On Tractability and Consistency of Probabilistic Inference in Relational Domains

Sagar Malhotra

Supervised by Luciano Serafini

Fondazione Bruno Kessler, Italy
University of Trento, Italy

This dissertation is submitted for the degree of
Doctor of Philosophy

Trento, Italy

June 2023

Thesis Referees:

Prof. Manfred Jaeger
University of Edinburgh, Scotland, UK

Prof. Vaishak Belle
University of Aalborg, Aalborg, Denmark

Examination Committee:

Prof. Manfred Jaeger
University of Aalborg, Aalborg, Denmark

Prof. Marco Montali
University of Bolzano, Bolzano, Italy

Prof. Roberto Sebastiani
University of Trento, Trento, Italy

To the people who believe in
doing science for science

Acknowledgements

Starting this section with thanking my advisor is no cliché. Luciano Serafini is someone you meet, and you know you have met a man driven by science. Luciano taught me a lot of things. He taught me logic. He taught me mathematical rigor. He literally dragged me through my first proof. But most importantly, he taught me to persist! His openness to dive into new fields, his spirit for research, his excited celebrations on solving problems, and the pizzas he offered me, are things I will always cherish.

No PhD can survive without a good support system, DKM and PDI formed that system. Alessandro's continuous support, both emotional and academic, helped me survive the rejection-ridden second year of my PhD. His discussion time and rigorous proofreading constitute a significant pillar to this thesis. Tommaso's freindly presence, the joyous cohesion that he brought to DKM, made coming to office a joyride. Dino the other "Gorilla incontrabille", "bene bene" was a sheer delight. Leonardo the "Testa di ..." sorry for I punched you so much, and thanks for all the help in Haifa. Patrizio i.e., "PapPat" — there is something about you that warms my heart, any day is a better day, if you are in it. Gianluca aka Gengi, you know what I am thinking, before I think it, so I will not write it. Andrei — my superior partner in running, thanks for all the marvelous food and great company. Gabriele — talking about books, logic and philosophy with you has been very exciting and intellectually uplifting. Davide — thanks for the proofreading and the interest and appreciation you have shown for this line of work. Srajan — I have immensely enjoyed talking about logic and life with you, thanks for the friendship and support you have provided me. Abhinandan — it surprises me in a very familiar way the distance that we have travelled.

I would also like to thank the fellow academics I came in contact with outside my home university. Felix Weitkämper, your appreciation and understanding of my work was a major boost in confidence. I hope one day I can match up to your knowledge, rigor and humility. Thomas Gärtner, I admire your holistic vision of Machine Learning — theory and applications, and I thank you for seeing my work through that lens.

I probably would not have finished this PhD if not for Beatrice, her continuous support and perspective-drawing discussions kept me relaxed and focused. I also thank my cats Archie and Otis. They woke me up on time every morning, without a fail. They made sure I did not exhaust myself, by sitting on my laptop, if I was working too long. Seeing them sleep is probably the most heart-warming sight, and I got to see it every day!

My parents, Mummy and Papa, I always pushed your boundaries, and you always accommodated me. Thanks for always giving me the space to rebel, the space to struggle, the space to do my own mistakes. My sister, Saloni, thanks for training me with multitasking — by asking me all the possible questions while I am busy. You are funny! You are marvelous! And I should tell you this more often.

Finally, the best part of writing this thesis has been your support Paola. You make me feel supported, you make me feel happy, you make me feel complete. Your love, humor and unconditional support made even working on weekends a great experience.

Abstract

Relational data is characterised by the rich structure it encodes in the dependencies between the individual entities of a given domain. Statistical Relational Learning (SRL) combines first-order logic and probability to learn and reason over relational domains by creating parametric probability distributions over relational structures. SRL models can succinctly represent the complex dependencies in relational data and admit learning and inference under uncertainty. However, these models are significantly limited when it comes to the tractability of learning and inference. This limitation emerges from the intractability of Weighted First Order Model Counting (WFOMC), as both learning and inference in SRL models can be reduced to instances of WFOMC. Hence, fragments of first-order logic that admit tractable WFOMC, widely known as *domain-liftable*, can significantly advance the practicality and efficiency of SRL models.

Recent works have uncovered another limitation of SRL models, i.e., they lead to unintuitive behaviours when used across varying domain sizes, violating fundamental consistency conditions expected of sound probabilistic models. Such inconsistencies also mean that conventional machine learning techniques, like training with batched data, cannot be soundly used for SRL models.

In this thesis, we contribute to both the tractability and consistency of probabilistic inference in SRL models. We first expand the class of domain-liftable fragments with counting quantifiers and cardinality constraints. Unlike the algorithmic approaches proposed in the literature, we present a uniform combinatorial approach, admitting analytical combinatorial formulas for WFOMC. Our approach motivates a new family of weight functions allowing us to express a larger class of probability distributions without losing domain-liftability. We further expand the class of domain-liftable fragments with constraints inexpressible in first-order logic, namely *acyclicity* and *connectivity* constraints. Finally, we present a complete characterization for a statistically consistent (a.k.a projective) models in the two-variable fragment of a widely used class of SRL models, namely Markov Logic Networks.

Table of contents

Notation

Mathematical Symbols

$[b]$	The set of b 2-tables
$[n]$	$\{1 \cdots n\}$
$[u]$	The set of u 1-types
$\alpha, \beta, \gamma \ldots$	arbitrary logical formulas
$\uplus$	Disjoint union
$\boldsymbol{k} = \langle k_1 \cdots k_u \rangle$	1-type cardinality vector
Δ	Set of constants in $\mathcal{L}$ a.k.a. Domain/Domain constants
Ψ, Φ, Γ	First Order Logic Sentences
$\mathcal{R}$	Set of relational symbols in $\mathcal{L}$
$i(x)$	i^{th} 1-type in variable x
$ijl(x, y)$	$i(x) \wedge j(y) \wedge l(x, y)$
$l(x, y)$	l^{th} 2-table
$\mathcal{L}$	First order logic language
$\mathcal{R}$	Set of relational symbols in $\mathcal{L}$

Acronyms / Abbreviations

C^2	FOL with two variables and counting quantifiers
FO^2	FOL with two variables

AI	Artificial Intelligence
ERGM	Exponential Random Graph Models
FOL	First Order Logic
MLN	Markov Logic Networks
SRL	Statistical Relational Learning

Chapter 1

Introduction

Artificial Intelligence (AI) models that can *learn* and *reason*, while exploiting both *expert-knowledge* and *relational data* are of significant importance. Such models can make a significant impact in fields like social network analysis [1], biological systems [2], planning [3], epidemiology [4], medicine [5], recommendation systems [6] e.t.c. The key feature of relational data is the *dependencies* it represents between individual data attributes. However, this key feature conflicts with the fundamental assumption of the modern day *Machine Learning (ML)*, namely the *Independent and Identically Distributed (IID)* data assumption [7]. Identically distributed means that each data point is sampled from the same underlying distribution that does not fluctuate over time or with more samples. The independence condition imposes that such samples are not connected in any way. Relational data differs from IID data as it involves multiple entities that are related to each other in complex ways. For instance, in a social network, users are connected to other users in various ways (e.g., friends, followers, etc.), and those connections may influence their attributes. As a result, relational data violates the independence assumption that is central to many ML techniques. This makes applying traditional ML techniques to relational data difficult without losing important information and relationships between entities. Therefore, new methods are needed to handle the complexity and dependencies of relational data.

The scale and complexity of relational data demand a machine-readable representation language that, on the one hand, is succinct enough to be human-interpretable and, on the other hand, is rich enough to capture all the data nuances. *First-Order Logic (FOL)* serves this purpose by allowing quantification over data entities, also referred to as the *domain*, and expressing *relations* among the domain elements. For example, given a dataset of a human population going through a Covid-19 epidemic, we are given information about whether individuals have Covid or not and the other people they come in contact with. We may create a simple FOL language with a unary predicate $\text{Covid}(x)$ representing that x has covid and a binary predicate $\text{Contact}(x, y)$ representing that x came in contact with y. We may write the following simple FOL rule for contact tracing:

$$\forall xy.\text{Covid}(x) \wedge \text{Contact}(x, y) \rightarrow \text{Covid}(y) \tag{1.1}$$

Hence, given that we have $\text{Covid}(a)$ and $\text{Contact}(a, b)$, one can infer that $\text{Covid}(b)$. In fact such a reasoning procedure can be automated to infer potential new Covid cases from the available data. However, paraphrasing Russel and Norvig [8], accurately modelling complex real-world domains with logic is limited by:

- **Complexity:** Listing out/learning all the possible exception-less rules is an intractable process. In most rich enough structured datasets, a complete exception less description of the data is likely to be either prohibitively large or the trivial representation, i.e. the data itself. For example, in the aforementioned FOL rule (1.1) for contact-tracing, it is unlikely that the observed-data follows the rule completely. We may find many individuals in the population that violate the rule. In such a case, the formula will be unsatisfiable w.r.t the data and lead to failure of inference.

- **Practical Uncertainty:** When dealing with expert-knowledge, a large amount of knowledge comprises qualitative or quantitative *beliefs*. FOL provides no straight-forward method of exploiting such knowledge. For example, it is unlikely that an expert can quantify their uncertainity in the FOL rule (1.1), in any meaningful manner.

- **Theoretical Uncertainty:** Real-world processes, and hence the data generated by them, are often inherently *stochastic*, a reliable and generalizable FOL model of the data is unlikely to exist in such cases.

Probability theory provides a method for representing both the uncertainty in expert knowledge and the inherent stochasticity in the data-generation process. Hence, models that can exploit succinct FOL representation while also expressing the uncertainties in the data are exceedingly desirable. The field of *Statistical Relational Learning* (SRL) [9, 10] builds on exactly these motivations. SRL aims to construct probabilistic models over rich structured data, expressed in FOL languages extended with additional weight/probabilistic parameters. In general, SRL models assume a parametric probability distribution $P_\theta(\omega)$ over the set of FOL interpretations Ω. The model learning task then consists of estimating the model parameters θ^* that maximize the likelihood of the observed data ω, i.e.

$$\theta^* = \operatorname*{argmax}_\theta P_\theta(\omega)$$

and *inference* tasks consist of getting the probability of a query q, by summing over the probability of all the models that satisfy q, i.e.,

$$P(q) := \sum_{\substack{\omega \in \Omega \\ \omega \models q}} P_\theta(\omega)$$

1.1 Motivation and Challenges

Despite the rich expressivity, when it comes to *tractability* of learning and inference, SRL models fare no better than having a naive tabular representation of the probability distribution, with individual random variables for each entity in the domain — w.r.t FOL this means that the computational complexity[1] of learning and reasoning essentially reduces to the one of a propositional probabilistic language comprising individual ground atoms as boolean random variables. Dan Roth [11] showed that inference in such a language is in the computational complexity class #P- complete [12]. This means that probabilistic inference is at least as hard as enumerating all the possible satisfiable assignments to a propositional logic formula. Furthermore, the intractability result holds even under very restrictive fragments of logic and is true also for computational complexity of approximate probabilistic inference.

Another key challenge in SRL emerges from *inexpressivity* of FOL for many relavant global properties of real-world data. Intuitively, this inexpressivity arises from the implicit locality of any FOL property [13]. This locality means that FOL sentence cannot express properties like *connectivity*, *acyclicity* e.t.c. [14]. Hence, the only possibility for expressing such properties is through propositionalisation a.k.a. grounding. However, such a representation is exponentially large and renders inference/learning intractable.

