf. (42, Section 3)) Given $k_1, \ldots, k_n \in \mathbb{N}$, let $f_{\mathbb{N}}(k_1, \ldots, k_n)$ denote the number of orbits of $\mathbb{N}^{(k_1)} \times \cdots \times \mathbb{N}^{(k_n)}$.

Proposition III.3.27. *(cf. (42, Section 3)) Let X_i be a nominal set with ℓ_i orbits and nominal dimension k_i for $i = 1, \ldots, n$. Let $X := X_1 \times \cdots \times X_n$ and $\ell = \ell_1 \cdots \ell_n$. Then X has at most $\ell f_{\mathbb{N}}(k_1, \ldots, k_n)$ many orbits.*

Proposition III.3.27 is stated but not proven in (42), so for completeness, we give the proof.

Proof. Let $(x_1, \ldots, x_n) \in X$. Its orbit $G \cdot (x_1, \ldots, x_n)$ must be contained in the product $(G \cdot x_1) \times \cdots \times (G \cdot x_n)$. Hence, it is enough to bound the number of orbits of any given product $O_1 \times \ldots \times O_n$, where each O_i is an orbit of X_i, and multiply by the number of possible products. Fix orbits $O_1, \ldots, O_n$ of $X_1, \ldots, X_n$, respectively. By Lemma III.3.25, there are equivariant surjections $f_{O_i} : \mathbb{N}^{(k_i)} \to O_i$. Taking the product gives us the equivariant surjection

$$\mathbb{N}^{(k_1)} \times \cdots \times \mathbb{N}^{(k_n)} \xrightarrow{f_{O_1} \times \cdots \times f_{O_n}} O_1 \times \cdots \times O_n.$$

By Lemma III.3.19, $O_1 \times \cdots \times O_n$ has at most as many orbits as $\mathbb{N}^{(k_1)} \times \cdots \times \mathbb{N}^{(k_n)}$, which has $f_{\mathbb{N}}(k_1, \ldots, k_n)$ orbits. Now, there were $\ell = \ell_1 \ell_2 \cdots \ell_n$ ways to choose the orbits $O_1, \ldots, O_n$, so in total, X can have at most $\ell f_{\mathbb{N}}(k_1, \ldots, k_n)$ orbits. □

In light of Proposition III.3.27, we can upper bound the number of orbits of a product of nominal sets by calculating an upper bound on the value of $f_{\mathbb{N}}$. We do so for the case $n = 2$:

Proposition III.3.28. *Suppose (without loss of generality) that $k_1 \geq k_2$. Then*

$$\left(\frac{k_1}{e}\right)^{k_2} \leq \binom{k_1}{k_2} k_2! \leq f_{\mathbb{N}}(k_1, k_2) \leq (2k_1)^{k_2}.$$

In particular, if k_2 is a constant, then $f_{\mathbb{N}}(k_1, k_2) = \Theta(k_1^{k_2})$.

Proof. Consider tuples $\bar{a} = (a_1, \ldots, a_{k_1}) \in \mathbb{N}^{(k_1)}$ and $\bar{b} = (b_1, \ldots, b_{k_2}) \in \mathbb{N}^{(k_2)}$. The orbit of the pair $(\bar{a}, \bar{b})$ is exactly determined by the collection of indices i, j such that $a_i = b_j$. So to choose an orbit, we can first choose the number of indices $0 \leq r \leq k_2$ that $\bar{a}$ and $\bar{b}$ coincide on. Then

we need to choose r indices $i_1, \ldots, i_r$ of $\bar{a}$, r indices of $j_1, \ldots, j_r$ of $\bar{b}$, and a bijection between $\{i_1, \ldots, i_r\}$ and $\{j_1, \ldots, j_r\}$ in order to determine the indices that $\bar{a}$ and $\bar{b}$ coincide on. There are $\binom{k_1}{r}\binom{k_2}{r}r!$ ways to choose these.

Thus the total number of orbits is

$$f_{\mathbb{N}}(k_1, k_2) = \sum_{r=0}^{k_2} \binom{k_1}{r}\binom{k_2}{r}r!,$$

This is lower bounded by $\binom{k_1}{k_2}k_2!$. Since $k_2! \geq \left(\frac{k_2}{e}\right)^{k_2}$ and $\binom{k_1}{k_2} \geq \left(\frac{k_1}{k_2}\right)^{k_2}$ for any value of k_1, k_2, we have that

$$f_{\mathbb{N}}(k_1, k_2) \geq \binom{k_1}{k_2}k_2! \geq \left(\frac{k_1}{k_2}\right)^{k_2}\left(\frac{k_2}{e}\right)^{k_2} = \left(\frac{k_1}{e}\right)^{k_2}.$$

On the other hand, since $\binom{n}{k} \leq \frac{n^k}{k!}$ for any n and k,

$$\begin{aligned}
\sum_{r=0}^{k_2} \binom{k_1}{r}\binom{k_2}{r}r! &\leq \sum_{r=0}^{k_2} \frac{k_1^r}{r!}\binom{k_2}{r}r! \\
&= \sum_{r=0}^{k_2} k_1^r\binom{k_2}{r} \\
&\leq \sum_{r=0}^{k_2} k_1^{k_2}\binom{k_2}{r} \\
&= k_1^{k_2}\sum_{r=0}^{k_2}\binom{k_2}{r} \\
&= k_1^{k_2}2^{k_2} = (2k_1)^{k_2}
\end{aligned}$$

□

Proposition III.3.28 will help us to calculate an explicit upper bound on the number of possible nominal automata later, but to illustrate its usefulness we first use it to substantiate a comment from the introduction about previously known query bounds. The bound in (42, Corollary 1) involves a $f_{\mathbb{N}}(p(n+m), pn(k+k\log k+1))$ factor, where n is the number of orbits of the state set of the target automaton, k is the nominal dimension of the target automaton, p is the nominal dimension of the alphabet, and m is the length of the longest counterexample. Notice that $f_{\mathbb{N}}$ is non-decreasing in all coordinates. Therefore, $f_{\mathbb{N}}(p(n+m), pn(k+k\log k+1))$ is lower bounded by both $f_{\mathbb{N}}(pn, pnk)$ and $f_{\mathbb{N}}(m, nk)$. Applying the lower bounds from Proposition III.3.28 yields the following:

Remark III.3.29. The bound on the (EQ+MQ)-query complexity of nominal automata given in (42, Corollary 1) is at least $(k^n n!)^p$, and at least $\min\left(\left(\frac{nk}{e}\right)^m, \left(\frac{m}{e}\right)^{nk}\right)$.

Our next result is a bound on the number of single-orbit nominal sets. We will use this to bound the number of possible nominal DFAs, but we note that this result is fully general and may be of independent interest. An *isomorphism* of nominal sets is an equivariant bijection.

Proposition III.3.30. *The number of distinct (up to isomorphism) single-orbit nominal sets of nominal dimension at most k is at most $2^{O(k^2)}$.*

The proof of Proposition III.3.30 is quite technical and will involve a lot of machinery adapted from (13), which we now set up.

Definition III.3.31. (cf. (13, Definition 9.11)) A *support representation* is a pair (k, S), where $k \in \mathbb{N}$, and S is a subgroup of $\mathrm{Sym}([k])$.

Definition III.3.32. (cf. (13, Definition 9.14)) Given a support representation (k, S), the *semantics* of (k, S), denoted $[k, S]^{ec}$, is the set $\mathbb{N}^{(k)}/\equiv_S$, where $\equiv_S$ is defined as

$$(a_1, \ldots, a_k) \equiv_S (b_1, \ldots, b_k) \iff \exists \tau \in S \ \forall i \in [k],\ a_{\tau(i)} = b_i.$$

There is a natural action of G on $[k, S]^{ec}$ defined by

$$\pi \cdot [(a_1, \ldots, a_k)]_S = [(\pi(a_1), \ldots, \pi(a_k))]_S.$$

Proposition III.3.33. *(cf. (13, Proposition 9.15)) For any support representation (k, S), $[k, S]^{ec}$ is a single-orbit nominal set of nominal dimension k, and every single-orbit nominal set X of nominal dimension k is isomorphic to $[k, S]^{ec}$ for some $S \leq \mathrm{Sym}([k])$.*

Proposition III.3.34. *(cf. (13, Proposition 9.16)) Let $X = [k, S]^{ec}$ and $Y = [\ell, T]^{ec}$ be single-orbit nominal sets. Let*

$$U = \{u : [\ell] \to [k] \mid u \text{ is injective and } \forall \sigma \in S\ \exists \tau \in T,\ \sigma \circ u = u \circ \tau\}.$$

Equivariant functions from X to Y are in bijective correspondence with $U/\equiv_T$ (where $\equiv_T$ is as in Definition III.3.32).

Lemma III.3.35. *$[k, S]^{ec}$ is determined, up to isomorphism, by k and the conjugacy class of S in $\mathrm{Sym}([k])$.*

Proof. Let $X = [k, S]^{ec}$ and $Y = [\ell, T]^{ec}$ be single-orbit nominal sets. Suppose that $k = \ell$ and that S, T are conjugate in $\mathrm{Sym}([k])$. That is, there is a permutation $\rho : [k] \to [k]$ such that $\rho S \rho^{-1} = T$. Define $F^\rho : \mathbb{N}^{(k)} \to \mathbb{N}^{(k)}$ by $F^\rho((a_1, \ldots, a_k)) = (a_{\rho(1)}, \ldots, a_{\rho(k)})$ (i.e., reorder the input using ρ). Notice that F^ρ is an equivariant bijection. By Lemma III.3.23, F^ρ and $\equiv_T$ induce an equivalence relation on $\mathbb{N}^{(k)}$. Since $\rho S \rho^{-1} = T$, the induced equivalence relation is actually $\equiv_S$. Then the induced function f is an equivariant bijection between X and Y.

In the other direction, suppose there is an isomorphism $f : X \to Y$. The proof of Proposition III.3.34 gives us a bijection $u : [\ell] \to [k]$ such that $uS = Tu$. In particular, $k = \ell$ and u is a permutation in $\mathrm{Sym}([k])$ that witnesses that S and T are conjugate. □

With this machinery in hand, we can give the proof of Proposition III.3.30:

Proof. Let $X = [k', S]^{ec}$ be a single-orbit nominal set with nominal dimension at most k. By Lemma III.3.35, X is determined up to isomorphism by the value of k' and the conjugacy class of S in $\mathrm{Sym}([k'])$. Since the nominal dimension of X is k, $k' \leq k$, and so there are k choices for k'. It remains to count the number of conjugacy classes of subgroups of $\mathrm{Sym}([k])$. The number of subgroups of $\mathrm{Sym}([k])$ is $2^{\Theta(k^2)}$ (47, Theorem 4.2), which certainly gives an upper bound for the number of conjugacy classes of subgroups. Additionally, any subgroup has at most $k!$ conjugates (one for each permutation in $\mathrm{Sym}([k])$), so the number of conjugacy classes is also at most $2^{O(k^2)}/k! = 2^{O(k^2)}$.

Thus the number of single-orbit nominal sets of nominal dimension at most k is at most $k2^{O(k^2)} = 2^{O(k^2)}$. □

The next result will be needed when we prove bounds on the consistency dimension.

Lemma III.3.36. *Let X be any nominal set, let $x \in X$, and let $D = supp(x)$. Suppose that $\tau \in G$ fixes every element of D except for one. Then $\tau \cdot x \neq x$.*

Proof. Suppose for contradiction that $\tau \cdot x = x$. Let $i \in D$ be the only element of D such that $\tau(i) \neq i$, and notice that $\tau(i) \notin D$ since τ fixes every element of D other than i. We claim that $D \setminus \{i\}$ supports x. To see this, let $\sigma \in G$ such that σ fixes every element of $D \setminus i$. We need to show that $\sigma \cdot x = x$. We may assume that $\sigma(i) \neq i$, since otherwise σ fixes every element of D, and since D supports x, we would have $\sigma \cdot x = x$. Additionally, $\sigma(i) \notin D$ since σ fixes every element of D other than i. Now, let $j = \tau(i)$ and $k = \sigma(i)$. Let π_{jk} be the permutation that swaps j and k, and consider the permutation $\pi_{jk}\sigma$. We have that $\pi_{jk}\sigma|_D = \tau|_D$, as σ and τ both fix every element of $D \setminus \{i\}$, and $\tau(i) = j = \pi_{jk}(k) = \pi_{jk}(\sigma(i)) = (\pi_{jk}\sigma)(i)$. Since D supports x, we can deduce that $\pi_{jk}\sigma \cdot x = \tau \cdot x = x$. Then $\sigma \cdot x = \pi_{jk} \cdot x$. Since $j, k \notin D$, π_{jk} fixes every element of D, and so $\pi_{jk} \cdot x = x$. This shows that $\sigma \cdot x = x$, and since σ was an arbitrary permutation that fixed every element of $D \setminus \{i\}$, we may conclude that $D \setminus \{i\}$ supports x. This is a contradiction, since D is the *least* support of x. Hence $\tau \cdot x \neq x$. □

The next result is also used to prove bounds on the consistency dimension. However, it also has an interesting consequence (Corollary III.3.38) which may be of independent interest.

Lemma III.3.37. *Let L be a G-language such that $A^*/\equiv_L$ has n orbits. Then for every $x \in A^*$, there is $x' \in A^*$ and $\tau \in G$ such that $|x'| < n$ and $[x']_L = \tau \cdot [x]_L$. That is, every orbit of $A^*/\equiv_L$ contains an $\equiv_L$-class that is represented by a string of length strictly less than n.*

Proof. Suppose for contradiction that there is some $x \in A^*$ such that for every string x' of length strictly less than n, $[x']_L \notin G \cdot [x]_L$. We may assume that x is of minimal length m in its $\equiv_L$-class. Write $x = a_1 \cdots a_m$.

Consider the set of prefixes of x of length up to $n-1$, i.e., the set $\{\epsilon, a_1, a_1a_2, \ldots, a_1 \cdots a_{n-1}\}$. Since these all have length strictly less than n, the $\equiv_L$-classes $[a_1 \cdots a_i]_L$ for $0 \leq i \leq n-1$ (here, $a_1 \cdots a_0$ denotes the empty string) must belong to the $(n-1)$-many orbits that are not $G \cdot [x]_L$. As there are n strings in the set, there must be $0 \leq i < j \leq n-1$ such that $[a_1 \cdots a_i]_L$ and $[a_1 \cdots a_j]_L$ belong to the same orbit. That is, there is $\tau \in G$ such that $\tau \cdot (a_1 \cdots a_i) \equiv_L a_1 \cdots a_j$. Notice that $\equiv_L$ is preserved under appending common suffixes; i.e., if $x \equiv_L y$, then for any $z \in A^*$, $xz \equiv_L yz$. Thus $[\tau \cdot (a_1 \cdots a_i)]a_{j+1} \cdots a_m \equiv_L a_1 \cdots a_j a_{j+1} \cdots a_m = x$. However, notice that $[\tau \cdot (a_1 \cdots a_i)]a_{j+1} \cdots a_m$ is strictly shorter than x, and is in the same $\equiv_L$-class as x, which contradicts the assumption that x was of minimal length in its $\equiv_L$-class. Thus for every orbit of $A^*/\equiv_L$, there must be a string of length strictly less than n that is in the orbit. □

Corollary III.3.38. *If L is a G-language over an alphabet A which has nominal dimension p such that $A^*/\equiv_L$ has n orbits, then $A^*/\equiv_L$ has nominal dimension at most $(n-1)p$.*

Proof. Let $[x]_L \in A^*/\equiv_L$. By Lemma III.3.37, there is some $x' \in A^*$ with $|x'| < n$ and $\pi \in G$ such that $\pi \cdot [x]_L = [x']_L$. Then Lemma III.3.4 and Lemma III.3.22 tell us that

$$|supp([x]_L)| = |supp(\pi \cdot [x]_L)| = |supp([x']_L)| \leq |supp(x')|.$$

Each of the letters of x' is supported by a set of size at most p. Since G acts on x' coordinate-wise, the least support of x' is contained in the union of the least supports of all the letters of x', which is a set of size at most $(n-1)p$. Hence every element of $A^*/\equiv_L$ is supported by a set of size at most $(n-1)p$, and so the nominal dimension of $A^*/\equiv_L$ is at most $(n-1)p$. □

In particular, if A is fixed, the dimension of $A^*/\equiv_L$ is bounded by a constant multiple of the number of orbits, suggesting that the number of orbits of $A^*/\equiv_L$ is what ultimately controls the complexity of L, not the nominal dimension.

III.3.3 Littlestone dimension of nominal DFAs

To prove bounds on the Littlestone dimension of the class of nominal DFAs, we bound the number of possible nominal DFAs. We will do this by bounding the number of ways to choose each of the defining parameters.

Lemma III.3.39. *The number of possible state sets for a nominal DFA with n orbits and nominal dimension k is $2^{O(nk^2)}$.*

Proof. Since the state set Q is an orbit-finite nominal set with n orbits and nominal dimension k, we count the number of such sets. We can view Q as the disjoint union of n single-orbit nominal sets, each with nominal dimension at most k, just by considering each orbit independently. So the number of possible state sets is the number of single-orbit nominal sets of nominal dimension at most k, raised to the power n. By Proposition III.3.30, the number of single-orbit nominal sets of nominal dimension $\leq k$ is at most $2^{O(k^2)}$. So the number of nominal sets with n orbits and nominal dimension k is at most $\left(2^{O(k^2)}\right)^n = 2^{O(nk^2)}$. □

Fix an input alphabet A, where A has ℓ orbits and nominal dimension p.

Lemma III.3.40. *The number of possible transition behaviors for a nominal DFA with n orbits and nominal dimension k is at most*

$$(n(k+p)!)^{O(nk^p)}.$$

Proof. We count the number of equivariant functions $\delta : Q \times A \to Q$, where Q has n orbits and nominal dimension k. By Lemma III.3.19, a single orbit of $Q \times A$ must map into a single orbit of Q, so we can first choose a target orbit of Q for each orbit of $Q \times A$. By Proposition III.3.27, $Q \times A$ has at most $n\ell f_{\mathbb{N}}(k,p)$ orbits. Since p is fixed, by Proposition III.3.28, for any k, $f_{\mathbb{N}}(k,p) = O(k^p)$. Furthermore, since ℓ is also fixed, $Q \times A$ has at most $O(nk^p)$ orbits. Thus there are $n^{O(nk^p)}$ many ways to choose the target orbits of each orbit of $Q \times A$.

Once we have chosen a target orbit of Q for each orbit of $Q \times A$, we must choose an equivariant function from each orbit O_1 of $Q \times A$ to one orbit O_2 of Q. By Proposition III.3.24, the nominal dimension of $Q \times A$ is at most $k+p$, and so O_1 also has nominal dimension at most $k+p$. Similarly, O_2 has nominal dimension at most k. Therefore $O_1 = [k', S]^{ec}$ where $k' \leq k+p$, and $O_2 = [\ell, T]^{ec}$ where $\ell \leq k$. By Proposition III.3.34, the number of equivariant functions from O_1 to O_2 is upper bounded by the number of injections $[\ell] \to [k]$, which is in turn at most $\frac{(k+p)!}{p!} \leq (k+p)!$.

We must choose one such equivariant function for each orbit of $Q \times A$, so the total number of choices of all of these functions is at most $((k+p)!)^{O(nk^p)}$. Once we have done this, we have chosen a transition behavior $\delta : Q \times A \to Q$.

Hence the total number of possible transition behaviors is upper bounded by

$$n^{O(nk^p)} \cdot ((k+p)!)^{nk^p} = (n(k+p)!)^{O(nk^p)}.$$

□

Let $\mathcal{L}_A^{\mathrm{nom}}(n,k)$ denote the set of G-languages over A recognized by a nominal DFA with at most n orbits and nominal dimension at most k.

Proposition III.3.41. *The Littlestone dimension of $\mathcal{L}_A^{nom}(n,k)$ is at most*

$$O\left(nk^p\left(\log n + k\log k\right)\right)$$

Proof. We bound the Littlestone dimension by bounding the size of $\mathcal{L}_A^{\mathrm{nom}}(n,k)$. To choose a nominal DFA with at most n orbits and nominal dimension at most k, we must choose the state set, transition function, initial state, and set of accepting states. By Lemma III.3.39, there are at most $2^{O(nk^2)}$ choices for the state set. By Lemma III.3.40, there are at most $(n(k+p)!)^{O(nk^p)}$ choices for the transition function. There are at most n choices for the initial state (since the

initial state must be an orbit of its own) and 2^n choices for the accepting states (since the set of accepting states is a union of orbits). So in total, we can upper bound $|\mathcal{L}_A^{nom}(n,k)|$ by

$$|\mathcal{L}_A^{nom}(n,k)| \leq 2^{O\left(nk^2\right)} \left(n(k+p)!\right)^{O(nk^p)} n2^n.$$

Applying Remark II.3.3, using Stirling's approximation, and remembering that p is constant,

$$\begin{aligned}
\mathrm{Ldim}(\mathcal{L}_A^{nom}(n,k)) &\leq \log |\mathcal{L}_A^{nom}(n,k)| \\
&\leq O\left(nk^2\right) + O\left(nk^p\right)\left(\log n + \log((k+p)!)\right) + \log n + n \\
&\leq O\left(nk^p\left(\log n + (k+p)\log(k+p)\right)\right) \\
&= O\left(nk^p\left(\log n + k\log k\right)\right).
\end{aligned}$$

□

III.3.4 Consistency dimension of nominal DFAs

It remains to bound the consistency dimension, which is the purpose of this subsection. Our first goal is to show that the nominal dimension of $A^*/\equiv_L$ is witnessed by a small set.

Proposition III.3.42. *Let L be a G-language over alphabet A, and suppose that $A^*/\equiv_L$ has nominal dimension at least $k+1$. Then there is a set $B \subseteq A^*$ of size $2(k+1)$ such that for any G-language L', if $L|_B = L'|_B$, then $A^*/\equiv_{L'}$ also has nominal dimension at least $k+1$.*

Proof. Since $A^*/\equiv_L$ has nominal dimension $\geq k+1$, there is some $x_0 \in A^*$ such that $|supp([x_0]_L)| \geq k+1$. Let D denote $supp([x_0]_L)$, and let D_0 be a subset of D of size $k+1$. Let

$j = \max(D) + 1$, and for each $i \in D_0$, let τ_i be the permutation that swaps i and $i + j$. Notice that τ_i fixes all but one element of D, and so by Lemma III.3.36, $[\tau_i \cdot x_0]_L = \tau_i \cdot [x_0]_L \neq [x_0]_L$. Thus for each $i \in D_0$, there is some $z_i \in A^*$ such that $L((\tau_i \cdot x_0)z_i) \neq L(x_0 z_i)$. Let

$$B = \{(\tau_i \cdot x_0)z_i \mid i \in D_0\} \cup \{x_0 z_i \mid i \in D_0\},$$

which has size $2(k + 1)$.

Now, let L' be a G-language extending $L|_B$, and suppose for contradiction that $A^*/\equiv_{L'}$ has nominal dimension at most k. That is, for every $w \in A^*$, $|supp([w]_{L'})| \leq k$. In particular, $|supp([x_0]_{L'})| \leq k$. For each $i \in D_0$, L' agrees with L on $(\tau_i \cdot x_0)z_i$ and $x_0 z_i$, so $L'((\tau_i \cdot x_0)z_i) \neq L'(x_0 z_i)$. This shows that $\tau_i \cdot [x_0]_{L'} = [\tau_i \cdot x_0]_{L'} \neq [x_0]_{L'}$. Since the only elements not fixed by τ_i are i and $i + j$, it must be that at least one of i and $i + j$ are in the least support of $[x_0]_{L'}$. Thus for each $i \in D_0$, $supp([x_0]_{L'})$ contains at least one of i and $i + j$. Since $j > \max(D)$, all values $i, i + j$ for $i \in D_0$ are distinct, and so $|supp([x_0]_{L'})| \geq |D_0| = k + 1$, which contradicts the fact that $|supp([x_0]_{L'})| \leq k$. □

Our next goal is to show that the number of orbits of $A^*/\equiv_L$ is witnessed by a small set.

Proposition III.3.43. *Let L be a G-language over alphabet A, where A has nominal dimension p, and suppose that $A^*/\equiv_L$ has at least $n + 1$ orbits. Then there is a set $B \subseteq A^*$ of size $2\binom{n+1}{2}\binom{pn}{k}(3pn)^k$ such that for any G-language L', if $L|_B = L'|_B$, then $A^*/\equiv_{L'}$ also has at least $n + 1$ orbits.*

Proof. Since $A^*/\equiv_L$ has at least $n+1$ orbits, there are strings $x_0, \ldots, x_n$ such that $[x_i]_L$ all belong to distinct orbits. That is, for every $\tau \in G$ and $0 \leq i < j \leq n$, $[\tau \cdot x_i]_L = \tau[x_i]_L \neq [x_j]_L$, and thus there is $z_{ij}^\tau \in A^*$ such that $L((\tau \cdot x_i)z_{ij}^\tau) \neq L(x_j z_{ij}^\tau)$. Moreover, by Lemma III.3.37, we may assume that $|x_i| \leq n$ for each $0 \leq i \leq n$.

For each $0 \leq i \leq n$, let $D_i = supp(x_i)$. Since the nominal dimension of A is p, we have that $|D_i| \leq p \cdot |x_i| \leq pn$. Also, let D be some subset of $\mathbb{N}$ such that D is disjoint from $D_0, \ldots, D_n$, and $|D| = \max\limits_{0 \leq i \leq n} |D_i| \leq pn$.

Now, for each $0 \leq i < j \leq n$, let

$$\Sigma'_{ij} := \{\sigma' : D'' \to D_i \cup D_j \cup D \ \mid \ D'' \subseteq D_i \text{ of size } k \text{ and } \sigma' \text{ is an injection}\}.$$

Notice that Σ'_{ij} has size at most $\binom{pn}{k}(3pn)^k$: to choose a σ', we choose a subset of D_i of size k and a value from $D_i \cup D_j \cup D$ for each of the k inputs. For each $\sigma' \in \Sigma'_{ij}$, let $\sigma \in G$ be an arbitrary but fixed extension of σ' to a permutation of $\mathbb{N}$, and let Σ_{ij} consist of all such σ (so $|\Sigma_{ij}| = |\Sigma'_{ij}| \leq \binom{pn}{k}(3pn)^k$).

Now, let

$$B = \{(\sigma \cdot x_i)z_{ij}^\sigma \ \mid \ 0 \leq i < j \leq n, \ \sigma \in \Sigma_{ij}\} \cup$$

$$\{x_j z_{ij}^\sigma \ \mid \ 0 \leq i < j \leq n, \ \sigma \in \Sigma_{ij}\}.$$

B has size at most $2\binom{n+1}{2}\binom{pn}{k}(3pn)^k$. Let L' be a G-language extending $L|_B$, and assume for contradiction that $A^*/\equiv_{L'}$ has at most n orbits. Thus there must be $0 \leq i < j \leq n$ such that

$[x_i]_{L'}$ is in the same orbit as $[x_j]_{L'}$. Let $\pi \in G$ such that $\pi \cdot [x_i]_{L'} = [x_j]_{L'}$. π does not have to be in Σ_{ij}, and so we cannot directly derive a contradiction. Instead, we will alter π in order to obtain a $\sigma \in \Sigma_{ij}$ such that $\sigma \cdot [x_i]_{L'} = [x_j]_{L'}$.

Let $\pi' \in G$ be defined as follows: let $\pi(D_i)$ denote the image of D_i under π, and for each element $a \in \pi(D_i) \setminus (D_i \cup D_j \cup D)$, select a unique element $b_a \in D \setminus \pi(D_i)$. This is possible because $|D| \geq |D_i| = |\pi(D_i)|$, and so $|D \setminus \pi(D_i)| \geq |\pi(D_i) \setminus D| \geq |\pi(D_i) \setminus (D_i \cup D_j \cup D)|$. Let π' be the permutation that swaps each $a \in \pi(D_i) \setminus (D_i \cup D_j \cup D)$ with its corresponding $b_a \in D \setminus \pi(D_i)$, and fixes every other element of $\mathbb{N}$.