Moreover, recent works [15] have shown that a large variety of widely used probabilistic models, encompassing almost all of existing SRL models [16], do not admit basic *consistency* requirements expected of sound statistical models. The lack of such consistency conditions means that SRL models do not admit consistency of parameter

[1]Whenever referring to computational complexity, we mean worst-case computational complexity.

estimation, i.e., as you see more and more data, it is not true that your parameters converge to the true value of the model parameters. Furthermore, SRL models are not even consistent under marginalization, i.e., if you take an SRL model describing a social network on (say) 2500 people, then the distribution it gives you on social networks of (say) 2499 people is not what you'd get by summing over networks of 2500 people. Such inherent inconsistencies in SRL models make them hard to be used across varying domain sizes [17, 18].

Both tractability and consistency are major challenges to SRL models. The former renders them unpractical, the latter unfounded. In this thesis, we make contributions towards addressing both these challenges, by characterizing (and expanding) SRL models that are tractable and consistent.

1.2 Contributions

This thesis makes five main contributions towards tractability, expressivity and consistency of SRL models. Since the computational bottleneck for learning and inference in a vast array of SRL models can be reduced to tractability (and expressivity) of Weighted First Order Model Counting (WFOMC), our tractability results are presented w.r.t WFOMC.

Firstly, we provide a sound and uniform framework for tractable WFOMC in the *two variable fragment of FOL, extended with cardinality constraints and counting quantifiers*. In comparison to existing decision-diagrams/logical-circuits based approaches, our approach is combinatorial in nature. We formalize a uniform framework, by defining new logic based combinatorial concepts (e.g., Definition 26). These concepts allow us to develop an analytical framework for WFOMC, allowing us to derive closed-form formulas for WFOMC in the two-variable fragment and its extensions with cardinality constraints and counting quantifiers.

Secondly, the framework we develop naturally motivates our second contribution: a larger, *more expressive class of weight functions* in the two-variable fragment. Although, past works have been able to express such distributions using complex valued weight-functions [19], we obtain the same expressivity using only real-valued weight functions in a rather simple manner.

Our third and fourth contributions take us towards expanding lifted inference to constraints not definable in FOL. We expand the domain-liftaibility of the two-variable fragment with an *acyclicity constraint*, i.e., a predicate in the language is axiomatized to represent a Directed Acyclic Graph (DAG). We then use similar techniques, to incorporate a *connectivity constraint*, i.e., a predicate in the language is axiomatized to represent a connected graph. We also show that both these constraints remain domain-liftable even in the presence of cardinality constrains and counting quantifiers.

Finally, we move on to the problem of consistency of inference. Recent works by Shalizi and Rinaldo Exponential Random Graphs (ERGMs) [15] have shown that ERGMs do not admit consistent parameter learning. They further show that this problem emerges from a deeper problem concerning consistency of ERGMs under marginalization. Jaeger and Shulte [16], formalized this problem for SRL, showing that only very restricted fragments of SRL models admit consistent marginalization properties. In this thesis, we provide a *complete characterization of consistent Markov Logic Networks in the 2-variable fragment.* We also show that this model reduces to a relational generalization of the Stochastic Block's Model [20].

1.3 Thesis Structure

The contributions in this thesis are based in the overlap of logic and probability theory, and the resultant SRL models.

Chapter 2 begins with providing essential background on FOL and probability distributions defined with respect to sufficient statistics. We then show that integrating these ideas, i.e. defining models on FOL interpretations with FOL-definable sufficient statistics leads us to a special class of SRL models, namely *Markov Logic Networks* [21]. We also provide some basic combinatorial principles, which are useful for many technical results of the thesis.

In Chapter 3, we present the first and the second contribution of this thesis i.e. WFOMC in the two variable fragment, expanded with cardinality constraints and counting quantifiers, and a general class of weight functions, which is strictly more expressive than symmetric weight functions. The work presented in this chapter is based on the following papers:

> Sagar Malhotra and Luciano Serafini. Weighted Model Counting in FO2 with Cardinality Constraints and Counting Quantifiers: A Closed Form Formula. *In proceedings of the AAAI Conference on Artificial Intelligence, 2022.* [22]

Sagar Malhotra and Luciano Serafini. A Combinatorial Approach to Weighted Model Counting in the Two-Variable Fragment with Cardinality Constraints. *In proceedings of International Conference of the Italian Association for Artificial Intelligence, 2021* [23]

In Chapter 4, we expand the class of domain-liftable theories in the two-variable fragment to a constraint in-expressible in FOL, namely an *acyclicity constraint*. The work presented in this chapter is based on the following (under-review) arxiv preprint:

Sagar Malhotra and Luciano Serafini. Weighted First Order Model Counting with Directed Acyclic Graph Axioms. arXiv:2302.09830 [24]

We further expand the class of domain-liftable theories in Chapter 5, by adding another constraint in-expressible in FOL: a *connectivity axiom*. The work presented in this chapter is based on the following under-review paper.

Sagar Malhotra and Luciano Serafini. Weighted First Order Model Counting with Connectivity Axioms.

Finally, in Chapter 6, we deal with consistency of learning and inference in Markov Logic networks. We provide the necessary and sufficient conditions for a 2-variable MLN to admit consistency of inference and learning. The work presented in this chapter forms part of the following publication:

Sagar Malhotra and Luciano Serafini. Projectivity in Markov Logic Networks. *In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022* [25]

Chapter 2

Background

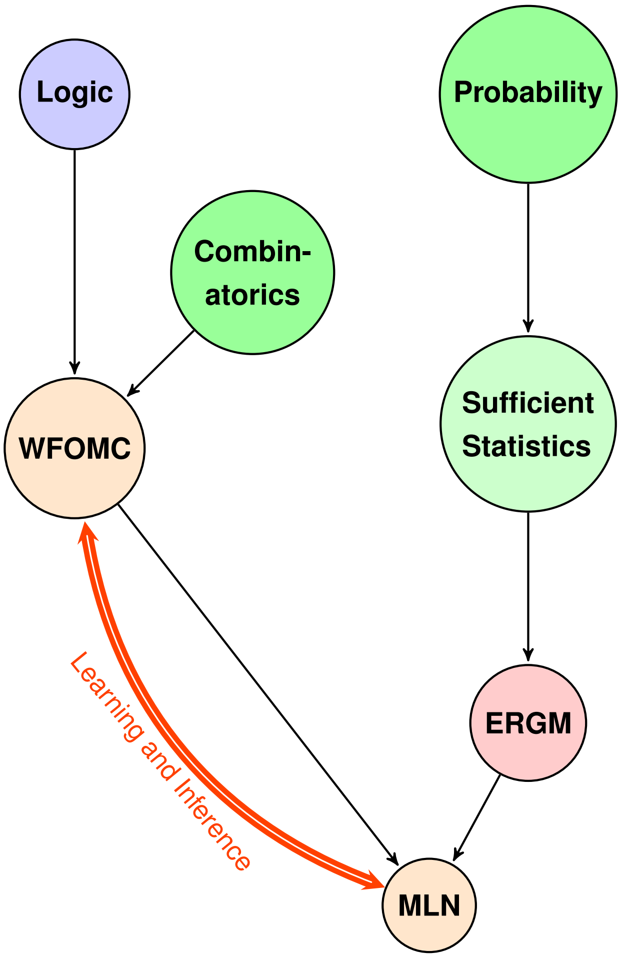


Fig. 2.1 Concept Dependency Graph for Background

Our goal in this thesis is to investigate learning and inference in probability distribution ascribed to First-Order Logic (FOL) interpretations. In this chapter, we provide the necessary background in FOL and probability theory. We assume basic understanding of propositional logic and directly begin with FOL in section 2.1, where we introduce the necessary background in FOL required for this thesis. We then briefly introduce Weighted First Order Counting (WFOMC) in section 2.2, and relevant results on complexity of WFOMC in the literature. We then move on to introducing

2.1 First Order Logic

Every logic comprises a formal *language* for making statements about objects and reasoning about their properties. Statements in a logical language are constructed according to a set of rules known as the *syntax*. The meaning to these statements (their truth value or their probabilities) is given by the *semantics*.

We will deal with a fragment of First-Order Logic (FOL), also known as the *Herbrand Logic* [26], which can be succinctly described as follows:

$$\text{Herbrand Logic} := \text{First Order Logic Syntax} + \text{Herbrand Semantics}$$

Furthermore, we will assume a completely relational language, i.e., there are no function symbols in the language. Whenever referring to Herbrand Logic, we intend its function-free variant. All the results are provided with respect to function-free Herbrand logic, with finite set of relational symbols and constants. We will now define the syntax and semantics of function-free Herbrand Logic formally. Our goal is not to capture every aspect of FOL or even Herbrand Logic, but to rather capture the fragment of FOL relevant to this thesis and largely used in the AI and SRL community.

2.1.1 Syntax

Herbrand Logic follows exactly the same syntax as FOL. We will now introduce a function-free syntax of FOL, which will be assumed throughout this thesis.

Definition 1 (Language)**.** *A function-free* FOL *language $\mathcal{L}$ consists of:*

- Logical connectives*: $\wedge$ (and), $\vee$ (or), $\neg$ (not), $\rightarrow$ (implication) and $\leftrightarrow$ (iff)*
- Quantifiers*: $\forall$ (forall) and $\exists$ (exists)*
- Relational symbols: *A finite set of relational symbols, $\mathcal{R} := \{R_i/r_i\}_i$, where R_i is a relational symbol with arity r_i*
- Variables*: A finite set of variables $\mathcal{V} := \{x_i\}_i$*

- Equality Symbol*:* $=$
- Constants*: A finite set of constants* $\Delta := \{c_i\}_i$
- Auxiliary Symbols: *Parenthesis and commas*

Definition 2 (Atom)**.** *An atom in* $\mathcal{L}$ *is a string of the form:*

- $(t_1 = t_2)$ *where* $t_1, t_2 \in \Delta \cup \mathcal{V}$
- $R(t_1, \ldots, t_r)$*, where* $t_i, \ldots, t_r \in \Delta \cup \mathcal{V}$*,* $R \in \mathcal{R}$ *and* r *is the arity of the relational symbol* R*.*

We denote the set of all atoms in $\mathcal{L}$*.*

Definition 3 (Litera)**.** *A literal is an atom or its negation.*

Definition 4 (Ground atom)**.** *An atom with no variables in it is called a ground atom. We denote the set of ground atoms in* $\mathcal{L}$ *with* $\mathcal{G}$*.*

Definition 5 (Predicate function)**.** *Given a ground atom* $g \in \mathcal{G}$ *in* $\mathcal{L}$*, then the function* $\mathrm{pred} : \mathcal{G} \to \mathcal{R}$*, maps the atom* g *to the predicate occurring in* g*.*

Example 1. *Given a ground atom* $R(a, b)$*, then* $\mathrm{pred}(R(a, b)) = R$*.*

Definition 6 (Formula)**.** *A formula is defined as follows:*

- *An atom*
- *$(\neg\alpha)$, where α is a formula*
- *$(\alpha \wedge \beta)$, where α and β are formulas*
- *$(\alpha \vee \beta)$, where α and β are formulas*
- *$(\alpha \to \beta)$, where α and β are formulas*
- *$(\alpha \leftrightarrow \beta)$, where α and β are formulas*
- *$(\forall x.\alpha)$, where α is a formula and x is a variable*
- *$(\exists x.\alpha)$, where α is a formula and x is a variable*
- *Only the expressions produced by above rules are formulas*

Definition 7 (Ground formula). *A formula with no variables in it is called a ground formula.*

Definition 8 (Free variables). *Free variables of formula α, denoted by $FV[\alpha]$, are recursively defined as follows:*