Notice that π' swaps elements not in D_j with elements in D, which is disjoint from D_J, and so $\pi'|_{D_j} = \mathrm{id}_{D_j}$. Hence $\pi' \cdot x_j = x_j$. Also, if $a \in D_i$, then $\pi' \circ \pi(a) \in D_i \cup D_j \cup D$: if $\pi(a) \in D_i \cup D_j \cup D$, then by definition π' does not affect $\pi(a)$ and so $\pi'(\pi(a)) = \pi(a) \in D_i \cup D_j \cup D$, whereas if $\pi(a) \in \pi(D_i) \setminus (D_i \cup D_j \cup D)$, π' will transpose $\pi(a)$ with an element in D. Let D'_i denote $supp([x_i]_{L'})$. Since $A^*/\equiv_{L'}$ has nominal dimension at most k, $|D'_i| \leq k$. Also, by Lemma III.3.22, $D'_i \subseteq supp(x_i) = D_i$. This then tells us that $(\pi' \circ \pi)|_{D'_i}$ is an injection from a subset of D_i of size at most k into $D_i \cup D_j \cup D$. Therefore, there is some $\sigma \in \Sigma_{ij}$ such that

$$\sigma|_{D'_i} = (\pi' \circ \pi)|_{D'_i}.$$

We may then deduce that

$$\begin{aligned}
[\sigma \cdot x_i]_{L'} &= \sigma \cdot [x_i]_{L'} \\
&= (\pi' \circ \pi) \cdot [x_i]_{L'} \\
&= \pi' \cdot (\pi \cdot [x_i]_{L'}) \\
&= \pi' \cdot [x_j]_{L'} \\
&= [\pi' \cdot x_j]_{L'} \\
&= [x_j]_{L'}.
\end{aligned}$$

This means that for all $z \in A^*$, $L'((\sigma \cdot x_i)z) = L'(x_j z)$. In particular, we may choose $z = z_{ij}^{\sigma}$. However, since $(\sigma \cdot x_i)z_{ij}^{\sigma}$ and $x_j z_{ij}^{\sigma}$ are in B, it must be that

$$\begin{aligned}
L((\sigma \cdot x_i)z_{ij}^{\sigma}) &= L'((\sigma \cdot x_i)z_{ij}^{\sigma}) \\
&= L'(x_j z_{ij}^{\sigma}) \\
&= L(x_j z_{ij}^{\sigma}),
\end{aligned}$$

a contradiction! So we may finally conclude that $A^*/\equiv_{L'}$ has at least $n+1$ orbits. □

Combining these facts, we can prove a bound on the consistency dimension of $\mathcal{L}_A^{nom}(n,k)$:

Proposition III.3.44. *The consistency dimension of $\mathcal{L}_A^{nom}(n,k)$ with respect to itself is at most $2\binom{n+1}{2}\binom{pn}{k}(3pn)^k$.*

Proof. Let $L : A^* \to \{0,1\}$, and suppose that $L \notin \mathcal{L}_A^{nom}(n,k)$, i.e., L is not recognized by any nominal DFA with at most n orbits and nominal dimension at most k. We must find a set B of at most $2\binom{n+1}{2}\binom{pn}{k}(3pn)^k$ strings such that any function extending $L|_B$ is not in $\mathcal{L}_A^{nom}(n,k)$.

Case 1: L is not equivariant. Then there is $x_0 \in A^*$ and $\pi \in G$ such that $L(x_0) \neq L(\pi \cdot x_0)$. Set $B = \{x_0, \pi \cdot x_0\}$. Any function L' extending $L|_B$ cannot be equivariant, and therefore cannot be nominal regular, so $L' \notin \mathcal{L}_A^{nom}(n,k)$.

Case 2: L is equivariant, and is recognized by some nominal DFA with at most n orbits. By Theorem III.3.13, $A^*/\equiv_L$ has at most n orbits. Thus the nominal dimension of $A^*/\equiv_L$ must be at least $k+1$, or else another application of Theorem III.3.13 would imply that L is recognized by a nominal DFA with at most n orbits and nominal dimension at most k, contradicting the assumption that $L \notin \mathcal{L}_A^{nom}(n,k)$. Let $B \subseteq A^*$ of size $2(k+1)$ be as given by Proposition III.3.42.

Now, let $L' : A^* \to \{0,1\}$ extend $L|_B$, and suppose for contradiction that $L' \in \mathcal{L}_A^{nom}(n,k)$. Since L' is recognized by a nominal DFA, it must be a G-language, so by the definition of B, $A^*/\equiv_{L'}$ has nominal dimension at least $k+1$. However, by Theorem III.3.13 and the fact that L' is recognized by a nominal DFA with at most n orbits and nominal dimension at most k, $A^*/\equiv_{L'}$ has nominal dimension at most k, a contradiction! So $L' \notin \mathcal{L}_A^{nom}(n,k)$.

Case 3: L is equivariant, and L is not recognized by any nominal DFAs with at most n orbits. By Theorem III.3.13, $A^*/\equiv_L$ has at least $n+1$ orbits. Let $B \subseteq A^*$ of size $2\binom{n+1}{2}\binom{pn}{k}(3pn)^k$ be as given by Proposition III.3.43.

Now, let $L' : A^* \to \{0,1\}$ extend $L|_B$, and suppose for contradiction that $L' \in \mathcal{L}_A^{nom}(n,k)$. Since L' is recognized by a nominal DFA, it must be a G-language, so by the definition of B, $A^*/\equiv_{L'}$ has at least $n+1$ orbits. However, by Theorem III.3.13 and the fact that L' is recognized by a nominal DFA with at most n orbits and nominal dimension at most k, $A^*/\equiv_{L'}$ has at most n orbits, a contradiction! So $L' \notin \mathcal{L}_A^{nom}(n,k)$.

In all three cases, we found a set B of size at most $2\binom{n+1}{2}\binom{pn}{k}(3pn)^k$ such that any language extending $L|_B$ cannot be in $\mathcal{L}_A^{nom}(n,k)$. We can then conclude that the consistency dimension of $\mathcal{L}_A^{nom}(n,k)$ with respect to itself is at most $2\binom{n+1}{2}\binom{pn}{k}(3pn)^k$. □

III.3.5 Learning bound for nominal DFAs

We can now use the bounds we proved for Littlestone dimension and consistency dimension to prove our main result on nominal DFAs:

Theorem III.3.45. *For a fixed G-alphabet A, the (EQ+MQ)-query complexity of $\mathcal{L}_A^{nom}(n,k)$ with queries from $\mathcal{L}_A^{nom}(n,k)$ is at most $\frac{n^{O(k)}}{k^k}$.*

Proof. By Proposition III.3.41, the Littlestone dimension of $\mathcal{L}_A^{nom}(n,k)$ is at most $O\left(nk^p\left(\log n + k\log k\right)\right)$. By Proposition III.3.44, the consistency dimension of $\mathcal{L}_A^{nom}(n,k)$ with respect to itself is at most $2\binom{n+1}{2}\binom{pn}{k}(3pn)^k$. This is upper bounded by $n(n+1)\frac{(e\cdot pn)^k}{k^k}(3pn)^k$, which is in turn at most $\frac{n^{O(k)}}{k^k}$. Applying Theorem II.3.7, the query complexity of $\mathcal{L}_A^{nom}(n,k)$ with queries from $\mathcal{L}_A^{nom}(n,k)$ is at most $O\left(nk^p\left(\log n + k\log k\right)\frac{n^{O(k)}}{k^k}\right) = \frac{n^{O(k)}}{k^k}$. □

CHAPTER IV

HEREDITARY PROPERTIES AND WEIGHTED FIRST-ORDER MODEL COUNTING

IV.1 Introduction

In this chapter, we study the weighted first-order model counting problem and connections to the speeds of hereditary properties. Most work on weighted first-order model counting has focused on proving computational complexity results, and in particular identifying certain fragments of first-order logic for which the weighted model counting problem is polynomial-time computable. The first of these results was for $\mathbf{FO}^2$, the fragment of first-order logic in which only two logical variables are allowed (57; 58) (with a standalone proof given in (12)). This was later extended to $\mathbf{C}^2$, which allows the sentences to also include counting quantifiers (33). Most recently, the results on $\mathbf{FO}^2$ and $\mathbf{C}^2$ were extended to allow an additional axiom stating that one of the binary relations in the language forms a linear order (55).

On the other hand, most work in the unweighted setting comes from combinatorics, where the focus was in characterizing the possible asymptotic growth rates of the model counting function for various classes of combinatorial objects. The most general result in this area is due to Laskowski and Terry, who work with hereditary properties of $\mathcal{L}$-structures for any finite relational language $\mathcal{L}$, and prove that the asymptotic growth rate of a hereditary $\mathcal{L}$-property must fall into one of four asymptotic growth rates that have distinct "jumps" between them

((36), and see Theorem II.6.3). This setting captures many natural combinatorial objects that were previously studied, such as graphs, directed graphs, tournaments, ordered graphs, k-uniform hypergraphs, posets, and linear orders.

While the two streams of work in the weighted and unweighted settings have differing goals, both often aim to gain insight into underlying structural conditions on the classes they are working with in order to prove their results, whether it be for computational guarantees in the weighted case or asymptotic characterizations in the unweighted case. Our main aim is to study how the structural conditions developed in each setting can play a role in further understanding the other setting.

In Section IV.2, we prove an auxilary result which allows us to assume that all relations only convey information of their exact arity, and moreover that the maximum arity of the language is at most the maximum number of logical variables that appear in a sentence for which we are computing the (weighted or unweighted) model count. This fact is implicitly used in several prior works on weighted model counting, citing what is called Scott's reduction as described in (28). However, as stated, Scott's reduction only preserves satisfiability and the finite spectrum of the sentence, and not necessarily the (weighted) model count. Hence, we provide a full proof that computational efficiency of the weighted model count is preserved by this transformation.

In Sections IV.3 and IV.4, we use the structural characterizations for the slowest speed of hereditary $\mathcal{L}$-properties and the fastest speed of hereditary graph properties to give formulas for the weighted model count in specific cases.

In Section IV.5, we study the unweighted model counting problem for $\mathbf{FO}^2$ and use the result as inspiration for an alternative proof of the polynomial-time computability of the weighted model counting problem for $\mathbf{FO}^2$.

We conclude the introduction by making a note about terminology. For the entire chapter, we will only use quantifier-free types, so all uses of the word "type" in this chapter will refer to a quantifier-free type.

IV.2 Strictly r-ary relations

Definition IV.2.1. Let R be an r-ary relation. Given a structure $\mathcal{M}$, we say that R is *strictly r-ary in $\mathcal{M}$* if whenever $\mathcal{M} \models R(a_1, \ldots, a_r)$, then all the a_i are distinct. We say that a theory T enforces that R is *strictly r-ary* if in every model $\mathcal{M} \models T$, R is strictly r-ary in $\mathcal{M}$.

Given a formula $\varphi(x_1, \ldots, x_k)$ with k free variables, an ordered partition $\mathcal{I} = (I_1, \ldots, I_\ell)$ of $[k]$, and variables or elements $a_1, \ldots, a_\ell$, let $\varphi(a_{I_1}, \ldots, a_{I_\ell})$ denote the formula obtained from φ by replacing x_j with a_i whenever $j \in I_i$.

Lemma IV.2.2. *Let φ be an $\mathbf{FO}^k$ sentence in a weighted language $(\mathcal{L}, w)$ that has maximum arity r. Then there are:*

- *a weighted language $(\mathcal{L}', w')$ with maximum arity $\min(r, k)$,*
- *a universal $\mathbf{FO}^k$ $\mathcal{L}'$-sentence φ',*
- *a polynomial-time computable function $c : \mathbb{N} \to \mathbb{N}$ with $c(n) = 2^{O(n^r)}$, and*
- *a $c(n)$-to-one function F that takes an $\mathcal{L}$-structure to an $\mathcal{L}'$-structure with the same domain,*

such that

(1) for each $R' \in \mathcal{L}'$ with arity r', φ' enforces that R' is strictly r'-ary.

(2) for any $\mathcal{L}'$-structure $\mathcal{M}'$, the total weight of the preimage $F^{-1}(\mathcal{M}')$ can be computed in polynomial time from the weight of $\mathcal{M}'$, and

(3) $\mathcal{M} \models \varphi$ if and only if $F(\mathcal{M}) \models \varphi'$.

Proof. First, by Theorem II.5.1, we may assume that φ is a universal sentence. Moreover, we may write φ as a universally quantified full DNF; i.e.,

$$\varphi = \forall x_1 \cdots \forall x_k \bigvee_{j=1}^{\ell} \bigwedge_{i=1}^{m} \psi_i(x_1, \ldots, x_k)^{t_{ij}},$$

where ψ_i ranges over all atoms involving only variables $x_1, \ldots, x_k$, for some $\ell \in \mathbb{N}$ and truth values $t_{ij} \in \{0, 1\}$.

Consider a relation $R \in \mathcal{L}$ that has arity s. Let $1 \leq r' \leq \min(s, k)$, and consider an ordered partition of the indices $[s]$ into nonempty subsets $\mathcal{I} = (I_1, \ldots, I_{r'})$. Add a new r'-ary relation symbol $R^{\mathcal{I}}$ to $\mathcal{L}'$ and give it weight $w'(R^{\mathcal{I}}) = w(R)$. Doing this for every relation in $\mathcal{L}$ and every partition of its indices defines the weighted language $(\mathcal{L}', w')$.

We now define φ' as follows. Notice that every atom $\psi_i(x_1, \ldots, x_k)$ is of the form $R(x_{I_{i_1}}, \ldots, x_{I_{i_{r'}}})$ for some $R \in \mathcal{L}$ of arity s, partition $\mathcal{I} = \{I_1, \ldots, I_{r'}\}$ of $[s]$, and distinct values $i_j \in [k]$. Con-

struct φ' by first replacing each atom in φ with the corresponding $\mathcal{L}'$-atom $R^{\mathcal{I}}(x_{I_1}, \ldots, x_{I_{r'}})$. Then, for every relation $R' \in \mathcal{L}'$, if R' has arity r', we append to φ' a conjunct with

$$\forall x_1, \ldots, x_{r'} \left(R'(x_1, \ldots, x_{r'}) \to \bigwedge_{i \neq j} x_i \neq x_j \right).$$

Notice that this ensures that φ' enforces that R' is strictly r'-ary, so item (1) is satisfied.

Now, we define the function F. Given an $\mathcal{L}$-structure $\mathcal{M}$ with domain M, we define the $\mathcal{L}'$-structure $F(\mathcal{M})$ with domain M as follows: let $R^{\mathcal{I}} \in \mathcal{L}'$, and let r' be the arity of $R^{\mathcal{I}}$. Given distinct $a_1, \ldots, a_{r'} \in M$, set

$$F(\mathcal{M}) \models R^{\mathcal{I}}(a_1, \ldots, a_{r'}) \qquad \iff \qquad \mathcal{M} \models R(a_{I_1}, \ldots, a_{I_{r'}}),$$

and set $F(\mathcal{M}) \models \neg R^{\mathcal{I}}(\overline{a})$ whenever $\overline{a}$ is not a tuple of distinct elements. This completely specifies the behavior of every relation on every tuple, so the result is an $\mathcal{L}'$-structure.

For any $\mathcal{L}'$-structure $\mathcal{M}'$ with domain M, every element $\mathcal{M}$ of the preimage $F^{-1}(\mathcal{M}')$ can be uniquely specified in the following way:

1. for each $R \in \mathcal{L}$, partition $\mathcal{I} = (I_1, \ldots, I_{r'})$ of the variables of R with r' at most k and the arity of R, and distinct $a_1, \ldots, a_{r'} \in M$, set $\mathcal{M} \models R(a_{I_1}, \ldots, a_{I_{r'}})$ if and only if $\mathcal{M}' \models R^{\mathcal{I}}(a_1, \ldots, a_{r'})$;
2. for each $R \in \mathcal{L}$ with arity $r' > k$ and tuple $\overline{a} \in [n]^{r'}$ with more than k distinct elements, make an arbitrary choice for whether or not $\mathcal{M} \models R(\overline{a})$.