- $FV[x_i] = \{x_i\}$, *for* $x_i \in \mathcal{V}$
- $FV[c_i] = \emptyset$, *for* $c_i \in \Delta$
- $FV[R(t_1, \ldots, t_r)] = FV[t_1] \cup \ldots \cup FV[t_r]$, *where* $R(t_1, \ldots, t_r)$ *is an atom*
- $FV[(t_1 = t_2)] = FV[t_1] \cup FV[t_2]$, *where* $(t_1 = t_2)$ *is an atom*
- $FV[(\neg\alpha)] = FV[\alpha]$, *where* α *is a formula*
- $FV[(\alpha * \beta)] = FV[\alpha] \cup FV[\beta]$, *where* $* \in \{\wedge, \vee, \leftrightarrow, \rightarrow\}$
- $FV(\forall x.\alpha) = FV(\alpha) - \{x\}$,*where* α *is a formula*
- $FV(\exists x.\alpha) = FV(\alpha) - \{x\}$,*where* α *is a formula*

Definition 9 (Sentence). *A formula with no free variables is called a sentence.*

2.1.2 Semantics

Definition 10 (Herbrand interpretation/model). *A Herbrand interpretation/model ω is a subset of ground atoms in the* FOL *language $\mathcal{L}$. We denote the set of Herbrand interpretations as Ω.*

Remark 1. *With an abuse of notation, we will also use the equivalent notion of Herbrand interpretation/models as a truth assignment to ground atoms, i.e., $\omega : \mathcal{G} \rightarrow \{\mathrm{T}, \mathrm{F}\}$. The two equivalent notations are defined as follows:*

$$g \in \omega \leftrightarrow \omega(g) = \mathrm{T}$$
$$g \notin \omega \leftrightarrow \omega(g) = \mathrm{F}$$

Definition 11 (Herbrand Satiafaction). *Let α be a sentence and ω be a Herbrand interpretation in $\mathcal{L}$. Then:*

- $\omega \models s = t$ *if and only if* s *and* t *are syntactically identical*
- $\omega \models R(t_1, \ldots, t_2)$ *if and only if* $R(t_1, \ldots, t_2) \in \omega$

- $\omega \models \neg\alpha$ *if and only if* $\omega \not\models \alpha$
- $\omega \models \alpha \wedge \beta$ *if and only if* $\omega \models \alpha$ *and* $\omega \models \beta$
- $\omega \models \alpha \vee \beta$ *if and only if* $\omega \models \alpha$ *or* $\omega \models \beta$
- $\omega \models \alpha \rightarrow \beta$ *if and only if* $\omega \not\models \alpha$ *or* $\omega \models \beta$
- $\omega \models \alpha \leftrightarrow \beta$ *if and only if* $\omega \models \alpha \wedge \beta$ *or* $\omega \models \neg\alpha \wedge \neg\beta$
- $\omega \models \forall x.\alpha(x)$ *if and only if* $\omega \models \alpha(c)$ *for all* $c \in \Delta$
- $\omega \models \exists x.\alpha(x)$ *if and only if* $\omega \models \alpha(c)$ *for some* $c \in \Delta$

Satisfaction of a set sentences Γ is defined with respect to a language that includes all the variables and constants that appear in Γ. This is to avoid situations where a model fails to satisfy either a sentence or its negation.

A consequence of above definitions is that Herbrand Logic satisfies the domain closure and unique-name assumption, commonly used in AI applications of FOL. *Furthermore, since the domain over which the variables are quantified is simply the set of constants* Δ*, we will call* Δ *the domain or domain constants, as is common in SRL literature.*

Definition 12 (Herbrand Entailment)**.** *Let* Γ *be a set of sentences in the language* $\mathcal{L}$*. Let* α *be another sentence in* $\mathcal{L}$*. Then* Γ *entails* α *if and only if every Herbrand model in* Ω *that satisfies* Γ *also satisfies* α*.*

Having defined Herbrand logic, we will now assume it for the rest of the thesis. Hence, whenever referring to FOL/FOL-language/FOL-interpretation/FOL-formula, we will intend function-free finite Herbrand logic as presented here.

Remark 2. *Since we are dealing with Herbrand interpretations, we can always bijectively label the domain constants with the set* $[n]$*, where* $n = |\Delta|$*, using a bijective map* $\pi : \Delta \rightarrow [n]$*. Any property that needs to be inferred on sentences w.r.t* Δ *can be equivalently inferred w.r.t the relabeled domain i.e.* $[n]$*. Hence, we can assume* $[n]$ *to be the domain without loss of generality.*

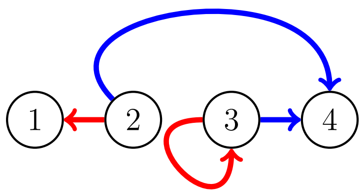
2.1.3 Projection and Cardinality Constraints

We will also use the notion of projection of an interpretation on both domain and relational symbols defined as follows:

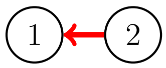
Definition 13 (Projection of an interpretation on subset of the domain)**.** *Given an* FOL *language* $\mathcal{L}$ *with a domain* Δ*, let* $\Delta' \subset \Delta$*, then* $\omega \downarrow \Delta'$ *is an interpretation that assigns truth assignment only to the ground atoms defined on* Δ'*.*

Definition 14 (Projection of an interpretation on a predicate)**.** *Given an* FOL *language* $\mathcal{L}$*, then* ω_R *for a relational symbol* $R \in \mathcal{L}$ *is an interpretation that assigns truth assignment only to the ground atoms* g*, such that* $\mathrm{pred}(g) = R$*.*

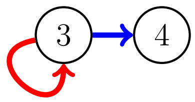
Example 2. *Let us have a language with only two relational symbol* R *and* B *both of arity* 2*, with a domain* $\Delta = [4]$*. We represent an interpretation* ω *as a multi-relational directed graph, where a pair of domain elements* c *and* d *have a red (resp. blue) directed edge from* c *to* d *if* $R(c,d)$ *(resp.* $B(c,d)$*) is true in* ω *and have no red (resp. blue) edge otherwise. Let us take for example the following interpretation* ω *on* $[4]$*:*



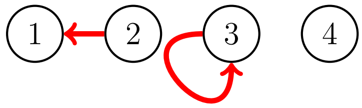
then $\omega' = \omega \downarrow [2]$ *and* $\omega'' = \omega \downarrow [\bar{2}]$ *are given as:*



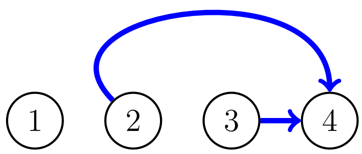
and



respectively. Projecting on the predicate R*, denoted by* ω_R *is given as:*



Similarly, projecting on the predicate B*, denoted by* ω_B *is given as:*



Definition 15. *[Cardinality Constraints] Let ω be an $\mathcal{L}$-interpretation, let $\{P_i\}_i \subseteq \mathcal{R}$ be a subset of predicate symbols in $\mathcal{L}$. We say $\omega \models (|P_i|= l)$, if the number of ground atoms $g \in \omega$, such that* $\mathrm{pred}(g) = P_i$, *is equal to l. Similarly, a cardinality constraint $\Gamma(\{|P_i|\}_i)$ is an arithmetic constraint on the cardinality of the set of predicates P_i i.e. $|P_i|$ in ω, and $\omega \models \Gamma(\{|P_i|\}_i)$, if the predicate cardinalities in ω satisfy the constraint $\Gamma(\{|P_i|\}_i)$.*

Example 3 (Cardinality Constraints)**.** *Continuing Example 2, let* $\Gamma(R) := |R| < 2$, *then $\omega \not\models \Gamma$, whereas $\omega \downarrow [2] \models \Gamma$ and $\omega \downarrow [\bar{2}] \models \Gamma$.*

Remark 3. *We use $\Delta = \Delta' \uplus \Delta''$ to denote that Δ is a union of two disjoint sets Δ' and Δ''. If ω' is an interpretation on Δ' and ω'' is an interpretation on Δ'', then we use $\omega' \uplus \omega''$ to denote the partial interpretation on $\Delta' \uplus \Delta''$, obtained by interpreting ground atoms over Δ' as interpreted in ω' and ground atoms over Δ'' as interpreted in ω''. However, the ground atoms involving domain constants from both Δ' and Δ'' are left uninterpreted in $\omega' \uplus \omega''$.*

2.2 Weighted First Order Model Counting

Given an FOL language $\mathcal{L}$ with a domain Δ of size n, First Order Model Counting (FOMC) is the problem of computing the sum of all the models that satisfy Φ. Formally,

$$\text{FOMC}(\Phi, n) := \sum_{\omega \models \Phi} 1 \tag{2.1}$$

Weighted First Order Model Counting (WFOMC), generalizes this problem by adding a weight function $\text{W} : \Omega \to \mathbb{R}$, that assigns a real-valued weight to each interpretation ω. Hence, WFOMC is formally defined as follows:

$$\text{WFOMC}(\Phi, \text{W}, n) := \sum_{\omega \models \Phi} \text{W}(\omega) \tag{2.2}$$

Since, we are dealing with Herbrand Logic, one may argue that the WFOMC can be performed by simply grounding out the formula and counting its satisfiable assignments using brute-force enumeration. However, such a task is known to be computationally intractable. Formally this intractability is captured in the following result by Valiant:

Theorem 1. *[Valiant, 1979 [27]] Let α be a Boolean formula, then the computational complexity of computing the number of satisfying assignments of α is #P-Complete.*

Here, $\#P$ is the counting analouge of the complexity class NP, i.e., it takes exponential amount of time in the size of the input. Surprisingly, intractability of counting solutions to boolean formula remains even when satisfiability is tractable:

Theorem 2. *[Valiant, 1979 [27]] Let α be a Boolean formula in DNF, then the computational complexity of computing the number of satisfying assignments of α is $\#P$-Complete.*

Furthermore, Beame et. al [28], extend this intractability result to WFOMC. However, in order to have the domain size as the notion of input, keeping Φ and weights fixed, the appropriate complexity class for their analysis is $\#P_1$: the class of counting problems with input in the unary language 1^n.

Theorem 3 (Beame et al., 2015 [28])**.** *There exists an FOL sentence Φ using at most three variables whose data complexity for (W)FOMC is in the class $\#P_1-$hard.*

There do exist positive results on counting complexity in WFOMC over formulas with at most two variables, and extensions to this result will form a major focus of this thesis. These results rely on a special class of weight functions known as the symmetric weight functions:

Definition 16. *(Symmetric Weight Function) Given a function-free first order logic language $\mathcal{L}$ over a domain Δ, where $\mathcal{G}$ are the set of ground atoms. A symmetric weight function associates two real-valued weights $w : \mathcal{R} \to \mathbb{R}$ and $\bar{w} : \mathcal{R} \to \mathbb{R}$ to each relational symbol in $\mathcal{L}$. The weight of an interpretation ω is then defined as:*

$$\mathrm{W}(\omega) = \prod_{\substack{\omega \models g \\ g \in \mathcal{G}}} w(pred(g)) \prod_{\substack{\omega \models \neg g \\ g \in \mathcal{G}}} \bar{w}(pred(g)). \tag{2.3}$$

We use $(w, \bar{w})$ to denote a symmetric weight function.

The class of weight functions and FOL formulas that admit tractable WFOMC are largely known as domain-liftable [29]. We formally have the following definition:

Definition 17 (Domain liftable)**.** *An FOL sentence Φ is said to be domain-liftable if for a given weight function W and domain of size n, $\mathrm{WFOMC}(\Phi, \mathrm{W}, n)$ can be computed in time polynomial w.r.t n.*

We are finally ready to provide the positive tractability results in WFOMC:

Theorem 4 (Beame et al. (2015) [28])**.** *Given a sentence Φ in FOL, with at most two variables, then the Symmetric Weighted First Order Model Counting of Φ is domain liftable.*

2.3 Sufficient Statistics

In statistics and probabilistic machine learning, we are concerned with estimating the value for parameters $\boldsymbol{\theta}$, of an assumed parametric family of distributions $P(\boldsymbol{X};\boldsymbol{\theta})$, given some observed data $\boldsymbol{x}$, such that the data likelihood, i.e., $P(\boldsymbol{X}=\boldsymbol{x};\boldsymbol{\theta})$, is maximized. A sufficient statistic $T(\boldsymbol{X})$, with respect to $P(\boldsymbol{X};\boldsymbol{\theta})$, is a function of $\boldsymbol{X}$ such that any other function calculated on $\boldsymbol{X}$ gives no more information about $\boldsymbol{\theta}$ than $T(\boldsymbol{X})$. Hence, given observed data $\boldsymbol{x}$ a statistician who knows $T(\boldsymbol{x})$, does no worse in estimating the value of $\boldsymbol{\theta}$, than a statistician who knows $\boldsymbol{x}$.