Item 1 gives a single choice. In item 2, for a given $R \in \mathcal{L}$ of arity r', the number of choices that need to be made is equal to the number of r'-tuples from $[n]$ with more than k distinct elements. There are $n^{r'}$ total r'-tuples from $[n]$, and there are $\binom{n}{j} j^{r'}$ total r'-tuples that involve up to j distinct elements. By an inclusion-exclusion argument, there are $n^{r'} - \sum_{j=0}^{k-1} (-1)^{k-j} \binom{n}{k-j} (k-j)^{r'}$ many r'-tuples from $[n]$ with more than k distinct elements, which can be calculated in polynomial time and is $\Theta(n^{r'})$. Summing over all relations $R \in \mathcal{L}$, the total number of choices in item 2 is $\Theta(n^r)$. Thus the function $c(n)$ which gives the number of elements of the preimage is $2^{\Theta(n^r)}$.

Now given an $\mathcal{L}'$-structure $\mathcal{M}'$ and structure $\mathcal{M} \in F^{-1}(\mathcal{M}')$, the weight of $\mathcal{M}$ is equal to the weight of $\mathcal{M}'$ multiplied by the weight of $\mathcal{L}$-relations on tuples of with more than k distinct elements. Since there is free choice for all relations on these tuples, the total weight of $F^{-1}(\mathcal{M}')$ is simply the sum over all ways to assign these relations, multiplied by the weight of $\mathcal{M}'$, which can be computed in polynomial time.

Finally, we show that $\mathcal{M} \models \varphi$ if and only if $F(\mathcal{M}) \models \varphi'$. Suppose that $\mathcal{M} \models \varphi$, and WLOG that the domain of $\mathcal{M}$ is $[n]$. φ' consists of two parts: one that is obtained by replacing $\mathcal{L}$-atoms in φ with the corresponding $\mathcal{L}'$-atom, and a conjunction with sentences enforcing that every $R' \in \mathcal{L}'$ is strictly r'-ary. By definition of $F(\mathcal{M})$, the second part is automatically satisfied. For the first part, let $R'_i(x_1, \ldots, x_{r'})$ be the $\mathcal{L}'$-atom corresponding to the atom ψ_i appearing in φ, and let $\overline{a} \in [n]^{r'}$. By definition of F, $\overline{a} \models \psi_i(\overline{a})$ in $\mathcal{M}$ if and only if $\overline{a} \models R'_i(\overline{a})$ in $\mathcal{M}'$, so since the first part of φ' is obtained from φ by replacing all the atoms ψ_i with the corresponding R'_i, we can conclude that $\mathcal{M}' \models \varphi'$.

On the other hand, suppose that $F(\mathcal{M}) \models \varphi'$. Then by the definition of F, $\mathcal{M}$ must have all the correct behavior on all ψ_i. Since φ makes no restrictions on truth values of atoms involving more than k variables, we can conclude that $\mathcal{M} \models \varphi$. □

IV.3 Weighted model counting for exponential growth rate classes

In this section, we prove a formula for the weighted first-order model count of theories whose Skolemization falls into the exponential growth rate for hereditary properties. Laskowski and Terry (36) prove that this growth rate is characterized by a very strong structural condition. We analyze this structural result in order to obtain an explicit formula for the weighted model count of the property.

Given an $\mathcal{L}$-structure $\mathcal{M}$ and $a, b \in M$, define $a \sim b$ if and only if $\text{qftp}(ab/(M \setminus \{a, b\}) = \text{qftp}(ba/(M \setminus \{b, a\})$; that is, for every quantifier-free formula $\varphi(x_1, \ldots, x_s)$ and $m_2, \ldots, m_s \in M \setminus \{a, b\}$,

$$\mathcal{M} \models \Big(\varphi(a, b, m_3, \ldots, m_n) \leftrightarrow \varphi(b, a, m_3, \ldots, m_n)\Big) \wedge \Big(\varphi(a, m_2, \ldots, m_s) \leftrightarrow \varphi(b, m_2, \ldots, m_s)\Big).$$

As an example, if $\mathcal{M}$ is a graph, then $a \sim b$ if and only if the neighborhoods of a and b, aside a and b themselves, are the same. It is straightforward to check that $\sim$ is an equivalence relation. Let $(a_1, \ldots, a_s), (b_1, \ldots, b_s) \in M^{(s)}$ such that $a_i \sim b_i$ for each $i \in [s]$. It is once again straightforward to check that $\text{qftp}(\overline{a}) = \text{qftp}(\overline{b})$, i.e., the type of a tuple only depends on which $\sim$-classes of its elements.

Definition IV.3.1. A hereditary $\mathcal{L}$-property $\mathcal{H}$ is *basic* if there is $k \in \mathbb{N}$ such that every $\mathcal{M} \in \mathcal{H}$ has at most k $\sim$-classes.

In the proof of (36, Theorem 1.4), Laskowski and Terry implicitly show that a hereditary property falls into the slowest growth rate of Theorem II.6.3 if and only if it is basic, and so we will conflate the two conditions.

For the following definitions, let $\mathcal{H}$ be a basic hereditary property and fix a countable model $\mathcal{M} \models T_{\mathcal{H}}$. Since $\mathcal{H}$ is basic, $\mathcal{M}$ has finitely many $\sim$-classes; enumerate them as $A_1, \ldots, A_k$ such that $0 < |A_1| \leq \cdots \leq |A_k|$—call this sequence the *canonical decomposition* of $\mathcal{M}$.

Now, let $t = \max\{i \in [k] \mid A_i \text{ is finite}\}$ and $K = \max\{r, |A_t|\}$. Given any set X, let $\Omega^{\mathcal{M}}(X)$ denote the set of ordered partitions $(X_1, \ldots, X_k)$ of X such that:

(i) for each $i \in [t]$, $|X_i| = |A_i|$, and

(ii) for each $t < i \leq k$, $|X_i| > K$.

Notice that $(A_1, \ldots, A_k) \in \Omega^{\mathcal{M}}(M)$.

Definition IV.3.2. (36, Definition 2.5) Let $\mathcal{N}$ be an $\mathcal{L}$-structure. We say that $\mathcal{N}$ is *compatible* with $\mathcal{M}$, and conversely that $\mathcal{M}$ is a *template* for $\mathcal{N}$, if there is a partition $(B_1, \ldots, B_k) \in \Omega^{\mathcal{M}}(N)$ such that for every tuple $(b_1, \ldots, b_s) \in N^{(s)}$ with $b_j \in B_{i_j}$ for each $j \in [s]$, then $\mathrm{qftp}(\overline{b}) = \mathrm{qftp}(\overline{a})$ for any (equivalently, every) $\overline{a} \in M^{(s)}$ such that $a_j \in A_{i_j}$ for each $j \in [s]$.

Suppose that $\mathcal{N}$ is compatible with $\mathcal{M}$ as witnessed by the partition $(B_1, \ldots, B_k) \in \Omega^{\mathcal{M}}(N)$ Then $B_1, \ldots, B_k$ are the $\sim$-classes of $\mathcal{N}$, and furthermore, $\mathcal{N}$ is isomorphic to a substructure of $\mathcal{M}$, so $\mathcal{N} \in \mathcal{H}$.

Let $\boldsymbol{\ell} \in [k]^s$ for some $s \in \mathbb{N}$. We say that a tuple $\overline{a} \in N^s$ has $\mathcal{M}$-*compatibility type* $\boldsymbol{\ell}$ if $a_i \in B_{\ell_i}$ for each $i \in [s]$. Since the B_i's are the $\sim$-classes of $\mathcal{N}$, if $\overline{a}$ and $\overline{b}$ have the same $\mathcal{M}$-compatibility type, then $a_i \sim b_i$ for each $i \in [s]$, and in particular $\mathrm{qftp}(\overline{a}) = \mathrm{qftp}(\overline{b})$ as noted previously.

The following lemmas, due to Laskowski and Terry, shows that compatibility is first-order definable and that compatibility with two templates (and hence by induction, any finite number) can be reduced to compatibility with a single template.

Lemma IV.3.3. *(36, Lemma 2.6) Let $\mathcal{H}$ be a basic hereditary property. There is a sentence $\theta_{\mathcal{M}}$ such that for any $\mathcal{L}$-structure $\mathcal{N}$, $\mathcal{N} \models \theta_{\mathcal{M}}$ if and only if $\mathcal{N}$ is compatible with $\mathcal{M}$.*

Lemma IV.3.4. *(36, Lemma 2.13) Let $\mathcal{H}$ be a basic hereditary property. Suppose that $\mathcal{M}_1, \mathcal{M}_2 \models T_{\mathcal{H}}$ are countably infinite, and that $\theta_{\mathcal{M}_1} \wedge \theta_{\mathcal{M}_2}$ is satisfiable. Then there is $i \in \{1, 2\}$ such that $\theta_{\mathcal{M}_1} \wedge \theta_{\mathcal{M}_2} \equiv \theta_{\mathcal{M}_i}$.*

For a countably infinite $\mathcal{M} \models T_{\mathcal{H}}$, set $\mathcal{H}(\mathcal{M}) := \{\mathcal{N} \mid \mathcal{N} \models \theta_{\mathcal{M}}\}$. The next result follows from the previous lemmas and a portion of the proof of (36, Theorem 2.14), and states that there are finitely many templates such that $\mathcal{H}$ can be decomposed into the classes of structures compatible with one of these templates.

Lemma IV.3.5. *Let $\mathcal{H}$ be a basic hereditary property. For sufficiently large n, there are countably infinite $\mathcal{M}_1, \ldots, \mathcal{M}_k \models T_{\mathcal{H}}$ such that the collection $\{\mathcal{H}(\mathcal{M}_1), \ldots, \mathcal{H}(\mathcal{M}_k)\}$ is pairwise disjoint and*

$$\mathcal{H}_n = \bigsqcup_{i=1}^{k} \mathcal{H}(\mathcal{M}_i)_n.$$

Proof. Every countably infinite model of $T_{\mathcal{H}}$ must be compatible with itself. Hence the following set of sentences is inconsistent:

$$T_{\mathcal{H}} \cup \{\neg\theta_{\mathcal{M}} \mid \mathcal{M} \models T_{\mathcal{H}},\ \mathcal{M} \text{ countably infinite}\} \cup \{\exists x_1 \dots \exists x_n \bigwedge_{i \neq j} x_i \neq x_j \mid n \geq 1\}.$$

Then by compactness, there are finitely many $\theta_{\mathcal{M}_1}, \dots, \theta_{\mathcal{M}_k}$ such that for sufficiently large n, for any $\mathcal{N} \in \mathcal{H}_n$, $\mathcal{N} \models \bigvee_{i=1}^{k} \theta_{\mathcal{M}_i}$. It immediately follows that $\mathcal{H}_n \subseteq \bigcup_{i=1}^{k} \mathcal{H}(\mathcal{M}_i)$. Additionally, since any finite model compatible with $\mathcal{M}_i$ must also be in $\mathcal{H}$, we have that in fact $\mathcal{H}_n = \bigcup_{i=1}^{k} \mathcal{H}(\mathcal{M}_i)$. Notice that *a priori*, the properties $\mathcal{H}(\mathcal{M}_i)$ may not be disjoint. However, we can easily force these properties to be disjoint.

Let $1 \leq i, j \leq k$. If $\theta_{\mathcal{M}_i} \wedge \theta_{\mathcal{M}_j}$ is satisfiable, then by Lemma IV.3.4, (without loss of generality) $\theta_{\mathcal{M}_i} \models \theta_{\mathcal{M}_j}$. Thus $\mathcal{H}(\mathcal{M}_i) \subseteq \mathcal{H}(\mathcal{M}_j)$, and we can remove $\mathcal{H}(\mathcal{M}_i)$ from our collection of properties. Otherwise, $\theta_{\mathcal{M}_i} \wedge \theta_{\mathcal{M}_j}$ is not satisfiable, and therefore $\mathcal{H}(\mathcal{M}_i)$ and $\mathcal{H}(\mathcal{M}_j)$ are already disjoint. So we may indeed assume that the properties are all disjoint, and conclude that

$$\mathcal{H}_n = \bigsqcup_{i=1}^{k} \mathcal{H}(\mathcal{M}_i)_n.$$

□

As a consequence, in order to find the weighted model count of a basic hereditary property $\mathcal{H}$, it suffices to find the weighted model count of $\mathcal{H}(\mathcal{M})_n$ for a finite collection of countably infinite models $\mathcal{M} \models T_{\mathcal{H}}$.

Proposition IV.3.6. *Let $\mathcal{H}$ be a basic hereditary property, and suppose that $\mathcal{M} \models T_{\mathcal{H}}$ is countably infinite and has k $\sim$-classes. Then there are $t, K \in \mathbb{N}$ and $\boldsymbol{c} \in \mathbb{N}^t$ such that*

$$\mathsf{WFOMC}(\mathcal{H}(\mathcal{M}), n, w) = \sum_{\substack{d_{t+1},\ldots,d_k > K \\ \sum c_i + \sum d_j = n}} \binom{n}{\boldsymbol{c}, \boldsymbol{d}} \prod_{R \in \mathcal{L}} w(R)^{p_{R,\boldsymbol{d}}(n)},$$

where $p_{R,\boldsymbol{d}}(n)$ is a polynomial depending only on R and $\boldsymbol{d}$ of degree at most the arity of R.

Proof. Let $A_1, \ldots, A_k$ be the canonical decomposition of $\mathcal{M}$, with $t = \max\{i \in [k] \mid A_i \text{ is finite}\}$ and $K = |A_t|$. For $i \in [t]$, let $c_i = |A_i|$. Suppose that $\mathcal{N}$ is compatible with $\mathcal{M}$, witnessed by partition $(B_1, \ldots, B_k) \in \Omega^{\mathcal{M}}(N)$. For $t+1 \leq i \leq k$, set $d_i := |B_i|$.