We define sufficient statistic more formally as follows:

Definition 18 (Sufficient Statistic)**.** *Given a parametric probability distribution $P(\boldsymbol{X};\boldsymbol{\theta})$, T is a sufficient statistic w.r.t $\boldsymbol{\theta}$ if:*

$$P(\boldsymbol{X}|T(\boldsymbol{X});\boldsymbol{\theta}) = P(\boldsymbol{X}|T(\boldsymbol{X});\boldsymbol{\theta'}) \tag{2.4}$$

for all possible values of parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta'}$.

Example 4. *Let $\boldsymbol{X} = X_1, \ldots, X_n$ be a sequence of Bernoulli trials with $P(X_i = 1) = \theta$. We will verify that $T(\boldsymbol{X}) = \sum_i X_i$ is sufficient for θ.*

$$P(X_1 = x_1, \ldots, X_n = x_n|T(X_1, \ldots, X_n) = t; \theta) = \frac{P((X_1 = x_1, \ldots, X_n = x_n) \wedge (T = t); \theta)}{P(T = t; \theta)}$$

when the number of X_i's with value 1 is different from t, then the numerator in RHS and LHS is identically equal to 0. Otherwise, in RHS, the probability in numerator is the probability of getting t X_i's with value 1 and other $n-t$ X_i's as 0s. Since X_i's are independent we have that the numerator is equal to $\theta^t(1-\theta)^{n-t}$. The denominator in RHS, on the other hand is given by $\binom{n}{t}\theta^t(1-\theta)^{n-t}$, as no order on trials is given. Hence, we have that:

$$\begin{aligned} P(X_1 = x_1, \ldots, X_n = x_n|T(X_1, \ldots, X_n) = t; \theta) &= \frac{\theta^t(1-\theta)^{n-t}}{\binom{n}{t}\theta^t(1-\theta)^{n-t}} \\ &= \frac{1}{\binom{n}{t}} \end{aligned}$$

Hence, $P(\boldsymbol{X}|T(\boldsymbol{X});\theta)$ does not depend on θ. Hence, $T(\boldsymbol{X}) = \sum_i X_i$ is sufficient for θ.

We now provide the Fisher-Neyman factorization theorem, which gives an easier method of recognizing sufficient statistics of a given parametric family of distributions.

Theorem 5 (Fisher–Neyman factorization). *$T(\boldsymbol{X})$ is sufficient for $P(\boldsymbol{X}|T(\boldsymbol{X});\boldsymbol{\theta})$ iff non-negative functions g and h can be found such that:*

$$P(\boldsymbol{X}|T(\boldsymbol{X});\boldsymbol{\theta}) = g(T(\boldsymbol{X}),\boldsymbol{\theta})h(\boldsymbol{X}) \tag{2.5}$$

We refer the reader to [30] or any other mathematical statistics textbook for a proof of the Fisher–Neyman factorization theorem.

Example 5. *We continue with the previous example of probability distribution on n Bernoulli random variables:*

$$\begin{aligned} P(\boldsymbol{X};\theta) &= \prod_{i=1}^{n} \theta^{X_i}(1-\theta)^{1-X_i} \\ &= \prod_{i=1}^{n} \theta^{\sum_i X_i}(1-\theta)^{\sum_i (1-X_i)} \\ &= \theta^{(\sum_i X_i)}(1-\theta)^{n-(\sum_i X_i)} \\ &= \theta^{T(\boldsymbol{X})}(1-\theta)^{n-T(\boldsymbol{X})} \end{aligned} \tag{2.6}$$

Now, $g(T(\boldsymbol{X}),\boldsymbol{\theta}) = \theta^{T(\boldsymbol{X})}(1-\theta)^{n-T(\boldsymbol{X})}$ and $h(\boldsymbol{X}) = 1$. Hence, due to Fisher-Nayman factorisation we have that $T(\boldsymbol{X}) = \sum_i X_i$ is a sufficient statistic.

Given Fisher-Neyman factorization, we get that maximizing likelihood, i.e., $P(\boldsymbol{X} = \boldsymbol{x};\boldsymbol{\theta})$ is equivalent to maximizing $g(T(\boldsymbol{X} = \boldsymbol{x}),\boldsymbol{\theta})$. Hence, there exists a maximum likelihood estimator $\hat{\boldsymbol{\theta}}$, which depends on $\boldsymbol{x}$ only through $T(\boldsymbol{x})$. Hence, there exists a maximum likelihood estimator that is a function of $T(\boldsymbol{x})$ [30].

2.4 Exponential Family of Distributions

In the previous section, we only saw examples of scalar valued parameters $\boldsymbol{\theta}$, however, when $\boldsymbol{\theta} = \langle \theta_1 \dots \theta_k \rangle$ is a vector, then the sufficient statistics can also be vector valued, i.e., $T(\boldsymbol{X}) = \langle T_1(\boldsymbol{X}) \dots T_k(\boldsymbol{X}) \rangle$, such that $T_i(\boldsymbol{X})$ is a sufficient statistic for θ_i. All the results of the previous section follow as it is in the vector valued case. One way to create a model where our desired statistics $T(\cdot)$ are sufficient is by using them in an exponential family distribution. In the following we give a slightly narrow definition of exponential family of distributions — which is general enough for our purposes.

Definition 19. *A model $P(\boldsymbol{X};\boldsymbol{\theta})$ is an exponential family distribution if it can be written in the form:*

$$P(\boldsymbol{X};\boldsymbol{\theta}) = \frac{\exp(\boldsymbol{\theta} \cdot T(\boldsymbol{X}))}{\sum_{\boldsymbol{x}} \exp(\boldsymbol{\theta} \cdot T(\boldsymbol{x}))} \tag{2.7}$$

where $\boldsymbol{\theta} \cdot T(\boldsymbol{X})$ denotes dot product between $\boldsymbol{\theta}$ and $T(\boldsymbol{X})$. The denominator in equation (2.7) *is called the partition function, denoted by $Z(\boldsymbol{\theta})$.*

Using Fisher-Neyman factorization, it can be seen that in an exponential distribution, $T(\boldsymbol{X})$ are a sufficient statistic w.r.t $\boldsymbol{\theta}$. Hence, exponential families provide a simple method to create models with predefined sufficient statistics.

2.5 Exponential Random Graphs

The idea of creating exponential families can be exploited to model rich probability distributions on graphs. Such models are known as Exponential Random Graph Models(ERGMs).

Definition 20 (ERGM)**.** *Exponential-family Random Graph Models (ERGMs) are exponential families over graphs. In other words, the sufficient statistics are functions of the graph/adjacency matrix.*

Example 6. *Let $\Omega^{(n)}$ represent the set of simple undirected graphs on $[n]$. In order to define an exponential family on $\Omega^{(n)}$, we define the number of edges $n_1(\omega)$ and the number of triangles $n_2(\omega)$ as sufficient statistics, with parameters θ_1 and θ_2 respectively. Hence, for $\omega \in \Omega^{(n)}$, we have the sufficient statistic $T(\omega) = \langle T_1(\omega) = n_1(\omega), T_2(\omega) = n_2(\omega)\rangle$, and the parameters $\boldsymbol{\theta} = \langle \theta_1, \theta_2 \rangle$.*

$$P_{\boldsymbol{\theta}}(\omega) = \frac{\exp(T(\omega) \cdot \boldsymbol{\theta})}{Z(\boldsymbol{\theta})} \tag{2.8}$$

where $T(\omega) \cdot \boldsymbol{\theta}$ represents the dot product. Hence, $T(\omega) \cdot \boldsymbol{\theta} = n_1(\omega)\theta_1 + n_2(\omega)\theta_2$.

2.6 Markov Logic Networks

Markov Logic Networks (MLNs) [21] are SRL models that ascribe an exponential family of probability distributions to FOL interpretations, by defining sufficient statistics represented by FOL formulas. Hence, an MLN is defined as follows:

Definition 21. *[Markov Logic Networks] Given a first order logic language $\mathcal{L}$ with a fixed and finite domain Δ, an* MLN $\mathcal{M}$ *consists of a finite set of weighted quantifier-free first-order logic formulas $\{\alpha_i, w_i\}_i$, where α_i are first-order logic formulas and each w_i are real-valued weights. An* MLN *ascribes the following probability distribution over all the possible interpretations of a given first-order logic sentence Φ:*

$$P_{\mathcal{M}}(\omega) := \frac{1}{Z} \exp\left(\sum_i w_i n_i(\omega)\right) \tag{2.9}$$

where $n_i(\omega)$ is the number of groundings of α_i satisfied in the interpretation ω and Z is the partition function defined as follows:

$$Z := \sum_{\omega \models \Phi} \exp\left(\sum_i w_i n_i(\omega)\right) \tag{2.10}$$

MLNs can be equivalently seen as exponential-family probability distributions described over FOL interpretations, where the sufficient statistics $T(\omega)$ has the following form:

$$T(\omega) = \langle T_1(\omega), \ldots, T_k(\omega) \rangle = \langle n_1(\omega), \ldots, n_k(\omega) \rangle$$

And the parameter vector $\boldsymbol{\theta}$ has the following form:

$$\boldsymbol{\theta} = \langle \theta_1 \ldots \theta_k \rangle = \langle w_1 \ldots w_k \rangle$$

Hence, following the Fisher-Neyman factorization as given in Theorem 5, it is easy to see that α_i's form sufficient statistics[1] for an MLN distribution.
Following is an illustrative example of a Markov Logic Networks:

Example 7. *[MLN Example] We go back to the example introduced in the introduction section. We have a human population and our goal is to build a model for contact tracing i.e. a model that allows for accurately predicting probability of a human having covid, given its contacts with other human-beings:*

$$w_1 : \text{Vaccinated}(x) \wedge \neg\text{Covid}(x) \tag{2.11}$$

$$w_2 : \text{Covid}(x) \wedge \text{Contact}(x, y) \rightarrow \text{Covid}(y) \tag{2.12}$$

where w_1 and w_2 are the weight's that reflect confidence in the logical formulas being true in the domain.

[1] More formally, the functions $n_i(\omega)$ form the sufficient statistics.

The weights w_1 and w_2 can reflect an expert's relative confidence in the rules or can be learned from data. Notice that negative weights can be used to reduce probability of certain logical formulas. In the example above, making w_1 negative is a reasonable choice.

2.6.1 MLNs and WFOMC

Van den Broeck et al. [31] provided a simple yet very useful method of converting the problem of computing the partition function of an MLN to a WFOMC problem, through the following encoding.

Given an MLN as defined in Definition 21, we can encode the computation of the partition function in to Symmetric-WFOMC of the following FOL sentence:

$$\Phi \wedge \bigwedge_{i} \forall FV[\alpha_i].(R_i(FV[\alpha_i]) \leftrightarrow \alpha_i) \tag{2.13}$$

where $FV[\alpha_i]$ are free variable in α_i, R_i are fresh predicate symbols not previously existing in α_i, and $R_i(FV[\alpha_i])$ has exactly the free variable in α_i. The symmetric weight function $(w, \bar{w})$, where $w(R_i) = \exp(w_i)$ and $\bar{w}(R_i) = 1$, the rest of the relational symbol and their negation has weight 1.