To find $w(\mathcal{N})$, it suffices to find the weight contributed by each relation $R \in \mathcal{L}$ and take the product over all relations. Fix $R \in \mathcal{L}$, and let s be the arity of R. Since for every $\boldsymbol{\ell} \in [k]^s$, every tuple with $\mathcal{M}$-compatibility type $\boldsymbol{\ell}$ has the same type, we can define

$$C_R := \{\boldsymbol{\ell} \in [k]^s \mid R \text{ holds on some (every) } \overline{a} \in N^s \text{ with } \mathcal{M}\text{-compatibility type } \boldsymbol{\ell}\}.$$

Every tuple $\overline{a} \in N^s$ some $\mathcal{M}$-compatibility type, and R will hold on $\overline{a}$ if and only if its $\mathcal{M}$-compatibility type is in C_R, so if we let $n_{\boldsymbol{\ell}}$ denote the number of tuples that have $\mathcal{M}$-compatibility type $\boldsymbol{\ell}$, then the weight contributed by R can be written as

$$\prod_{\boldsymbol{\ell} \in C_R} w(R)^{n_{\boldsymbol{\ell}}}$$

The exponent of $w(R)$ in the above expression can be written as

$$\sum_{\boldsymbol{\ell} \in C_R} n_{\boldsymbol{\ell}},$$

which is what we will analyze to keep the notation simple. Fix $\boldsymbol{\ell} \in [k]^s$. For simplicity of notation, let $b_i = c_i$ if $i \in [t]$, and let $b_i = d_i$ if $t+1 \leq i \leq k$. Tuples with $\mathcal{M}$-compatibility type $\boldsymbol{\ell}$ are such that the i-th coordinate is in B_{ℓ_i}, which has size b_{ℓ_i}, and so $n_{\boldsymbol{\ell}} = \prod_{i=1}^{s} b_{\ell_i}$. If $\ell_i \leq t$, then $b_{\ell_i} = c_{\ell_i}$ is a constant, and if $\ell_i > t$, then we can rewrite $d_{\ell_i} = n - d'_{\ell_i}$ for some $d'_{\ell_i} \in [n]$. Thus $n_{\boldsymbol{\ell}}$ can be written as a polynomial in n of degree at most s that only depends on the values $d_{t+1}, \ldots, d_k$. C_R has size at most k^s, which is a constant with respect to n, so summing over all $\boldsymbol{\ell} \in C_R$, we find that the exponent of $w(R)$ is also a polynomial of degree at most s depending only on $d_{t+1}, \ldots, d_k$, which we will denote $p_{R,\boldsymbol{d}}(n)$.

Taking the product over all relations, we find that

$$w(\mathcal{N}) = \prod_{R \in \mathcal{L}} w(R)^{p_{R,\boldsymbol{d}}(n)}.$$

Notice that this value depends only on $\boldsymbol{d}$, which is the sizes of the unbounded parts in the partition witnessing that $\mathcal{N}$ is compatible with $\mathcal{M}$, and not on any other information about $\mathcal{N}$.

Now, fix values $d_{t+1}, \ldots, d_k$ with $d_j > K$ for each j such that $\sum c_i + \sum d_j = n$. Any model $\mathcal{N} \in \mathcal{H}(\mathcal{M})_n$ with $\sim$-classes of sizes exactly $c_1, \ldots, c_t, d_{t+1}, \ldots, d_k$ will have weight $\prod_{R \in \mathcal{L}} w(R)^{p_{R,\boldsymbol{d}}(n)}$. Since a model in $\mathcal{H}(\mathcal{M})_n$ is exactly determined by the choice of which

elements of $[n]$ are in each $\sim$-class, there are $\binom{n}{\boldsymbol{c},\boldsymbol{d}}$ many ways to choose an $\mathcal{N}$ as above. Summing over all possible choices of the tuple $\boldsymbol{d}$, we find that

$$\mathsf{WFOMC}(\mathcal{H}(\mathcal{M}), n, w) = \sum_{\substack{d_{t+1},\ldots,d_k > K \\ \sum c_i + \sum d_j = n}} \binom{n}{\boldsymbol{c},\boldsymbol{d}} \prod_{R \in \mathcal{L}} w(R)^{p_{R,\boldsymbol{d}}(n)}.$$

□

IV.4 Weighted model counting for minimal fast-growth classes

In (15), Bollobás and Thomason give a characterization of "minimal" hereditary classes of graphs in the fastest growth rate. This characterization allows for efficient computation of the weighted model count of such minimal classes.

Definition IV.4.1. Let $0 \leq s \leq r$ be integers. A graph $G = (V, E)$ is called (r, s)-*colorable* if there exists a partition of V into $V_1, \ldots, V_r$ such that $V_1, \ldots, V_s$ are cliques, and $V_{s+1}, \ldots, V_r$ are independent sets.

Define

$$\mathcal{C}_n(r, s) := \{H \mid |H| = n \text{ and } H \text{ is } (r, s)\text{-colorable}\}$$

and

$$\mathcal{C}(r, s) := \bigcup_{n=1}^{\infty} C_n(r, s).$$

Definition IV.4.2. Given a hereditary property of graphs $\mathcal{H}$, define the *coloring number* of $\mathcal{H}$, denoted $r(\mathcal{H})$ to be

$$r(\mathcal{H}) := \max\{r \mid \text{for some } s,\ \mathcal{C}(r,s) \subseteq \mathcal{H}\}.$$

That is, $r(\mathcal{H})$ is the largest value of r such that $\mathcal{H}$ contains all (r,s)-colorable graphs for some value of s.

Theorem IV.4.3. *(15, Theorem 4) Let $\mathcal{H}$ be a non-trivial hereditary property of graphs, and let c_n be the sequence of numbers such that $|\mathcal{H}_n| = 2^{c_n\binom{n}{2}}$. Then*

$$\lim_{n\to\infty} c_n = 1 - \frac{1}{r(\mathcal{H})}.$$

This theorem leads to a characterization of minimal hereditary properties of graphs in the fastest class in the following sense: given integers $0 \leq r \leq s$, the class $C(r,s)$ is a hereditary property of graphs whose speed is $2^{\Theta(1-1/r)\binom{n}{2}}$, and any proper sub-hereditary property of $C(r,s)$ has speed at most $2^{\Theta(1-1/(r-1))\binom{n}{2}}$. In particular, if $r = 2$, any proper sub-hereditary property of $C(2,s)$ is not in the fastest growth rate. The three values for s of 0, 1, and 2 correspond to what are known in the graph theory literature as co-bipartite, split, and bipartite graphs, respectively. That is, a hereditary property of graphs is not in the fastest speed if and only if it omits a bi-partite graph, a co-bipartite graph, and a split graph.

To simplify the counting problem, we will consider *colored* graphs instead of colorable graphs. We work in the language $\mathcal{L}_{\mathrm{Gr}}^{(r,s)-\mathrm{col}}$ consisting of a binary relation E and unary predicates

$P_1, \dots P_r$, where the weights of each of the P_i's is 1. Consider the class $\mathcal{C}^{\text{col}}(r,s)$ of all $\mathcal{L}_{\text{Gr}}^{(r,s)-\text{col}}$-structures such that E is a graph relation, the sets $P_1, \dots, P_s$ are cliques, and $P_{s+1}, \dots P_r$ are cliques (i.e., (r,s)-colorable graphs where the coloring is given by the sets $P_1, \dots, P_s$). It is straightforward to check that this class is closed under isomorphism and substructure, so it is a hereditary property.

Theorem IV.4.4. *Let $0 \leq s \leq r$. $\mathsf{WFOMC}(\mathcal{C}^{\text{col}}(r,s), n, w)$ can be computed in polynomial time for any weight $w \in \mathbb{R}$ for the edge relation.*

Proof. We first consider how to construct a (r,s)-colored graph on the vertex set $[n]$. First, we must decide the partition of $[n]$. We can do this by first choosing a vector of non-negative integers $\boldsymbol{k} = (k_1, \dots, k_r)$ such that $\sum_{i=1}^{r} k_i = n$, which will be the sizes of the each of parts of the partition. Then, we can choose a $\boldsymbol{k}$-partition of $[n]$ into sets $P_1, \dots, P_r$. There are $\binom{n}{k_1,\dots,k_r}$ ways to do this. Once we have the partition, we set $P_1, \dots, P_s$ to be independent sets and $V_{s+1}, \dots, V_r$ to be cliques. On the other hand, for each pair of vertices in distinct parts P_j and P_k, we can freely choose whether or not it has an edge. There are $\sum_{0 \leq j < \ell \leq r} k_j k_\ell$ many such pairs of edges. For ease of notation, let $m_{\boldsymbol{k}}$ denote the quantity $\sum_{0 \leq j < \ell \leq r} k_j k_\ell$. Then to finalize the choice of a graph, we must choose a number i between 0 and $m_{\boldsymbol{k}}$, then choose a subset of i edges to include. The number of edges in this graph is

$$i + \sum_{j=s+1}^{r} \binom{k_j}{2},$$

and so the weight of this graph is

$$w^{i+\sum_{j=s+1}^{r}\binom{k_j}{2}} = w^i \prod_{j=s+1}^{r} w^{\binom{k_j}{2}}.$$

Summing over all the ways to construct a graph, we find that

$$\mathsf{WFOMC}(\mathcal{C}^{\text{col}}(r,s),n,w) = \sum_{k_1+\cdots k_r=n} \binom{n}{\boldsymbol{k}} \sum_{i=0}^{m_{\boldsymbol{k}}} \binom{m_{\boldsymbol{k}}}{i} w^i \prod_{j=s+1}^{r} w^{\binom{k_j}{2}}.$$

In the innermost product, each k_j is upper bounded by n, so the exponentiation can be done in time polynomial in n. The product is taken over a constant number $(r-s)$ of factors. Inside the inner summation, all values are upper bounded by n^2, so these operations are once again polynomial in n, and the sum is taken over a value that is $O(n^2)$. The multinomial coefficient $\binom{n}{\boldsymbol{k}}$ can be computed in polynomial time, and the outer sum is taken over fewer than n^r terms, so the entire expression can be computed in polynomial time.

□

IV.5 The FO^2 Case

In this section, we investigate the model counting problem in the restricted setting of **FO**^{2}.

We first isolate a property of **FO**^{2} that has been implicitly used in all prior analyses of the model counting problem, which essentially states that for models of an **FO**^{2} sentence, after conditioning on the partition of the domain into the 1-types, the 2-type of all pairs are

completely independent of each other. We also extend to the ordered case, in which conditioning on the 1-types as well as the ordering results in the 2-types being independent.

Lemma IV.5.1. *Let φ be a universal $\mathbf{FO}^2$ $\mathcal{L}$-sentence. Let $\mathcal{M}, \mathcal{N} \models \varphi$, both with domain $[n]$. Suppose that $a \neq b \in [n]$ such that $\mathrm{qftp}^{\mathcal{M}}(a) = \mathrm{qftp}^{\mathcal{M}}(b) = \mathrm{qftp}^{\mathcal{N}}(a) = \mathrm{qftp}^{\mathcal{N}}(b)$. Then the structure $\mathcal{M}'$ obtained from $\mathcal{M}$ by only changing the type of ab to be $\mathrm{qftp}^{\mathcal{N}}(ab)$ is also a model of φ.*

Furthermore, suppose that $\mathcal{L}$ contains a binary relation symbol $\leq$, that $\mathcal{M}$ and $\mathcal{N}$ additionally satisfy $LO(\leq)$, where $LO(\leq)$ are the axioms stating that $\leq$ is a linear order, and that $\mathcal{M}, \mathcal{N} \models a < b$. Then $\mathcal{M}' \models \varphi \wedge LO(\leq)$ as well.

Proof. First, convert φ into a universally quantified full DNF; i.e.,

$$\varphi = \forall x \forall y \bigvee_{k=1}^{\ell} \bigwedge_{j=1}^{m} \psi_j(x,y)^{t_{jk}},$$

where ψ_j ranges over all $\mathcal{L}$-atoms involving at most two variables, for some ℓ and truth values t_{ij}. In particular, we can write φ as

$$\varphi = \forall x \forall y \bigvee_{k=1}^{\ell} q_k(x,y),$$

where each q_k is a complete 2-type. Notice that for each $q_k(x,y)$, the type $q_k^{opp}(x,y)$ must also appear in φ, since if $q_k(a,b)$ holds in a model of φ, then $q_k^{opp}(b,a)$ holds as well.

Since $\mathcal{M}, \mathcal{N} \models \varphi$, there are values $1 \leq i, j \leq \ell$ such that $\mathcal{M} \models q_i(a,b)$ and $\mathcal{N} \models q_j(a,b)$. By assumption, q_i and q_j must both specify the same 1-types for x and y. Now, let $\mathcal{M}'$ be obtained

from $\mathcal{M}$ by only changing the type of ab to be q_j—in particular, the only change is on relations from a to b. To check that $\mathcal{M}' \models \varphi$, let $c, d \in [n]$. If $c = a$ and $d = b$, then $\mathcal{M}' \models q_j(c, d)$, and so $\mathcal{M}' \models \bigvee_{k=1}^{\ell} q_k(c, d)$. If $c = b$ and $d = a$, then the same reasoning applies, only with q_j^{opp}. Otherwise, at least one of $c \neq a$ and $d \neq b$ is true. Since $\mathcal{M} \models \varphi$, there is $1 \leq k \leq \ell$ such that $\mathcal{M} \models q_k(c, d)$. Notice that no 1-types were changed between $\mathcal{M}$ and $\mathcal{M}'$, so any unary atoms in q_k still hold of c and d in $\mathcal{M}'$. Also, the only binary atoms that differ between $\mathcal{M}$ and $\mathcal{M}'$ are from a to b (or vice versa), so since at least one of c and d is distinct from a and b, none of the values of binary atoms from c to d could have changed, and so $\mathcal{M}' \models p_k(c, d)$, and so $\mathcal{M}' \models \bigvee_{k=1}^{\ell} q_k(c, d)$. Therefore, we can conclude that $\mathcal{M}' \models \forall x \forall y \bigvee_{k=1}^{\ell} q_k(c, d)$, which is simply φ.

For the furthermore, we once again only need to consider whether or not $\mathcal{M}' \models \bigvee_{k=1}^{\ell} q_k(a, b)$. Since the information that $a < b$ is contained in q_i and q_j, the exact same reasoning works in this case as well. □

IV.5.1 Unweighted counting dichotomy for $\mathbf{FO}^2$

In this subsection, we utilize the proof technique for polynomial-time computability of WFOMC for $\mathbf{FO}^2$ sentences done by Beame, van den Broeck, Gribkoff, and Suciu (12) to prove a dichotomy theorem for the unweighted model count for universal $\mathbf{FO}^2$ formulas. This strengthens Theorem II.6.3 by stating that for universal $\mathbf{FO}^2$ sentences, the speed of the associated hereditary property must fall into either the slowest or fastest growth rates. Along the

way, we prove a formula for FOMC for $\mathbf{FO}^2$ sentences similar to (39, Theorem 1); however, our notation and terminology differs somewhat, so we will write out all the details for clarity.

Theorem IV.5.2. *Let φ be a universal $\mathbf{FO}^2$ sentence. Then $\mathsf{FOMC}(\varphi, n)$ is either in the slowest growth class (i.e., of the form $\sum_{i=1}^{k} p_i(n) i^n$ for sufficiently large n, where the p_i's are rational polynomials), or $\mathsf{FOMC}(\varphi, n)$ is in the fastest growth class (i.e., $\mathsf{FOMC}(\varphi, n) = 2^{Cn^r + o(n^r)}$ for sufficiently large n and some constant C, where r is the maximum arity of $\mathcal{L}$).*

Proof. First, consider the case that $r > 2$. By Lemma IV.2.2, there is a language $\mathcal{L}'$, universal $\mathbf{FO}^2$ $\mathcal{L}$-sentence, and function $c : \mathbb{N} \to \mathbb{N}$ with $c(n) = 2^{\Theta(n^r)}$ such that there is a $c(n)$-to-one function F from $\mathcal{L}$-structures with domain $[n]$ to $\mathcal{L}'$-structures with domain $[n]$. Automatically this implies that $\mathsf{FOMC}(\varphi, n)$ is at least $2^{\Omega(n^r)}$ and so in the fastest growth class.

Next, we can consider the case that $r = 2$. We may assume that φ is written as a universally quantified full DNF; i.e.,

$$\varphi = \forall x \forall y \bigvee_{k=1}^{\ell} \bigwedge_{j=1}^{m} \rho_j(x, y)^{t_{jk}},$$

where ρ_j ranges over all $\mathcal{L}$-atoms involving at most two variables, for some ℓ and values $t_{ij} \in \{0, 1\}$. In particular, we can write φ as

$$\varphi = \forall x \forall y \bigvee_{k=1}^{\ell} q_k(x, y),$$

where each q_k is a complete 2-type.

Fix n. In order to construct a model of φ with domain $[n]$, we can first decide the 1-type of each element of $[n]$. Let $p_1, \ldots, p_m$ denote the possible 1-types that can be formed in $\mathcal{L}$. If

we assume that $p_i(a)$ and $p_j(b)$ hold for some $a, b \in [n]$ and $1 \leq i, j \leq m$, this rules out certain possible 2-types for ab from the entire collection of q_k's. Let

$$I_{ij} := \{k \in [\ell] \mid q_k(x, y) \text{ is consistent with } p_i(x) \wedge p_j(y)\},$$

and let

$$t_{ij} := |I_{ij}|$$

(note that it is possible that I_{ij} is empty, so $t_{ij} = 0$). With this notation, we can say that the pair (a, b) must satisfy $\bigvee_{k \in I_{ij}} q_k(a, b)$. That is, the 2-type of ab is chosen from the values q_k where $k \in I_{ij}$. By Lemma IV.5.1, these choices are all independent. Thus, to count the number of models of φ, we simply need to find the total number of ways to make all of the above choices. First, deciding the 1-type of each element is equivalent to choosing an ordered partition of $[n]$ into m parts. This is counted by choosing a tuple $\boldsymbol{k} = (k_1, \ldots, k_m)$, then choosing a $\boldsymbol{k}$-partition $(C_1, \ldots, C_m)$ of $[n]$, which can be expressed as

$$\sum_{\sum k_i = n} \binom{n}{\boldsymbol{k}}.$$

For convenience of notation, for every $1 \leq i \leq j \leq m$, let

$$k_{ij} = \begin{cases} k_i k_j & \text{if } i \neq j \\ \binom{k_i}{2} & \text{if } i = j \end{cases},$$

i.e., k_{ij} is the number of pairs between elements in C_i and C_j.

Now for each $1 \leq i \leq j \leq m$, we must choose the 2-type of all pairs ab where $a \in C_i$ and $b \in C_j$. For each pair, there are t_{ij}-many choices for the 2-type, which are all independent, and there are k_{ij}-many pairs. Thus for fixed i, j, there are $t_{ij}^{k_{ij}}$ possible choices.

Therefore, the formula for $\mathsf{FOMC}(\varphi, n)$ is

$$\sum_{\sum k_i = n} \binom{n}{\boldsymbol{k}} \prod_{1 \leq i \leq j \leq m} t_{ij}^{k_{ij}}$$

We will condition on the values of t_{ij} to compute the model count.

First, suppose that for some values $i < j$, $t_{ij} > 1$, $t_{ii} \neq 0$, and $t_{jj} \neq 0$. Then if we choose $n_i = n_j = \frac{n}{2}$ and let all other values $n_{i'}$ be 0, this gives a term in the sum of the form

$$\binom{n}{\frac{n}{2}} t_{ii}^{\binom{n/2}{2}} t_{jj}^{\binom{n/2}{2}} t_{ij}^{n^2/4}.$$

Notice that this term will be at least $t_{ij}^{n^2/4} = 2^{Cn^2}$ for some constant C. Hence the model count will fall into the fastest growth class.

Alternatively, suppose that for some value i, $t_{ii} > 1$. Then if we choose $n_i = n$ and let all other values $n_{i'}$ be 0, then the formula for $\mathsf{FOMC}(\varphi, n)$ will contain a term of the form

$$\binom{n}{n} t_{ii}^{\binom{n}{2}}$$

which is at least 2^{Cn^2} for some constant C. So again the model count will fall into the fastest growth class.

If neither of the above cases hold, then for all i, either $t_{ii} = 0$ or $t_{ii} = 1$, and if $t_{ii} = t_{jj} = 1$, then $t_{ij} = 1$ as well. Thus every non-zero term of the formula for $\mathsf{FOMC}(\varphi, n)$ will be of the form

$$\binom{n}{\boldsymbol{k}},$$

and so $\mathsf{FOMC}(\varphi, n)$ is upper bounded by

$$\sum_{\sum k_i = n} \binom{n}{\boldsymbol{k}} = m^n = 2^{O(n)}.$$

Therefore, the model count falls into the slowest growth class. □

IV.5.2 Weighted model counting for $\mathbf{FO}^2$

In the previous subsection, we showed that for univeral $\mathbf{FO}^2$ sentences, there is a sharp dichotomy in the growth rate of the unweighted model count, and hence a dichotomy in the structural conditions on the hereditary properties definable by such sentences. In particular, suppose that we are dealing with only a single graph relation E. Then the only statements expressible in $\mathbf{FO}^2$ are 1) $\forall x \forall y\ E(x, y)$, 2) $\forall x \forall y\ \neg E(x, y)$, and 3) $\forall x \forall y (E(x, y) \vee \neg E(x, y))$. That is, the hereditary property must be 1) only complete graphs, 2) only empty graphs, or 3) all graphs. All three of these hereditary properties have extremely easy-to-compute weighted model counting problem. Extending this idea, we give an alternate proof of polynomial-time computability of WFOMC for $\mathbf{FO}^2$ sentences by converting any $\mathbf{FO}^2$ sentence into one involving

only unary relations and binary relations that are enforced to be symmetric and irreflexive. This allows for a more precise version of the formula for the weighted model count.

By Lemma IV.2.2, we may assume that we are working in a language with only unary and binary relations. The first step is to reduce the problem to one where every binary relation is a graph relation (irreflexive and asymmetric). To do this, we will need to use a linear order relation. While the axioms for a linear order are not expressible in $\mathbf{FO}^2$, we will see later how to introduce a linear order in a way that preserves the computational complexity of the weighted model counting problem. We will say that a structure $\mathcal{M}$ is *irreflexive* or *symmetric* if in $\mathcal{M}$, every binary relation is irreflexive or symmetric, respectively, except for the relation $\leq$ if it is in $\mathcal{L}$.

Lemma IV.5.3. *Let $(\mathcal{L}, w)$ be a weighted language of maximum arity 2 that contains a distinguished binary relation symbol $\leq$ of weight 1, and let ψ be a sentence of the form $LO(\leq) \wedge \varphi$, where φ is $\mathbf{FO}^2$ and enforces that every binary relation is strictly binary. There is a weighted language $(\mathcal{L}^{sym}, w^{sym})$ of maximum arity 2 which contains $\leq$ and an $\mathcal{L}^{sym}$-sentence $\psi^{sym} = LO(\leq) \wedge \varphi^{sym}$ with φ^{sym} in $\mathbf{FO}^2$ such that:*

1. *every model of ψ^{sym} is irreflexive and symmetric, and*
2. *there is a bijection F taking irreflexive and symmetric $\mathcal{L}^{sym}$-structures which are totally ordered by $\leq$ to irreflexive $\mathcal{L}$-structures which are totally ordered by $\leq$, such that for any $\mathcal{L}^{sym}$-structure $\mathcal{M}^{sym}$,*
   (a) *$F(\mathcal{M}^{sym})$ has the same domain as $\mathcal{M}^{sym}$, and $\leq^{\mathcal{M}^{sym}} = \leq^{F(\mathcal{M}^{sym})}$,*

(b) $\mathcal{M}^{sym} \models \psi^{sym} \iff F(\mathcal{M}^{sym}) \models \psi$, *and*

(c) $w^{sym}(\mathcal{M}^{sym}) = w(F(\mathcal{M}^{sym}))$.

In particular, $\mathsf{WFOMC}(\psi, w, n) = \mathsf{WFOMC}(\psi^{sym}, w^{sym}, n)$.

Proof. By Theorem II.5.1, we may assume that φ is a universal sentence. Write φ as a universally quantified full DNF, i.e.,

$$\varphi = \forall x \forall y \bigvee_{k=1}^{\ell} q_k(x, y),$$

where each q_k is a complete 2-type. By factoring out the atoms $x < y$, $x = y$, and $x > y$, we can write φ as

$$\begin{aligned}\varphi = \forall x \forall y &\left((x < y) \to \bigvee_{k=1}^{\ell_2} q_k(x, y) \right) \\ &\wedge \left((x > y) \to \bigvee_{k=1}^{\ell_2'} q_k'(x, y) \right) \\ &\wedge \left((x = y) \to \bigvee_{j=1}^{\ell_1} p_j(x) \right),\end{aligned}$$

where now each q_k and q_k' is a 2-type involving all non-order atoms, and each p_j is a complete 1-type. It must be the case that $\ell_2 = \ell_2'$ and $\{q_k' \mid k \in [\ell_2]\} = \{q_k^{opp} \mid k \in [\ell_2]\}$. To see this, for each $k \in [\ell_2]$, there must be some model of ψ and elements $a < b$ of that model such that $ab \models q_k(x, y)$. Since $b < a$, there must be $k' \in [\ell_2']$ such that $ba \models q_{k'}'(x, y)$. However, $\text{qftp}(ba) = \text{qftp}(ab)^{opp}$, which implies that $q_{k'}' = q_k^{opp}$. A similar argument works in the other direction, so every type q_k is the opposite of some other type $q_{k'}'$ and vice versa.

Let $R_1, \ldots, R_m$ enumerate the binary relations of $\mathcal{L}$ other than $\leq$, and expand out each q_k as $q_k = p_{k_x}(x) \wedge p_{k_y}(y) \wedge \bigwedge_{i=1}^{m} \left(R_i(x,y)^{t_{ik}^{xy}} \wedge R_i(y,x)^{t_{ik}^{yx}} \right)$, for some values $k_x, k_y \in [\ell_1]$ and $t_{ik}^{xy}, t_{ik}^{yx} \in \{0,1\}$. Then we can write ψ as

$$\begin{aligned} LO(\leq) \wedge \forall x \forall y \Bigg[& (x < y) \rightarrow \bigvee_{k=1}^{\ell_2} \left(p_{k_x}(x) \wedge p_{k_y}(y) \wedge \bigwedge_{i=1}^{m} \left(R_i(x,y)^{t_{ik}^{xy}} \wedge R_i(y,x)^{t_{ik}^{yx}} \right) \right) \Bigg] \\ & \Bigg[(x > y) \rightarrow \bigvee_{k=1}^{\ell_2} \left(p_{k_x}(y) \wedge p_{k_y}(x) \wedge \bigwedge_{i=1}^{m} \left(R_i(y,x)^{t_{ik}^{xy}} \wedge R_i(x,y)^{t_{ik}^{yx}} \right) \right) \Bigg] \\ \wedge & \left[(x = y) \rightarrow \bigvee_{j=1}^{\ell_1} p_j(x) \right], \end{aligned} \tag{IV.1}$$

with the second line coming from the fact that the types q'_k are obtained by interchanging x and y in the types q_k.

For each $i = 1, \ldots, m$, let $R_i^{<}$ and $R_i^{>}$ be new binary relation symbols. Let $\mathcal{L}^{\leq}$ consist of the unary relation symbols of $\mathcal{L}$, the relation symbols $R_i^{<}$ and $R_i^{>}$ for each i, and the symbol $\leq$.

Now, let ψ^{sym} be the following sentence:

$$\begin{aligned} LO(\leq) \wedge \forall x \forall y & \left[(x < y) \rightarrow \bigvee_{k=1}^{\ell_2} \left(p_{k_x}(x) \wedge p_{k_y}(y) \wedge \bigwedge_{i=1}^{m} \left(R_i^{<}(x,y)^{t_{ik}^{xy}} \wedge R_i^{>}(x,y)^{t_{ik}^{yx}} \right) \right) \right] \\ & \wedge \left[(x > y) \rightarrow \bigvee_{k=1}^{\ell_2} \left(p_{k_x}(y) \wedge p_{k_y}(x) \wedge \bigwedge_{i=1}^{m} \left(R_i^{>}(x,y)^{t_{ik}^{xy}} \wedge R_i^{<}(x,y)^{t_{ik}^{yx}} \right) \right) \right] \\ & \wedge \left[(x = y) \rightarrow \bigvee_{j=1}^{\ell_1} p_j(x) \right] \\ & \wedge \left[\bigwedge_{i=1}^{m} (R_i^{<}(x,y) \leftrightarrow R_i^{<}(y,x)) \wedge (R_i^{>}(x,y) \leftrightarrow R_i^{>}(y,x)) \right] \end{aligned} \tag{IV.2}$$

The first two lines of (Equation IV.2) state that $R_i^{<}$ and $R_i^{>}$ reflect the behavior of R_i going from lesser elements to greater elements and greater elements to lesser elements, respectively, while the third line maintains the behavior of ψ on unary relations, and the fourth line enforces that all binary relations are symmetric. Also, notice that in the first two lines, for greater uniformity, we change the order of the variables in the last atom from (y, x) to (x, y), which we may do since we enforce that all binary relations are symmetric. Since the 1-types $p_1, \ldots, p_j$ all state that binary relations (other than $\leq$) are irreflexive, ψ^{sym} also still enforces that every binary relation is irreflexive.

We note that it is not strictly neccesary to include the second lines in our forms for both ψ and ψ^{sym}, since the information contained in them is actually captured in the first lines already. However, including these lines will make some of the later analysis simpler.

To define F, let $\mathcal{M}^{sym}$ be an irreflexive and symmetric $\mathcal{L}^{sym}$-structure that is totally ordered by $\leq$. We define an $\mathcal{L}$-structure $\mathcal{M}$ on the same domain as $\mathcal{M}^{sym}$ with the same behavior

as $\mathcal{M}^{sym}$ for unary relations and $\leq$. For each relation $R_i \in \mathcal{L}$ and pair $a, b \in M$, we set $\mathcal{M} \models R_i(a,b)$ if and only if $\mathcal{M}^{sym} \models (a < b \wedge R_i^<(a,b)) \vee (a > b \wedge R_i^>(a,b))$. Set $F(\mathcal{M}^{sym}) = \mathcal{M}$. This operation is invertible: given an $\mathcal{L}$-structure $\mathcal{M}$, let $\mathcal{M}^{sym}$ be an $\mathcal{L}^{sym}$-structure on the same domain with the same behavior of unary predicates and $\leq$. Set $\mathcal{M}^{sym} \models R_i^<(a,b)$ if and only if $\mathcal{M} \models (a < b \wedge R_i(a,b)) \vee (a > b \wedge R_i(b,a))$. It is straightforward to check that this gives an inverse to F.