2.6.2 Learning

Given an observed MLN $\mathcal{M} = \{\alpha_i, w_i\}_{i=1}^k$, and an interpretation ω, we can learn the parameters $\boldsymbol{\theta} := \langle w_1, \ldots, w_k \rangle$, through maximizing the log-likelihood of the observed interpretation.

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log P_{\mathcal{M},\boldsymbol{\theta}}(\omega) \tag{2.14}$$

Although, simple to state, but a slightly deeper investigation of the log-likelihood shows its hidden complexity:

$$\log P_{\mathcal{M},\theta}(\omega) = \sum_{i} w_i n_i(\omega) - \log Z(\boldsymbol{\theta}) \tag{2.15}$$

Now, writing gradient descent for estimating the parameters, we have that:

$$w_i^{t+1} = w_i^t - \gamma \frac{\partial \log P_{\mathcal{M},\theta}(\omega)}{\partial w_i^t} \tag{2.16}$$

where $\frac{\partial \log P_{\mathcal{M},\theta}(\omega)}{\partial w_i^t}$, reflects the following gradient:

$$\frac{\partial \log P_{\mathcal{M},\theta}(\omega)}{\partial w_i} = n_i(\omega)\frac{1}{Z(\boldsymbol{\theta})}\frac{\partial \log P_{\mathcal{M},\theta}(\omega)}{\partial w_i} \tag{2.17}$$

Notice, that $Z(\boldsymbol{\theta})$ is computed in each iteration of equation (2.16), hence tractability of $Z(\boldsymbol{\theta})$ is intimately connected to learning complexity. Kuželka and Kungurtsev [32] formalize this in the following theorem:

Theorem 6 (Kuželka and Kungurtsev [32])**.** *Let $\Psi = (\alpha_1, \ldots, \alpha_l)$ be a list of first-order logic formulas and Φ_0 be a set of first-order logic sentences. Let Ω_{Φ_0} be the set of models of Φ_0 over a given domain Δ. Let $\omega_b \in \Omega$ be a training example. If computing the partition function of the MLN given by the formulas Ψ on Ω_{Φ_0} is domain liftable, then there is an algorithm which finds weights $w = (w_1, \ldots, w_l)$ such that the difference between probability, of an interpretation ω, assigned by the MLN with weights w and the optimal MLN is at most 2ϵ. The algorithm runs in time polynomial in $|\Delta|$ and $1/\epsilon$.*

2.6.3 Inference

Inference in MLN can also be reduced to WFOMC, let q a query then the probability of the query reduces to:

$$P(q) = \sum_{\omega \models q} P(\omega) \tag{2.18}$$

If q is an FOL sentence then the problem reduces to Symmetric-WFOMC.

2.7 Notation and Basic Combinatorics

We use the following basic notation. The set of integers $\{1, ..., n\}$ is denoted by $[n]$. We use $[m : n]$ to denote the set of integers $\{m, ..., n\}$. Wherever the set of integers $[n]$ is obvious from the context we will use $[\overline{m}]$ to represent the set $[m + 1 : n]$. Bold-face small-case Latin letters (with or without indices), e.g., $\boldsymbol{k}$ are used to represent vectors, corresponding normal-face letters, e.g., k_i, are used to represent the components of the vector. Given a vector $\boldsymbol{k} = \langle k_1 \ldots k_{[u]} \rangle$, we use $|\boldsymbol{k}|$ to denote the sum of its components, i.e., $\sum_{i \in [u]} k_i$. We will also use multinomial coefficients denoted by

$$\binom{n}{k_1, ..., k_u} = \binom{|\boldsymbol{k}|}{\boldsymbol{k}} = \frac{n!}{\prod_{i\in[u]} k_i!}$$

We also introduce the principle of inclusion-exclusion which will be used extensively throughout the thesis.

2.7.1 Principle of Inclusion-Exclusion

Given a set of finite sets $\{A_i\}_{i\in[n]}$, let $A_J := \bigcap_{j\in J} A_j$ for an arbitrary subset J of $[n]$. Then the principle of inclusion-exclusion (PIE) states that:

$$\Big|\bigcup_i A_i\Big| = \sum_{\emptyset\neq J\subseteq[n]} (-1)^{|J|+1}|A_J| \tag{2.19}$$

For all subsets $J, J' \subseteq [n]$, such that $|J|= |J'|= m$ for some $m \geq 1$, if A_J and $A_{J'}$ have the same cardinality, then there are $\binom{n}{m}$ terms in equation (2.19), with value $A_{[m]}$. Hence, equation (2.19) reduces to:

$$\Big|\bigcup_i A_i\Big| = \sum_{m=1}^{n}(-1)^{m+1}\binom{n}{m}A_{[m]} \tag{2.20}$$

Remark 4. *PIE can be easily extended to the case when A_i are sets of weighted FOL interpretations, where each interpretation ω has a weight $\mathrm{W}(\omega)$, where W is the symmetric weight function as given in Definition 16. In this case PIE allows us to computed the weighted sum of all the interpretations in $\bigcup_i A_i$.*

Let $\mathrm{W}(A_i)$ denote the weighted sum of all the interpretations in A_i, Then the PIE reduces to:

$$\mathrm{W}\Big(\bigcup_i A_i\Big) = \sum_{\emptyset\neq J\subseteq[n]} (-1)^{|J|+1}\mathrm{W}(A_J) \tag{2.21}$$

Similarly, when $\mathrm{W}(A_J)$ and $\mathrm{W}(A_{J'})$ are the same for each $m = |J|= |J'|$, we have that:

$$\mathrm{W}\Big(\bigcup_i A_i\Big) = \sum_{m=1}^{n}(-1)^{m+1}\binom{n}{m}\mathrm{W}(A_{[m]}) \tag{2.22}$$

Chapter 3

WFOMC in $\mathbf{FO^2}$ and $\mathbf{C^2}$

The work presented in this chapter has been previously published across the following publications:

Sagar Malhotra and Luciano Serafini. Weighted Model Counting in FO2 with Cardinality Constraints and Counting Quantifiers: A Closed Form Formula. *In proceedings of the AAAI Conference on Artificial Intelligence, 2022.*

Sagar Malhotra and Luciano Serafini. A Combinatorial Approach to Weighted Model Counting in the Two-Variable Fragment with Cardinality Constraints. *In proceedings of International Conference of the Italian Association for Artificial Intelligence, 2021*

3.1 Introduction

Most of the algorithms for WFOMC rely on a first-order logic variant of decision diagrams known as the FO d-DNNF [29, 33, 19, 34, 31]. In this chapter we present a completely combinatorial approach to WFOMC in *two variable fragment of* FOL (FO^2) *and its extension with counting quantifiers* (C^2). To this end, we introduce the notion of *2-type consistency*[1] with respect to a universally quantified FO^2 formula. Consistent 2-types characterize a universally quantified FO^2 formulas in terms of a subset of 2-types in the language $\mathcal{L}$. We use this notion to derive a closed-form for FOMC in the universally quantified fragment of FO^2 and its extensions with cardinality

[1] In our original papers, we used the name "lifted interpretations" for consistent 2-types

constraints, existential quantifiers, and counting quantifiers. The presented approach makes the following key contributions:

1. *The closed-form is easily extended to* FO^2 *with existential quantifiers, cardinality constraints and counting quantifiers, without losing domain-liftability.* A cardinality constraint on an interpretation is a constraint on the number of elements for which a certain predicate holds. Counting quantifiers admit expressions of the form $\exists^{\geq m} x \Phi(x)$ expressing that there exist at least m elements that satisfy $\Phi(x)$. Previous works have relied on Lagrange interpolation and Discrete Fourier Transform [19] for evaluating cardinality constraints. In this work, we deal with cardinality constraints in a completely combinatorial fashion.

2. *We provide a complete and uniform treatment of* WFOMC *in the two-variable fragment.* Multiple extensions of FO^2 have been proven to be domain liftable [35, 19, 36]. Most of these works rely extensively on a variety of logic-based algorithmic techniques. In this thesis, we provide a uniform and self-contained combinatorial treatment for all these extensions.

3. *The proposed closed form admits a class of weight functions strictly larger than symmetric weight functions.* The extended class of weight functions allows modelling the recently introduced count distributions.

Most of the chapter focuses on First-Order Model Counting (FOMC) i.e. counting the number of models of a formula Φ over a finite domain of size n denoted by $\textsc{fomc}(\Phi, n)$. We then show how WFOMC can be obtained by multiplying each term of the resulting formula for FOMC with the corresponding weight function. This allows us to separate the treatment of the counting part from the weighting part. The chapter is therefore structured as follows: We first present some background on the 2-variable fragments FO^2 and C^2, where we define some combinatorial notions essential to this chapter and the rest of the thesis. We first present our formulation of the closed-form formula for FOMC given in [28] for the universally quantified fragment of FO^2. We then present an alternative variant of this formula, that allows us to incorporate cardinality constraints. Using principle of inclusion-exclusion, we extend this approach to incorporate existential quantification. We then show how the aforementioned approach can be expanded to incorporate functionality constrains, i.e., a relation in the language represents a function. Finally, we extend this approach to counting quantifiers.

The last part of the chapter extends the formula for FOMC to WFOMC for the case of symmetric weight functions and for a larger class of weight functions that allow modeling count distributions [37].

3.2 Related Works

WFOMC for the purposes of probabilistic inference was independently defined and proposed by Gogate et al. [38] and Van den Broeck et al. [31]. Van den Broeck et al. [31] provide an algorithm for Symmetric-WFOMC over universally quantified theories based on *knowledge compilation* techniques. The notion of a *domain lifted theory* i.e. a first-order theory for which WFOMC can be computed in polynomial time w.r.t domain cardinality was first formalized by Van den Broeck in [29]. The same paper shows that a theory composed of a set of universally quantified clauses containing at most two variables is domain liftable. Van den Broeck et al.[36] extend this procedure to theories in full FO^2 (i.e. where existential quantification is allowed) by introducing a skolemization procedure for WFOMC.

Beame et al. [28] and Jaeger et al. [39, 40] investigate the theoretical aspects of WFOMC. Importantly, Beame et al. [28] show that there exists a formula in the three-variable fragment of FOL, where WFOMC cannot be computed in polynomial time and Jaeger [40] showed that even $\epsilon-$approximation of WFOMC is intractable. Beame et al. [28] also provide a closed-form formula for WFOMC in the universally quantified fragment of FO^2. Kuusisto and Lutz [35] extend the domain liftability results to FO^2 with a functionality axiom, and for sentences in *uniform one-dimensional fragment* U_1 [41]. They also propose a closed-form formula for WFOMC in FO^2 with functionality constraints. Recently, Kuželka [19] proposed a uniform treatment of WFOMC for FO^2 with cardinality constraints and counting quantifiers, proving these theories to be domain-liftable.

With respect to the state-of-the-art approaches to WFOMC, we propose an approach that provides a closed-form for WFOMC with cardinality constraints and counting quantifiers from which the PTIME data complexity is immediately evident. Moreover, Kuželka [19] relies on a sequence of reductions for proving domain liftability of counting quantifiers in the two variable fragment, on the other hand, our approach relies on a single reduction and exploits the principle of inclusion-exclusion to provide a closed-form formula for WFOMC. Finally, Kuželka [37] introduced Complex Markov Logic Networks, which use complex-valued weights and allow for full expressivity over a class of distributions called *count distributions*. We show in the last section of the chapter that our formalization is complete w.r.t. this class of distributions without using complex-valued weight functions.