Now, suppose that $\mathcal{M}^{sym} \models \psi^{sym}$. Let $\mathcal{M} = F(\mathcal{M}^{sym})$. We must check that $\mathcal{M} \models \psi$. By definition, $\mathcal{M} \models LO(\leq)$. Let $a, b \in M$. If $a = b$, then since $\mathcal{M}^{sym} \models \bigvee_{j=1}^{\ell_1} p_j(x)$, it must be that $p_j(a)$ holds for some $j \in [\ell_1]$, and so a satisfies the third line of (Equation IV.1). Now suppose $a < b$. Since $\mathcal{M}^{sym} \models \psi^{sym}$, there is some $k \in [\ell_2]$ such that $\mathcal{M}^{sym} \models p_{k_x}(a) \wedge p_{k_y}(b) \wedge \bigwedge_{i=1}^{m} \left(R_i^<(a,b)^{t_{ik}^{xy}} \wedge R_i^>(a,b)^{t_{ik}^{yx}} \right)$. By tracing the definition of F, we can find that $\mathcal{M} \models p_{k_x}(a) \wedge p_{k_y}(b) \wedge \bigwedge_{i=1}^{m} \left(R_i(a,b)^{t_{ik}^{xy}} \wedge R_i(b,a)^{t_{ik}^{yx}} \right)$, and so ab satisfies the first line of (Equation IV.1). Since the the second line of (Equation IV.1) is the same as the first line with the roles of x and y swapped, the case that $a > b$ is also covered. Hence $\mathcal{M} \models \psi$. A similar argument shows that $\mathcal{M} \models \psi \Rightarrow \mathcal{M}^{sym} \models \psi^{sym}$.

Finally, we define w^{sym} and prove that condition 2(c) holds. Let w^{sym} take the same value as w on all unary relations, and set $w^{sym}(R_i^<) = w^{sym}(R_i^>) = \sqrt{w(R_i)}$ for each $i \in [m]$. Given an $\mathcal{L}^{sym}$-structure $\mathcal{M}^{sym}$, $\mathcal{M} = F(\mathcal{M}^{sym})$, and pair $a < b \in M$, $\mathcal{M} \models R_i(a,b)$ if and only if $\mathcal{M}^{sym} \models R_i^<(a,b) \wedge R_i^<(b,a)$. Hence $\mathcal{M} \models R_i(a,b)$ contributes a factor of $w(R_i)$ to $w(\mathcal{M})$ if and only if $\mathcal{M}^{sym} \models R_i^<(a,b) \wedge R_i^<(b,a)$ contributes a factor of $w(R_i^<)^2 = w(R_i)$ to $w^{sym}(\mathcal{M}^{sym})$. A similar argument works for $R_i^>$ when $a > b$, and since the two structures have the same behavior

on unary predicates and w, w^{sym} assign the same weights to unary predicates, we conclude that $w(\mathcal{M}) = w^{sym}(\mathcal{M}^{sym})$. □

Next, we introduce some tools for computing weighted sums of binary relations.

Definition IV.5.4. Let $(\mathcal{L}, w)$ be a weighted language, $s \in \mathbb{N}$ and $\phi(x, y)$ be a quantifier-free formula in two variables. Define $\mathsf{WFOMC}_2(\phi, s, w)$ to be the weighted sum of all ways to assign binary relations appearing in ϕ, on s pairs independently such that each pair ab satisfies $\phi(a, b)$.

Definition IV.5.5. Let $\phi(x, y)$ be a quantifier-free formula in two variables. Given a conjunction of binary literals q, construct a new formula as follows. For each positive literal $R(x, y)$ appearing in q, replace all instances of $R(x, y)$ in ϕ with $\top$, and for each negative literal $\neg R(x, y)$ appearing in q, replace all instances of $R(x, y)$ in ϕ with $\bot$. Denote this new formula as $\phi(x, y \mid q)$.

The previous definition can be considered a form of "conditioning" a formula on the information given in q. We'll illustrate these definitions with an example. Let $\mathcal{L} = \{R_1, R_2\}$, with $w(R_1) = 2,\ w(R_2) = 2$, and

$$\phi = (R_1(x, y) \vee R_2(x, y)) \wedge (R_1(x, y) \leftrightarrow R_1(y, x)) \wedge (R_2(x, y) \leftrightarrow R_2(y, x)).$$

Then

$$\phi(x, y \mid R_1(x, y)) \equiv R_2(x, y) \leftrightarrow R_2(y, x),$$

while

$$\phi(x, y, \mid \neg R_1(x, y)) \equiv R_2(x, y) \wedge R_2(y, x).$$

Furthermore, $\mathsf{WFOMC}_2(\phi(x, y, \mid R_1(x, y)), s, w) = 3^s$ since on all s pairs, R_2 can either hold or fail, so the total weight is $\sum_{t=0}^{s} \binom{s}{t} 2^t = 3^s$, while $\mathsf{WFOMC}_2(\phi(x, y, \mid \neg R_1(x, y)), s, w) = 2^s$, since on every pair, R_2 must hold. In both cases, the weight of R_1 is ignored since it has been eliminated when conditioning ϕ on the value of R_1.

Lemma IV.5.6. *Let $(\mathcal{L}, w)$ be a weighted language and let $\phi(x, y)$ be a quantifier-freeformula in two variables that enforces that every binary relation is symmetric (i.e., of the form $\phi'(x, y) \wedge \bigwedge_{R \in \mathcal{L}} R(x, y) \leftrightarrow R(y, x)$ for some quantifier-free ϕ'). Then there is a constant $w_\phi \in \mathbb{R}$ such that for any $s \in \mathbb{N}$, $\mathsf{WFOMC}_2(\phi, s, w) = w_\phi^s$. Furthermore, for fixed ϕ, w_ϕ can be computed in constant time, and thus $\mathsf{WFOMC}_2(\phi, s, w)$ can be computed in time polynomial in s.*

Proof. We proceed by induction on the number of relations that appear in ϕ.

Suppose that only one relation E appears in ϕ. Then by converting ϕ to a DNF, ϕ must be one of $E(x, y)$, $\neg E(x, y)$, and $E(x, y) \vee \neg E(x, y)$. Then $\mathsf{WFOMC}_2(\phi, s, w)$ is $w(E)^s$, 1, or $\sum_{t=0}^{s} \binom{s}{t} w(E)^t = (w(E) + 1)^s$ (by the binomial theorem), respectively. Hence we can take w_ϕ to be $w(E)$, 1, or $(w(E) + 1)$, depending on which case we are in.

Now, suppose that relations $E_1, \ldots, E_m$ appear in ϕ. If $\phi(x, y) \wedge E_m(x, y)$ is inconsistent, then we may replace all instances of E_m in ϕ with $\bot$, which gives a formula with one fewer relation, and so we may apply induction. Similarly, we may apply induction if $\phi(x, y) \wedge \neg E_m(x, y)$ is inconsistent by replacing all instances of E_m with $\top$. So we may assume that it is consistent with

ϕ for either E_m or $\neg E_m$ to hold on a pair. We consider all ways to choose assignments of the binary relations on the pairs, starting by choosing the pairs for which E_m holds. For any $s_m \in [s]$, there are $\binom{s}{s_m}$-many ways to choose s_m pairs on which E_m will hold. These pairs will contribute $w(E_m)^{s_m}$ weight to the assignment. Once we have chosen these pairs, we must choose the assignments for the remaining relations. For the s_m-many pairs on which E_m holds, the total weighted count of the assignments of remaining relations is given by $\mathsf{WFOMC}(\phi(x, y \mid E_m), s_m, w)$, while for the remaining pairs, the weighted count is given by $\mathsf{WFOMC}(\phi(x, y \mid \neg E_m), s - s_m, w)$. Notice that $\phi(x, y \mid E_m)$ and $\phi(x, y \mid \neg E_m)$ both involve one fewer relation than ϕ, so by induction, we can rewrite $\mathsf{WFOMC}(\phi(x, y \mid E_m), s_m, w) = \left(w_{\phi(x,y|E_m)}\right)^{s_m}$ and $\mathsf{WFOMC}(\phi(x, y \mid \neg E_m), s - s_m, w) = \left(w_{\phi(x,y,|\neg E_m)}\right)^{s-s_m}$. Since all assignments are independent between distinct pairs, the total weight of all assignments where E_m holds on the chosen s_m-many pairs is the products of all of these weights, i.e. $w(E_m)^{s_m} \left(w_{\phi(x,y|E_m)}\right)^{s_m} \left(w_{\phi(x,y,|\neg E_m)}\right)^{s-s_m}$. Summing over all choices of s_m and the ways to choose the s_m-many pairs for which E_m holds, we find that

$$\begin{aligned}
\mathsf{WFOMC}_2(\phi, s, w) &= \sum_{s_m}^{s} \binom{s}{s_m} w(E_m)^{s_m} \left(w_{\phi(x,y|E_m)}\right)^{s_m} \left(w_{\phi(x,y,|\neg E_m)}\right)^{s-s_m} \\
&= \left(w_{\phi(x,y|\neg E_m)}\right)^{s} \sum_{s_m}^{s} \binom{s}{s_m} \left(\frac{w(E_m) w_{\phi(x,y|E_m)}}{w_{\phi(x,y|\neg E_m}}\right)^{s_m} \\
&= \left(w_{\phi(x,y|\neg E_m)}\right)^{s} \left(\frac{w(E_m) w_{\phi(x,y|E_m)}}{w_{\phi(x,y|\neg E_m}} + 1\right)^{s} \\
&= \left(w(E_m) w_{\phi(x,y,|E_m)} + w_{\phi(x,y,|\neg E_m)}\right)^{s},
\end{aligned}$$

so we may take $w_\phi = w(E_m) w_{\phi(x,y,|E_m)} + w_{\phi(x,y,|\neg E_m)}$. □

Theorem IV.5.7. *Let φ be a* $\mathbf{FO}^2$ *sentence. Then* $\mathsf{WFOMC}(\varphi, n, w)$ *is computable in polynomial time.*

Proof. Let $\leq$ be a binary relation symbol not in $\mathcal{L}$. By Lemma IV.2.2, we may assume that φ enforces that every binary relation is irreflexive. Let $\mathcal{L}^{\leq} := \mathcal{L} \cup \{\leq\}$ and define $w^{\leq} : \mathcal{L}^{\leq} \to \mathbb{R}$ as follows: $w^{\leq}|_{\mathcal{L}} = w$ and $w^{\leq}(\leq) = 1$. Notice that the models of $\psi := \varphi \wedge LO(\leq)$ are exactly the models of φ with $\leq$ chosen to be an arbitrary ordering of the domain elements. Moreover, the weight of such a model is the same as its reduct to $\mathcal{L}$. Hence $\mathsf{WFOMC}(\psi, n, w^{\leq}) = \mathsf{WFOMC}(\varphi, n, w) \cdot n!$, and so it suffices to compute $\mathsf{WFOMC}(\psi, n, w^{\leq})$ in polynomial time.

We want to apply Lemma IV.5.3, but even more can be said in this case. Since φ does not use the symbol $\leq$, when writing ψ in the form (Equation IV.1), we may write it so that first two rows are identical except for $x < y$ and $x > y$. Hence, after applying the transformation in Lemma IV.5.3, we may assume that ψ takes the following form:

$$\begin{aligned} LO(\leq) \wedge \forall x \forall y \Bigg[& (x < y) \to \bigvee_{k=1}^{\ell_2} \left(p_{k_x}(x) \wedge p_{k_y}(y) \wedge \bigwedge_{i=1}^{m} \left(R_i^{<}(x,y)^{t_{ik}^{xy}} \wedge R_i^{>}(x,y)^{t_{ik}^{yx}} \right) \right) \Bigg] \\ \wedge & \left[(x > y) \to \bigvee_{k=1}^{\ell_2} \left(p_{k_x}(x) \wedge p_{k_y}(y) \wedge \bigwedge_{i=1}^{m} \left(R_i^{>}(x,y)^{t_{ik}^{xy}} \wedge R_i^{<}(x,y)^{t_{ik}^{yx}} \right) \right) \right] \\ \wedge & \left[(x = y) \to \bigvee_{j=1}^{\ell_1} p_j(x) \right] \\ \wedge & \left[\bigwedge_{i=1}^{m} (R_i^{<}(x,y) \leftrightarrow R_i^{<}(y,x)) \wedge (R_i^{>}(x,y) \leftrightarrow R_i^{>}(y,x)) \right], \end{aligned} \tag{IV.3}$$

with the difference between (Equation IV.3) and (Equation IV.2) being that in the second line, we no longer need to swap the role of x and y in the 1-types.

Now to compute $\mathsf{WFOMC}(\psi, n, w^{\leq})$, we first want to construct the models of ψ. By (55, Corollary 1), we may assume that the domain $[n]$ is ordered as $1 \leq 2 \leq \cdots \leq n$. The full weighted model count is then the value for this particular ordering multiplied by $n!$.

Let $\boldsymbol{k}$ be a tuple such that $\sum k_i = n$, and fix a $\boldsymbol{k}$-partition of $[n]$ into $C_1, \ldots, C_{\ell_1}$ so that every element of C_i is assigned type p_i. By Lemma IV.5.1, given the ordering and the partition $(C_1, \ldots, C_{\ell_1}$, the choices of 2-types for each pair are independent from one another, so it suffices to compute the weighted sum of assignments of binary relations that are consistent with ψ over every pair, and take the product of all these values.

Fix values $i < j \in [\ell_1]$, and let

$$I_{ij} := \{k \in [\ell_2] \mid k_x = i \wedge k_y = j\},$$

that is, I_{ij} is the collection of indices k for which $p_i(x)$ and $p_j(y)$ appear in the disjuncts in the first line of (Equation IV.2). The symmetry of the set of types $\{q_k \mid k \in [\ell_2]\}$ implies that $I_{ij} = I_{ji}$: if $k \in I_{ij}$, then $p_i(x), p_j(y) \in q_k$. Then Let $a \in C_i$ and $b \in C_j$. If $a < b$, then in any model of ψ,

$$(a, b) \models \bigvee_{k \in I_{ij}} \bigwedge_{i=1}^{m} \left(R_i^{<}(x, y)^{t_{ik}^{xy}} \wedge R_i^{>}(x, y)^{t_{ik}^{yx}} \right)$$

On the other hand, if $a > b$, then

$$(a, b) \models \bigvee_{k \in I_{ij}} \bigwedge_{i=1}^{m} \left(R_i^{>}(x, y)^{t_{ik}^{xy}} \wedge R_i^{<}(x, y)^{t_{ik}^{yx}} \right)$$

Since $R_i^<$ and $R_i^>$ have the same weight, the total weight in of all disjuncts in both of the above formulas are equal. Hence the weight of relations on ab only depends on the fact that $a \in C_i$ and $b \in C_j$, and not on the order. Therefore, without loss of generality we may assume that $a < b$, and so

$$(a,b) \models \bigvee_{k \in I_{ij}} \bigwedge_{i=1}^{m} \left(R_i^<(x,y)^{t_{ik}^{xy}} \wedge R_i^>(x,y)^{t_{ik}^{yx}} \right). \tag{IV.4}$$

At this point, it does not matter if a given binary relation is of the form $R_i^<$ or $R_i^>$, so re-label all the binary relations as $E_1, \ldots, E_{m'}$, which are all enforced to be irreflexive and asymmetric. Thus (Equation IV.4) can be rewritten as

$$(a,b) \models \bigvee_{k \in I_{ij}} \bigwedge_{i=1}^{m'} E_i(x,y)^{t_{ik}}, \tag{IV.5}$$

where each t_{ik} is a value in $\{0,1\}$. Let

$$\phi_{ij}(x,y) := \bigvee_{k \in I_{ij}} \bigwedge_{i=1}^{m'} E_i(x,y)^{t_{ik}},$$

and let

$$k_{ij} := \begin{cases} k_i k_j & \text{if } i \neq j \\ \binom{k_i}{2} & \text{if } i = j \end{cases},$$

i.e., the number of pairs between elements in C_i and C_j. Then the weighted sum of all assignments of the binary relations to pairs between C_i and C_j consistent with ψ is given by $\mathsf{WFOMC}_2(\phi_{ij}, k_{ij}, w)$, which by Lemma IV.5.6 is of the form $w_{\phi_{ij}}^{k_{ij}}$.

ByLemma IV.5.1, the assignments for varying values of i and j are independent, so the total weighted sum of all models of φ whose partition of 1-types is $C_1, \ldots, C_{\ell_1}$ is

$$\prod_{i=1}^{\ell_1} w(p_i)^{k_i} \cdot \prod_{1 \le i \le j \le \ell_1} w_{\phi_{ij}}^{k_{ij}},$$

where $w(p_i)$ is the product of the weights of all unary relations that hold in p_i. Since this value only depends on the size of the parts, and not on the parts themselves, the overall weighted model count can be found by

$$\sum_{\sum k_i = n} \binom{n}{\boldsymbol{k}} \prod_{i=1}^{\ell_1} w(p_i)^{k_i} \cdot \prod_{1 \le i \le j \le \ell_1} w_{\phi_{ij}}^{k_{ij}},$$

which, since $w(p_i)$ and $w_{\phi_{ij}}$ are constants not depending on n, is computable in polynomial time.

This gives the weighted model count of models of ψ where the ordering of the domain is $1 \le \cdots \le n$. As noted, $\mathsf{WFOMC}(\psi, n, w^{\le})$ is this value multiplied by $n!$. However, we also know that $\mathsf{WFOMC}(\psi, n, w^{\le}) = \mathsf{WFOMC}(\varphi, n, w) \cdot n!$, so $\mathsf{WFOMC}(\varphi, n, w)$ is also this same quantity and thus is computable in polynomial time. $\square$

CITED LITERATURE

1. Alon, N., Bun, M., Livni, R., Malliaris, M., and Moran, S.: Private and online learnability are equivalent. J. ACM, 69(4), aug 2022.

2. Angluin, D.: Learning regular sets from queries and counterexamples. Inf. Comput., 75:87–106, 1987.

3. Angluin, D. and Dohrn, T.: The power of random counterexamples. In Proceedings of the 28th International Conference on Algorithmic Learning Theory, eds, S. Hanneke and L. Reyzin, volume 76 of Proceedings of Machine Learning Research, pages 452–465. PMLR, 15–17 Oct 2017.

4. Angluin, D. and Fisman, D.: Learning regular omega languages. Theoretical Computer Science, 650:57–72, 2016. Algorithmic Learning Theory.

5. Balcázar, J. L., Castro, J., Guijarro, D., and Simon, H.-U.: The consistency dimension and distribution-dependent learning from queries. Theoretical Computer Science, 288(2):197–215, 2002. Algorithmic Learning Theory.

6. Balogh, J., Bollobás, B., and Morris, R.: Hereditary properties of combinatorial structures: Posets and oriented graphs. Journal of Graph Theory, 56, 2007.

7. Balogh, J., Bollobás, B., and Morris, R.: Hereditary properties of tournaments. Electron. J. Comb., 14, 2007.

8. Balogh, J., Bollobás, B., and Weinreich, D.: The speed of hereditary properties of graphs. Journal of Combinatorial Theory, Series B, 79(2):131–156, 2000.

9. Balogh, J., Bollobás, B., and Weinreich, D.: The penultimate rate of growth for graph properties. European Journal of Combinatorics, 22(3):277–289, 2001.

10. Balogh, J., Bollobás, B., and Weinreich, D.: A jump to the bell number for hereditary graph properties. Journal of Combinatorial Theory, Series B, 95(1):29–48, 2005.

11. Bárány, V.: A hierarchy of automatic ω-words having a decidable mso theory. RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications, 42(3):417–450, 2008.

12. Beame, P., Van den Broeck, G., Gribkoff, E., and Suciu, D.: Symmetric weighted first-order model counting. In Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS '15, page 313–328, New York, NY, USA, 2015. Association for Computing Machinery.

13. Bojańczyk, M., Klin, B., and Lasota, S.: Automata theory in nominal sets. Logical Methods in Computer Science, Volume 10, Issue 3, August 2014.

14. Bollig, B., Habermehl, P., Kern, C., and Leucker, M.: Angluin-style learning of nfa. In IJCAI, pages 1004–1009, 07 2009.

15. Bollobás, B. and Thomason, A.: Hereditary and Monotone Properties of Graphs, pages 70–78. Berlin, Heidelberg, Springer Berlin Heidelberg, 1997.

16. Bousquet, O., Hanneke, S., Moran, S., and Zhivotovskiy, N.: Proper learning, helly number, and an optimal svm bound. In Annual Conference Computational Learning Theory, 2020.

17. Carton, O. and Thomas, W.: The monadic theory of morphic infinite words and generalizations. In Mathematical Foundations of Computer Science 2000, eds, M. Nielsen and B. Rovan, pages 275–284, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

18. Chase, H. and Freitag, J.: Model theory and machine learning. The Bulletin of Symbolic Logic, 25(3):319–332, 2019.

19. Chase, H. and Freitag, J.: Bounds in query learning. In Proceedings of Thirty Third Conference on Learning Theory, eds, J. Abernethy and S. Agarwal, volume 125 of Proceedings of Machine Learning Research, pages 1142–1160. PMLR, 2020.

20. Chase, H. S.: Model Theory and Machine Learning. PhD thesis, University of Illinois Chicago, April 2020.

21. Dong, X. L., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., and Zhang, W.: Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August

24 - 27, 2014, pages 601–610, 2014. Evgeniy GabrilovichWilko HornNi LaoKevin MurphyThomas StrohmannShaohua SunWei ZhangGeremy Heitz.

22. Dotson, R. and Nagle, B.: Hereditary properties of hypergraphs. Journal of Combinatorial Theory, Series B, 99(2):460–473, 2009.

23. Drews, S. and D'Antoni, L.: Learning symbolic automata. In Tools and Algorithms for the Construction and Analysis of Systems, eds, A. Legay and T. Margaria, pages 173–189, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.

24. Elgot, C. C. and Rabin, M.: Decidability and undecidability of extensions of second (first) order theory of (generalized) successor. J. Symb. Log., 31:169–181, 1966.

25. Fisman, D. and Saadon, S.: Learning and characterizing fully-ordered lattice automata. In Automated Technology for Verification and Analysis, eds, A. Bouajjani, L. Holík, and Z. Wu, pages 266–282, Cham, 2022. Springer International Publishing.

26. Gabbay, M. J. and Pitts, A. M.: A new approach to abstract syntax with variable binding. Formal Aspects of Computing, 13(3):341–363, 07 2002.

27. Getoor, L. and Taskar, B.: Introduction to Statistical Relational Learning. The MIT Press, 2007.

28. Graedel, E., Kolaitis, P., and Vardi, M.: On the decision problem for two-variable first-order logic. Bulletin of Symbolic Logic, 3, 01 2001.

29. Gribkoff, E., Van den Broeck, G., and Suciu, D.: Understanding the complexity of lifted inference and asymmetric weighted model counting. ArXiv, abs/1405.3250, 2014.

30. Hanneke, S., Livni, R., and Moran, S.: Online learning with simple predictors and a combinatorial characterization of minimax in 0/1 games. In COLT, 2021.

31. Hellerstein, L., Pillaipakkamnatt, K., Raghavan, V., and Wilkins, D.: How many queries are needed to learn? J. ACM, 43(5):840–862, September 1996.

32. Kruckman, A., Rubin, S., Sheridan, J., and Zax, B.: A myhill-nerode theorem for automata with advice. Electronic Proceedings in Theoretical Computer Science, 96, 10 2012.

33. Kuzelka, O.: Weighted first-order model counting in the two-variable fragment with counting quantifiers. Journal of Artificial Intelligence Research, 70:1281–1307, 03 2021.

34. Laskowski, M. C.: Vapnik-chervonenkis classes of definable sets. Journal of The London Mathematical Society-second Series, 45:377–384, 1992.

35. Laskowski, M. C.: Mutually algebraic structures and expansions by predicates. The Journal of Symbolic Logic, 78(1):185–194, 2013.

36. Laskowski, M. C. and Terry, C.: Jumps in speeds of hereditary properties in finite relational languages. Journal of Combinatorial Theory, Series B, 2022.

37. Littlestone, N.: Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. In 28th Annual Symposium on Foundations of Computer Science (sfcs 1987), pages 68–77, 1987.

38. Lovász, L. and Szegedy, B.: Regularity Partitions and The Topology of Graphons, pages 415–446. Berlin, Heidelberg, Springer Berlin Heidelberg, 2010.

39. Malhotra, S. and Serafini, L.: Weighted model counting in fo2 with cardinality constraints and counting quantifiers: A closed form formula. Proceedings of the AAAI Conference on Artificial Intelligence, 36:5817–5824, 06 2022.

40. Malliaris, M. and Moran, S.: The unstable formula theorem revisited via algorithms, 2023, arXiv:2212.05050 [math.LO].

41. Malliaris, M. and Shelah, S.: Regularity lemmas for stable graphs. Transactions of the American Mathematical Society, 366(3):1551–1585, 2014.

42. Moerman, J., Sammartino, M., Silva, A., Klin, B., and Szynwelski, M.: Learning nominal automata. In Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, page 613–625, New York, NY, USA, 2017. Association for Computing Machinery.

43. Montanari, U. and Pistore, M.: History-Dependent Automata: An Introduction, pages 1–28. Berlin, Heidelberg, Springer Berlin Heidelberg, 2005.

44. Nguyen, T., Scott, A., and Seymour, P.: Induced subgraph density. vi. bounded vc-dimension, 2024, arXiv:2312.15572 [math.CO].

45. Nies, A.: Describing groups. The Bulletin of Symbolic Logic, 13(3):305–339, 2007.

46. Pitts, A. M.: Nominal Sets: Names and Symmetry in Computer Science. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2013.

47. Pyber, L.: Asymptotic results for permutation groups. In Groups And Computation, 1991.

48. Rabinovich, A. and Thomas, W.: Decidable theories of the ordering of natural numbers with unary predicates. In Computer Science Logic, ed. Z. Ésik, pages 562–574, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

49. Raedt, L. D., Kersting, K., and Natarajan, S.: Statistical Relational Artificial Intelligence: Logic, Probability, and Computation. Morgan & Claypool Publishers, 2016.

50. Richardson, M. and Domingos, P. M.: Markov logic networks. Machine Learning, 62:107–136, 2006.

51. Sakakibara, Y.: Learning context-free grammars from structural data in polynomial time. In Proceedings of the First Annual Workshop on Computational Learning Theory, COLT '88, page 330–344, San Francisco, CA, USA, 1988. Morgan Kaufmann Publishers Inc.

52. Salomaa, A.: On finite automata with a time-variant structure. Information and Control, 13(2):85–98, 1968.

53. Scheinerman, E. and Zito, J.: On the size of hereditary classes of graphs. Journal of Combinatorial Theory, Series B, 61(1):16–39, 1994.

54. Tsankov, T.: The additive group of the rationals does not have an automatic presentation. The Journal of Symbolic Logic, 76(4):1341–1351, 2011.

55. Tóth, J. and Kuželka, O.: Lifted inference with linear order axiom. Proceedings of the AAAI Conference on Artificial Intelligence, 37:12295–12304, 06 2023.

56. Valiant, L. G.: The complexity of enumeration and reliability problems. SIAM J. Comput., 8:410–421, 1979.

57. Van den Broeck, G.: On the completeness of first-order knowledge compilation for lifted probabilistic inference. In Neural Information Processing Systems, 2011.

58. Van den Broeck, G., Meert, W., and Darwiche, A.: Skolemization for weighted first-order model counting. In Proceedings of the Fourteenth International Conference

on Principles of Knowledge Representation and Reasoning, KR'14, page 111–120. AAAI Press, 2014.

VITA

KEVIN ZHOU

EDUCATION

Aug 2024	**Ph.D. in Pure Mathematics**
	University of Illinois Chicago, Chicago, IL
	Advisor: James Freitag
May 2018	**B.S. in Mathematical Sciences** (with University Honors)
	Carnegie Mellon University, Pittsburgh, PA

PAPERS

Kevin Zhou, Query learning bounds for advice and nominal automata, *ATVA*, to appear, 2024

Kevin Zhou, Hereditary properties and weighted model counting, in preparation

INVITED TALKS

Apr 2024	AMS Central Sectional Meeting | University of Wisconsin-Milwaukee
Apr 2024	UIC Logic Seminar | University of Illinois Chicago
Mar 2023	AMS Southeastern Sectional Meeting | Georgia Institute of Technology
Oct 2022	UIC Logic Seminar | University of Illinois Chicago
Apr 2022	ASL North American Annual Meeting | Cornell University

AWARDS & GRANTS

Spring 2022	UIC NSF TRIPODS Research Fellow
Summer 2019	UIC NSF RTG Pre-doctoral Fellow

TEACHING

As Primary Instructor:

MATH 090 Intermediate Algebra

MATH 179 ESP Workshop for Calculus I

MATH 294 ESP Workshop for Introduction to Advanced Mathematics

MATH 294 ESP for Abstract Algebra I

As Teaching Assistant:

MATH 090 Intermediate Algebra

MATH 110 College Algebra

MATH 125 Elementary Linear Algebra

MATH 180 Calculus I

MCS 260 Introduction to Computer Science

SERVICE

2023	Co-organizer, 23rd Graduate Student Conference in Logic
2021 – 2024	Co-organizer, UIC Louise Hay Logic Seminar (Graduate Logic Seminar)
2019 – 2020	Secretary, UIC Mathematics Graduate Students' Association