3.3 The Two-Variable Fragments: FO^2 and C^2

FO^2 is a fragment of FOL, where the set of variables in $\mathcal{L}$ is restricted to have only two variables. In this thesis we will mainly be dealing with this fragment and its various extensions. From now on, we will use capital Greek letters to denote FOL formula. We will write $\Phi(x)$ to denote the fact that x is a free-variable in $\Phi(x)$. Hence, a sentence will be represented by capital Greek letter e.g. Ψ, Φ etc. C^2 is an extension of FO2, where besides the quantifiers $\forall$ and $\exists$, quantifiers like $\exists^{=k}$ (exist exactly k), $\exists^{<k}$ (exist less than k) and $\exists^{>k}$ (exist more than k) are admitted. We will now introduce the concepts of types and tables, which will serve us throughout this thesis.

Definition 22 (1-type)**.** *Given an* FOL *language* $\mathcal{L}$*, a* 1*-type is a maximally consistent conjunction of literals containing exactly one variable and no constants.*

We use u to denote the number of 1-types in $\mathcal{L}$. We assume an arbitrary ordering on the 1-types and denote the i^{th} 1-type in variable x as $i(x)$. Hence the set of all 1-types can be described by $[u]$.

Example 8 (1-type)**.** *Let us have an* FOL *language* $\mathcal{L}$ *consisting of relational symbols* $\mathcal{R} = \{A/1, R/2\}$*. Then following are all the 1-types in variable* x*.*

$$1(x): \ A(x) \ \wedge \ R(x,x) \tag{3.1}$$

$$2(x): \neg A(x) \ \wedge \ R(x,x) \tag{3.2}$$

$$3(x): \ A(x) \ \wedge \neg R(x,x) \tag{3.3}$$

$$4(x): \neg A(x) \ \wedge \neg R(x,x) \tag{3.4}$$

Definition 23 (2-table)**.** *Given an* FOL *language* $\mathcal{L}$*, a* 2*-table is a maximally consistent conjunction of literals containing exactly two distinct variable and no constants.*

We use b to denote the number of 2-tables. We assume an arbitrary order on the 2-table, hence the l^{th} 2-table is denoted by $l(x, y)$. Hence, the set of all 2-tables can be described by $[b]$.

Example 9 (2-table). *Let us have an* FOL *language* $\mathcal{L}$ *consisting of relational symbols* $\mathcal{R} = \{A/1, R/2\}$*. Then following are all the 2-tables.*

$$\begin{aligned}
1(x,y) &: R(x,y) \wedge R(y,x) \wedge (x \neq y) \\
2(x,y) &: \neg R(x,y) \wedge R(y,x) \wedge (x \neq y) \\
3(x,y) &: R(x,y) \wedge \neg R(y,x) \wedge (x \neq y) \\
4(x,y) &: \neg R(x,y) \wedge \neg R(y,x) \wedge (x \neq y)
\end{aligned}$$

Definition 24 (2-type). *Given an* FOL *language* $\mathcal{L}$*, a 2-type is a maximally consistent conjunction of literals containing at most two distinct variable and no constants. Equivalently a 2-type is a conjunction of the form:*

$$i(x) \wedge j(y) \wedge l(x,y)$$

where $i(x)$ *and* $j(y)$ *are the* i^{th} *and* j^{th} *1-types and* $l(x,y)$ *is the* l^{th} *2-table. We use* $ijl(x,y)$ *to denote a 2-type* $i(x) \wedge j(y) \wedge l(x,y)$*.*

In a given interpretation ω, we say a single constant c realizes the 1-type $i(x)$ if $\omega \models i(c)$. We say a pair of distinct domain constants (c,d) realize a 2-table $l(x,y)$ if $\omega \models l(c,d)$ and similarly (c,d) realizes the 2-type $ijl(x,y)$ if $\omega \models i(c) \wedge j(d) \wedge l(c,d)$.

1/2-types and 2-tables allow us to create complete description of an interpretation for evaluating any two variable formula on it. We formalize this notion in the following Lemmas.

Lemma 1. *In a given interpretation* ω*, a single domain constant* c *realizes one and only one 1-type.*

Proof. Assuming by contradiction that $\omega \models i(c)$ and $\omega \models j(c)$ for a pair of distinct 1-types i.e. $i \neq j$. Hence, we have that $\omega \models i(c) \wedge j(c)$. Since 1-types $i(x)$ and $j(x)$ are a maximally consistent conjunction of literals, hence so are $i(c)$ and $j(c)$. Hence, $i(c) \wedge j(c) = \perp$. Hence, we have that $\omega \models \perp$, which is a contradiction. □

Definition 25. *[1-type Cardinality Vector] An interpretation* ω *is said to have the 1-type cardinality vector* $\boldsymbol{k} = \langle k_1, \dots, k_u \rangle$ *if for all* $i \in [u]$ *it has* k_i *domain elements* c *such that* $\omega \models i(c)$*, where* $i(x)$ *is the* i^{th} *1-type. If* ω *has 1-type cardinality vector* $\boldsymbol{k}$*, then we say that* $\omega \models \boldsymbol{k}$*.*

Notice that by construction we have that, for a given $\boldsymbol{k}$, $\sum_i k_i = n$, where n is the domain cardinality.

Lemma 2. *In a given interpretation ω, a pair of distinct domain constants (c, d) realizes one and only one 2-table.*

Proof. Assuming by contradiction that $\omega \models l(c, d)$ and $\omega \models l'(c, d)$ for a pair of distinct 2-tables i.e. $l \neq l'$. Hence, we have that $\omega \models l(c, d) \wedge l'(c, d)$. Since 2-tables $l(x, y)$ and $l'(x, y)$ are a maximally consistent conjunction of literals, hence so are $l(c, d)$ and $l'(c, d)$. Hence, $l(c, d) \wedge l'(c, d) = \perp$. Hence, we have that $\omega \models \perp$, which is a contradiction. □

Lemma 3. *In a given interpretation ω, a pair of distinct domain constants (c, d) realizes one and only one 2-type.*

Proof. Assuming by contradiction that $\omega \models ijl(c, d)$ and $\omega \models i'j'l'(c, d)$ for a pair of distinct 2-types, i.e., $i \neq i'$ or $j \neq j'$ or $l \neq l'$. Hence, $\omega \models ijl(c, d) \wedge i'j'l'(c, d)$. But due to Lemma 1 and Lemma 2, this leads to a contradiction. □

Definition 26. *[2-type Cardinality Vector] An interpretation ω is said to have the 2-type cardinality vector $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$, where $\boldsymbol{h}$ is a vector with components h_{ijl} and $i \leq j$, representing the number of unordered pair of distinct domain constants $\{c, d\}$ such that $\omega \models ijl(c, d)$ or $\omega \models ijl(d, c)$, and $\boldsymbol{k}$ is the 1-type cardinality vector of ω. If ω has 2-type cardinality vector $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$, then we say $\omega \models \langle \boldsymbol{k}, \boldsymbol{h} \rangle$. Given a $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$, we also use $\boldsymbol{h}_{ij}$ to denote the vector $\langle h_{ij1} \cdots h_{ijb} \rangle$.*

With an abuse of notation, if ω is a model of Φ, and $\omega \models \boldsymbol{k}$ (resp. $\omega \models \langle \boldsymbol{k}, \boldsymbol{h} \rangle$) , then we denote this by $\omega \models \Phi \wedge \boldsymbol{k}$ (resp. $\omega \models \Phi \wedge \langle \boldsymbol{k}, \boldsymbol{h} \rangle$).

3.4 FOMC for Universally Quantified FO^2

Universally quantified FO^2 formulas are formulas of the form $\forall xy.\Phi(x, y)$, where $\Phi(x, y)$ is quantifier-free. In this section, we first present a re-formulated version of closed form formula for FOMC of universally quantified FO^2 formulas as presented in [28]. We then present an alternative — combinatorially more informative — variant of this closed-form.

Definition 27. *Given a universally quantified FO^2 formula $\forall xy.\Phi(x, y)$, where $\Phi(x, y)$ is quantifier-free. We define $\Phi(\{x, y\})$ as follows:*

$$\Phi(\{x, y\}) := \Phi(x, x) \wedge \Phi(x, y) \wedge \Phi(y, x) \wedge \Phi(y, y) \tag{3.5}$$

Example 10. *Given an* FOL *language* $\mathcal{L}$ *with relational symbols* $\{R/2, A/1\}$*, let* $\Phi(x, y) = A(x) \land R(x, y) \rightarrow A(y)$*, then* $\Phi(\{x, y\})$ *is the following formula*

$$\begin{aligned} &(A(x) \land R(x, x) \rightarrow A(x)) \\ \land\ &(A(x) \land R(x, y) \rightarrow A(y)) \\ \land\ &(A(y) \land R(y, x) \rightarrow A(x)) \\ \land\ &(A(y) \land R(y, y) \rightarrow A(y)) \end{aligned} \tag{3.6}$$

We now define the notion of 2-type consistency with respect to a universally quantified FO^2 formula.

Definition 28 (2-Type Consistency)**.** *Given a universally quantified* FO^2 *formula* $\forall xy.\Phi(x, y)$*, a 2-type is consistent with* $\forall xy.\Phi(x, y)$ *if:*

$$ijl(x, y) \models \Phi(\{x, y\}) \tag{3.7}$$

where the entailment in equation 3.7 is checked by assuming a propositional language consisting of only the constant-free literals in $\mathcal{L}$*.*

Example 11. *The following is an example of a consistent 2-type for the formula* (3.6) *of Example 10:*

$$\tau(x, y) := \neg A(x) \land R(x, x) \land \neg A(y) \land R(y, y) \land \neg R(x, y) \land R(y, x) \tag{3.8}$$

It is easy to see that, assuming a propositional language consisting of constant-free literals in $\mathcal{L}$*, i.e., with propositional variables* $\{A(x), A(y), R(x, x), R(y, y), R(x, y), R(y, x)\}$*, that:*

$$\tau(x, y) \models \Phi(\{x, y\})$$

Lemma 4. *If a 2-type* $ijl(x, y)$ *is not consistent (inconsistent) with a universally quantified* FO^2 *formula* $\forall xy.\Phi(x, y)$ *then:*

$$ijl(x, y) \models \neg\Phi(\{x, y\}) \tag{3.9}$$

Proof. Since a 2-type $ijl(x, y)$ is a maximally consistent conjunction of literals with at most two variables, it forms a maximally consistent theory in the propositional language consisting of the constant free literals in $\mathcal{L}$. Hence, for any 2-type if $ijl(x, y) \not\models \Phi(\{x, y\})$, then $ijl(x, y) \models \neg\Phi(\{x, y\})$. □

Lemma 5. *Given a universally quantified* FO^2 *sentence* $\forall xy.\Phi(x,y)$ *and an interpretation* ω*, then* $\omega \models \forall xy.\Phi(x,y)$ *if and only if for all pairs of domain constants* (c,d) *in the domain,* $\omega \models ijl(c,d)$ *only if* $ijl(x,y)$ *is consistent with* $\forall xy.\Phi(x,y)$*.*

Proof. Grounding of $\forall xy.\Phi(x,y)$ over the domain Δ can be written as follows:

$$\bigwedge_{\substack{\{c,d\}\subseteq\Delta \\ c \neq d}} \Phi(\{c,d\}) \tag{3.10}$$

If $\omega \models \forall xy.\Phi(x,y)$, then equivalently:

$$\omega \models \bigwedge_{\substack{\{c,d\}\subseteq\Delta \\ c \neq d}} \Phi(\{c,d\}) \tag{3.11}$$

Assuming to the contrary let's say there exists a pair of constants (c,d), such that (c,d) realizes the 2-type $ijl(x,y)$, i.e., $\omega \models ijl(c,d)$, but $ijl(x,y)$ is inconsistent with $\forall xy.\Phi(x,y)$. Due to Lemma 4, if $ijl(x,y) \not\models \Phi(\{x,y\})$, then $ijl(x,y) \models \neg\Phi(\{x,y\})$. Hence, $ijl(c,d) \models \neg\Phi(\{c,d\})$. Hence, we have that $\omega \models \neg\Phi(\{c,d\})$, which is a contradiction to expression (3.11) and hence to the fact that $\omega \models \forall xy.\Phi(x,y)$. $\square$

Lemma 6. *Given an* FOL *sentence* Φ*, then the following always holds.*

$$\mathrm{FOMC}(\Phi, n) = \sum_{\boldsymbol{k}} \mathrm{FOMC}(\Phi, \boldsymbol{k}) \tag{3.12}$$

where $\boldsymbol{k}$ *are the 1-type cardinality vectors (see Definition 25) and* $\mathrm{FOMC}(\Phi,\boldsymbol{k})$ *is defined as the number of models such that* $\omega \models \Phi \wedge \boldsymbol{k}$*.*

Proof. Using Lemma 1, we cannot have that for any interpretation ω, $\omega \models \boldsymbol{k}$ and $\omega \models \boldsymbol{k}'$, where $\boldsymbol{k}$ and $\boldsymbol{k}'$ are distinct 1-type cardinality vectors. Also, for each interpretation ω, $\omega \models \boldsymbol{k}$, for some 1-type cardinality $\boldsymbol{k}$. Hence, we have that:

$$\{\omega : \omega \models \Phi\} = \biguplus_{\boldsymbol{k}} \{\omega : \omega \models \Phi \wedge \boldsymbol{k}\} \tag{3.13}$$

where $\biguplus$ represents disjoint union. Therefore:

$$|\{\omega : \omega \models \Phi\}| = \sum_{\boldsymbol{k}} |\{\omega : \omega \models \Phi \wedge \boldsymbol{k}\}| \tag{3.14}$$

$\square$

Using Lemma 6, we can decompose the model counting problem into model counting w.r.t the 1-type cardinality vectors $\boldsymbol{k}$. Furthermore, we have the following corollary:

Corollary 1 (of Lemma 6)**.** *Given an* FOL *sentence* Φ*, if* $\text{FOMC}(\Phi, \boldsymbol{k})$ *is domain liftable then so is* $\text{FOMC}(\Phi, n)$*.*

Proof. Using Lemma 6, we have that $\text{FOMC}(\Phi, n)$ can be decomposed into $\text{FOMC}(\Phi, \boldsymbol{k})$. Irrespective of Φ, $\boldsymbol{k}$ ranges over all possible u-tuples $\langle k_1 \cdots k_u \rangle$, where $\sum_i k_i = n$. Using stars and bars method, the number of such u-tuples is given as $\binom{n+u-1}{u-1}$, which is bounded above by n^u, where u is the number of 1-types in the language. Hence, we have that there are only polynomial $O(n^u)$ summands in the summation $\sum_{\boldsymbol{k}} \text{FOMC}(\Phi, \boldsymbol{k})$. Hence, if $\text{FOMC}(\Phi, \boldsymbol{k})$ can be computed in polynomial time w.r.t n, say $O(n^c)$, for some positive constant c, then $\text{FOMC}(\Phi, n)$ can be computed in time $O(n^u) \cdot O(n^c) = O(n^{u+c})$, which is polynomial in n. □

Hence, we can worry about only showing the closed-forms and domain-liftability for $\text{FOMC}(\Phi, \boldsymbol{k})$, as domain-liftability and closed-forms for $\text{FOMC}(\Phi, n)$ automatically follow from Corollary 1 and Lemma 6 respectively. In the following we provide the formula for computing $\text{FOMC}(\forall xy.\Phi(x, y), \boldsymbol{k})$. Althoug, the proof ideas are based on the derivation of $\text{FOMC}(\Phi, n)$ as presented in [28], the concepts used in the following proof rely on weighted types and tables, rather than "cells" and their probabilities as introduced in [28]. We will use the introduced notation and concepts throughtout.

Theorem 7 (Beame et al. (2015)[28] reformulated)**.** *Given a universally quantified* FO^2 *formula* $\forall xy.\Phi(x, y)$*, interpreted over a domain* Δ *of size* n*, then the model count of the models* ω *such that* $\omega \models \forall xy.\Phi(x, y)$ *and* ω *has the 1-type cardinality* $\boldsymbol{k}$ *is given as:*

$$\text{FOMC}(\forall xy.\Phi(x, y), \boldsymbol{k}) = \binom{n}{\boldsymbol{k}} \prod_{i \leq j \in [u]} n_{ij}^{\boldsymbol{k}(i,j)} \tag{3.15}$$

where $\boldsymbol{k}(i, j)$ *is defined as follows:*

$$\boldsymbol{k}(i, j) = \begin{cases} \frac{k_i(k_i-1)}{2} & \text{if } i = j \\ k_i k_j & \text{otherwise} \end{cases} \tag{3.16}$$

and $n_{ij} = \sum_{l \in [b]} n_{ijl}$*, where* n_{ijl} *is* 1 *if* $ijl(x, y) \models \Phi(\{x, y\})$ *and* 0 *otherwise.*

Proof. Our goal is to count the models ω over the domain *Delta* of size n, such that $\omega \models \forall xy.\Phi(x, y) \wedge \boldsymbol{k}$. Given a domain Δ, the number of interpretations with 1-type cardinality vector $\boldsymbol{k}$ is given as:

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

If $\omega \models \boldsymbol{k}$ then we have $k_i k_j$ pairs of domain constants (c, d) such that $\omega \models i(c) \wedge j(d)$, where $i \neq j$. Due to Lemma 5 each such pair of domain constants can realize a 2-table $l(x, y)$ if $ijl(x, y) \models \Phi(\{x, y\})$. Hence, we have $\sum_{l \in [b]} n_{ijl} = n_{ij}$ independent and mutually-exclusive choices for assigning the 2-tables to each such (c, d). Hence, when $i \neq j$, the number of ways of assigning 2-table to the pairs of domain constants realizing the i^{th} and the j^{th} 1-type respectively is given as:

$$n_{ij}^{k_i k_j}$$

The pair of domain constants (c, d) such that $\omega \models i(c) \wedge i(d)$ is given as $\binom{k_i}{2}$. Using the same reasoning as above, we have $\sum_{l \in [b]} n_{iil} = n_{ii}$ independent and mutually-exclusive choices for assigning the 2-tables to each such (c, d). Hence, when $i = j$, the number of ways of assigning 2-table to the pairs of domain constants realizing the i^{th} 1-types is given as:

$$n_{ii}^{\binom{k_i}{2}}$$

Hence, the possible 2-table assignments to the pair of domain elements realizing the i^{th} and j^{th} 1-type respectively is given as:

$$n_{ij}^{k(i,j)}$$

Hence the number of models ω such that $\omega \models \forall xy.\Phi(x, y) \wedge \boldsymbol{k}$ is given as:

$$\binom{n}{\boldsymbol{k}} \prod_{i \leq j \in [u]} n_{ij}^{\boldsymbol{k}(i,j)}$$

□

It can be seen that (3.15) can be computed in polynomial time w.r.t n. Hence, using Corollary 1, we have that FOMC for universally quantified FO^2 formulas is domain-liftable.

Example 12 (Example 10 continued)**.** *Consider a domain of 3 elements (i.e., $n = 3$). The formula* (3.15) *takes the form:*

$$\binom{3}{k_0, k_1, k_2, k_3} \prod_{i=0}^{3} n_{ii}^{\frac{k_i(k_i-1)}{2}} \prod_{\substack{i<j \\ i=0}}^{3} n_{ij}^{k_i k_j}$$

which is the number of models with k_0 elements for which $A(x)$ and $R(x,x)$ are both false; k_1 elements for which $A(x)$ is false and $R(x,x)$ true, k_2 elements for which $A(x)$ is true and $R(x,x)$ is false and k_3 elements for which $A(x)$ and $R(x,x)$ are both true. For instance: $\binom{3}{2,0,0,1} n_{00}^1 n_{03}^2 = \binom{3}{2,0,0,1} 4^1 \cdot 2^2 = 3 \cdot 16 = 48$ *is the number of models in which 2 elements are such that $A(x)$ and $R(x,x)$ are false and* 1 *element such that $A(x)$ and $R(x,x)$ are both true.*

We now move on to demonstrating an extension of formula (3.15), which will be later exploited to incorporate cardinality constraints and counting quantifiers. We will also see that this formula motivates a very general class of weight functions for WFOMC in the 2-variable fragment. The key extension this formula provides is simple, instead of decomposing FOMC over 1-type cardinality vectors $\boldsymbol{k}$, we decompose it over the 2-type cardinality vectors $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$. We first provide a lemma and corollary analogous to Lemma 6 and Corollary 1, for the 2-type cardinality vectors $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$.

Lemma 7. *Given an* FOL *sentence Φ, then the following always holds.*

$$\text{FOMC}(\Phi, n) = \sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle} \text{FOMC}(\Phi, \langle \boldsymbol{k}, \boldsymbol{h} \rangle) \tag{3.17}$$

where $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$ are the 2-type cardinality vectors (see Definition 26) and $\text{FOMC}(\Phi, \langle \boldsymbol{k}, \boldsymbol{h} \rangle)$ *is defined as the number of models such that $\omega \models \Phi \wedge \langle \boldsymbol{k}, \boldsymbol{h} \rangle$.*

Proof. Using Lemma 3, we cannot have that for any interpretation ω, $\omega \models \langle \boldsymbol{k}, \boldsymbol{h} \rangle$ and $\omega \models \langle \boldsymbol{k}', \boldsymbol{h}' \rangle$, where $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$ and $\langle \boldsymbol{k}', \boldsymbol{h}' \rangle$ are distinct 2-type cardinality vectors. Also, for each interpretation ω, $\omega \models \langle \boldsymbol{k}, \boldsymbol{h} \rangle$, for some 2-type cardinality vector $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$. Hence, we have that:

$$\{\omega : \omega \models \Phi\} = \biguplus_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle} \{\omega : \omega \models \Phi \wedge \langle \boldsymbol{k}, \boldsymbol{h} \rangle\} \tag{3.18}$$

where $\biguplus$ represents disjoint union. Therefore:

$$|\{\omega : \omega \models \Phi\}| = \sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle} |\{\omega : \omega \models \Phi \wedge \langle \boldsymbol{k}, \boldsymbol{h} \rangle\}| \tag{3.19}$$

□

Lemma 7, leads to the following corollary about decomposition of domain liftaibility over the 2-type cardinality vectors $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$

Corollary 2 (of Lemma 7). *Given an* FOL *sentence* Φ*, if* $\textsc{fomc}(\Phi, \langle \boldsymbol{k}, \boldsymbol{h} \rangle)$ *is domain liftable then so is* $\textsc{fomc}(\Phi, n)$.

Proof. The proof idea is similar to the one of Corollary 1, i.e., we show that there are only polynomially many values of $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$ in an FOL language, irrespective of Φ. We have $O(n^u)$ 1-type vectors $\boldsymbol{k}$. Hence, we only need to show that given a $\boldsymbol{k}$, there are only polynomially many $\boldsymbol{h}$. If an $\omega \models \boldsymbol{k}$, then we have $\boldsymbol{k}(i,j)$ (see equation (3.16)) pairs of distinct domain constant (c,d) realizing the 1-type i and j respectively. Each such pair can be extended to a 2-type by realizing any of the possible b 2-tables. Hence, we have $\binom{\boldsymbol{k}(i,j)}{\boldsymbol{h}_{ij}}$ possible choices for realizing the two tables for the $\boldsymbol{k}(i,j)$ pairs. Now using that stars and bars method, the number of possible $\boldsymbol{h}_{ij}$ values is $\binom{\boldsymbol{k}(i,j)+b-1}{b-1}$. This is bounded above by $O(\boldsymbol{k}(i,j)^b)$, which is bounded above by $O(n^{2b})$, as the $\boldsymbol{k}(i,j)$ cannot have a larger value than n^2 — as the number of possible pairs of distinct domain constants is only $\binom{n}{2}$. Finally, we have only $\frac{u(u+1)}{2}$ possible pairs of 1-types i and j, such that $i \leq j$. Hence, given a 1-type cardinality vector $\boldsymbol{k}$, the vector $\boldsymbol{h}$ can take $\frac{u(u+1)}{2}O(n^{2b}) = O(n^{2b})$ values. The total possible values of $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$ are bounded above by the upper-bound for total possible values of $\boldsymbol{k}$ times the upper-bound for total possible $\boldsymbol{h}$. Hence, the total possible values of $\boldsymbol{k}$ are bounded by $O(n^u) \cdot O(n^{2b}) = O(n^{u+2b})$. Hence using Lemma 7, if $\textsc{fomc}(\Phi, \langle \boldsymbol{k}, \boldsymbol{h} \rangle)$ can be computed in $O(n^c)$ for some constant c, then $\textsc{fomc}(\Phi, n)$ can be computed in $O(n^{c+u+2b})$. □

We now present the closed-form formula for $\textsc{fomc}(\Phi, \langle \boldsymbol{k}, \boldsymbol{h} \rangle)$.

Theorem 8. *Given a universally quantified* FO^2 *formula* $\forall xy.\Phi(x,y)$*, interpreted over a domain* Δ *of size* n*, then the model count of the models* ω *such that* $\omega \models \forall xy.\Phi(x,y)$ *and* ω *has the 2-type cardinality* $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$ *is given as:*

$$\textsc{fomc}(\forall xy.\Phi(x,y), \langle \boldsymbol{k}, \boldsymbol{h} \rangle) = \binom{n}{\boldsymbol{k}} \prod_{i \leq j \in [u]} \binom{\boldsymbol{k}(i,j)}{\boldsymbol{h}_{ij}} \prod_{l \in [b]} n_{ijl}^{h_{ijl}} \tag{3.20}$$

where $\boldsymbol{k}(i,j)$ *is defined as follows:*

$$\boldsymbol{k}(i,j) = \begin{cases} \frac{k_i(k_i-1)}{2} & \textit{if } i = j \\ k_i k_j & \textit{otherwise} \end{cases} \tag{3.21}$$

where n_{ijl} is 1 *if* $ijl(x,y) \models \Phi(\{x,y\})$ *and* 0 *otherwise.*

Proof. We first enumerate the number of models ω such that $\omega \models \langle \boldsymbol{k}, \boldsymbol{h} \rangle$. Given a 1-type cardinality vector $\boldsymbol{k}$, we have $\binom{n}{\boldsymbol{k}}$ choices for assigning 1-types to domain constants. Given that $\omega \models \boldsymbol{k}$, it has $\boldsymbol{k}(i,j)$ pairs of distinct domain constants (c,d) such that $i(c)$ and $j(d)$, each such pair can be extended to any of the possible b 2-tables, such that we have h_{ijl} pairs with 2-type $ijl(x,y)$. Hence, we have $\binom{\boldsymbol{k}(i,j)}{\boldsymbol{h}_{ij}}$ choices for assigning 2-tables, where $\boldsymbol{h}_{ij} = \langle h_{ij1}, \cdots, h_{ijb} \rangle$. Hence, we have that the number of models ω, such that $\omega \models \langle \boldsymbol{k}, \boldsymbol{h} \rangle$ is given as:

$$\binom{n}{\boldsymbol{k}} \prod_{i \leq j \in [u]} \binom{\boldsymbol{k}(i,j)}{\boldsymbol{h}_{ij}} \tag{3.22}$$

Finally, in order to count the models such that $\omega \models \forall xy.\Phi(x,y) \wedge \langle \boldsymbol{k}, \boldsymbol{h} \rangle$, we can only admit ω that have two types consistent with $\forall xy.\Phi(x,y)$. We introduce an indicator variable n_{ijl}, which is 1 if $ijl(x,y) \models \Phi(\{x,y\})$ and 0 otherwise. We multiply expression (3.22) with n_{ijl} for each realization of a 2-type $ijl(x,y)$, hence by $\prod_{l \in [b]} n_{ijl}^{h_{ijl}}$. Giving us expression (3.20).

Notice that the indicator variables n_{ijl} act as a filter for selecting and deselecting models or $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$ that need to be counted, a single realization of an inconsistent 2-type leads to an $n_{ijl} = 0$ being present in the product $\prod_{l \in [b]} n_{ijl}^{h_{ijl}}$, hence leading to a 0 contribution for that model or for that $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$ in FOMC$(\forall xy.\Phi(x,y), \langle \boldsymbol{k}, \boldsymbol{h} \rangle)$.

□

It can be seen that equation (3.20), can be computed in polynomial time w.r.t the domain cardinality n. Hence, using Corollary 2, we have that FOMC for universally quantified formulas in FO^2 is domain liftable, under this new formulation of the FOMC formula.

3.5 FOMC for Cardinality Constraints

Cardinality constraints as defined in Definition 15 are arithmetic expressions that impose restrictions on the number of times a certain predicate is interpreted to be true. A simple example of a cardinality constraint is $|A| = m$, for some unary predicate A and positive integer m. This cardinality constraint is satisfied by any interpretation in which $A(c)$ is interpreted to be true for exactly m distinct constants c in the domain Δ. A more complex example of a cardinality constraint could be: $|A| + |B| \leq |C|$, where A, B and C are some predicates in the language.

Lemma 8. *Given the interpretations ω_1 and ω_2, having the same 2-type cardinality vector, i.e., $\omega_1 \models \langle \boldsymbol{k}, \boldsymbol{h} \rangle$ and $\omega_2 \models \langle \boldsymbol{k}, \boldsymbol{h} \rangle$, then for any cardinality constraint Γ :*

$$\omega_1 \models \Gamma \leftrightarrow \omega_2 \models \Gamma$$

Proof. The proof is a consequence of the fact that $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$ uniquely defines the cardinality of the predicates. Let us assume to the contrary that $\omega_1 \models \langle \boldsymbol{k}, \boldsymbol{h} \rangle$ and $\omega_2 \models \langle \boldsymbol{k}, \boldsymbol{h} \rangle$, but $\omega_1 \models (|A| = l)$ and $\omega_2 \models (|A| = l')$, where $l \neq l'$. Let $s \subset [u]$ represents the set of 1-types, such that if $i \in s$, then $i(x) \models A(x)$. If A is a unary predicate then due to Lemma 1, we have that for ω_1 $\sum_{i \in s} k_i = l$. Similarly, for ω_2 we have that $\sum_{i \in s} k_i = l'$. But ω_1 and ω_2 have the same 1-type cardinality vectors, hence $l = l'$, which is a contradiction.

We now repeat the same argument for binary predicates, let R be a binary predicate, and assume to the contrary that $\omega_1 \models \langle \boldsymbol{k}, \boldsymbol{h} \rangle$ and $\omega_2 \models \langle \boldsymbol{k}, \boldsymbol{h} \rangle$, but $\omega_1 \models (|R| = q)$ and $\omega_2 \models (|R| = q')$, where $q \neq q'$. Let $s \subset [u]$, be the set of 1-types such that if $i \in s$, then $i(x) \models R(x,x)$. Let $f_1 \subseteq [b]$ be the set of 2-tables such that if $l \in f_1$, then $l(x,y) \models R(x,y) \wedge \neg R(x,y)$ or $l(x,y) \models \neg R(x,y) \wedge R(x,y)$. Let $f_2 \subseteq [b]$ be the set of 2-tables such that if $l \in f_2$, then $l(x,y) \models R(x,y) \wedge R(y,x)$. Then using Lemma 1 and Lemma 2, we have that for ω_1: $\sum_{i \in s} k_i + \sum_{l \in f_1} h_{ijl} + \sum_{l \in f_2} 2h_{ijl} = q$, and for ω_2: $\sum_{i \in s} k_i + \sum_{l \in f_1} h_{ijl} + \sum_{l \in f_2} 2h_{ijl} = q'$. But both ω_1 and ω_2 have the same 2-type cardinality vector, hence $q = q'$, which is a contradiction. □

Example 13. *Consider $\Phi(x,y)$ as given in Example 10, we wish to compute FOMC of $\forall xy.\Phi(x,y)$ with the additional conjunct $|A| = 2$ and $|R| = 2$, i.e., $\forall xy.\Phi(x,y) \wedge (|A| = 2) \wedge (|R| = 2)$. The constraint $|A| = 2$ implies that when computing FOMC we have to consider $\boldsymbol{k}$ such that $k_2 + k_3 = 2$. $|R| = 2$ constraint translates to only considering $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$ with $k_1 + k_3 + \sum_{i \leq j}(h_1^{ij} + h_2^{ij} + 2h_3^{ij}) = 2$. Hence, FOMC for $\forall xy.\Phi(x,y) \wedge (|A| = 2) \wedge (|R| = 2)$ can be written as:*

$$\sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle \models \Gamma_1 \wedge \Gamma_2} \textsc{fomc}(\forall xy.\Phi(x,y), \langle \boldsymbol{k}, \boldsymbol{h} \rangle) \tag{3.23}$$

where Γ_1 is the constraint such that $|A| = 2$ and Γ_2 is the constraint such that $|R| = 2$. Both these constraints can be checked in polynomial time w.r.t domain cardinality, from just the $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$, which are polynomially many.

For a given $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$, we use the notation $\boldsymbol{k}(A)$ to denote cardinality of A if A is unary and $\langle \boldsymbol{k}, \boldsymbol{h} \rangle(A)$ if A is binary. Using Lemma 8, we can conclude that $\textsc{fomc}(\Phi \wedge \Gamma, n)$ where Φ is a pure universal formula with 2 variables can be computed by considering only the $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$'s that satisfy Γ, i.e., those $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$'s where Γ evaluates to true, when $|P|$ is substituted with $\langle \boldsymbol{k}, \boldsymbol{h} \rangle(P)$ when P is binary and $\boldsymbol{k}(P)$ when P is unary.

Theorem 9. *Given a universally quantified* FO^2 *formula* Φ *and cardinality constraint* Γ*, then:*

$$\text{FOMC}(\Phi \wedge \Gamma, n) = \sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle \models \Gamma} \text{FOMC}(\Phi, \langle \boldsymbol{k}, \boldsymbol{h} \rangle) \tag{3.24}$$

Proof. Due to Lemma 8, if ω_1 and ω_2 have the same 2-type cardinality vector then they agree on any cardinality constraint. Hence,

$$\{\omega : \omega \models \Gamma\} = \{\omega : \omega \models \langle \boldsymbol{k}, \boldsymbol{h} \rangle \text{ and } \langle \boldsymbol{k}, \boldsymbol{h} \rangle \models \Gamma\} \tag{3.25}$$

Hence, we have that:

$$\text{FOMC}(\Phi \wedge \Gamma, n) = \sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle \models \Gamma} \text{FOMC}(\Phi, \langle \boldsymbol{k}, \boldsymbol{h} \rangle)$$

□

3.6 FOMC for Existential Quantifiers

Scott [42] proposed an equi-satisfiable normal form for an FO^2 formula, largely known as the Scott's Normal Form(SNF). In [35], the authors show that every interpretation of a formula in FO^2, can be extended to a unique interpretation of its SNF reduction. They further show that SNF has no more interpretations than the original FO^2 formula.

Theorem 10 (Scott's Normal Form [42] and [35])**.** *For every* FO^2 *sentence* Γ *in a language* $\mathcal{L}$*, an equi-satisfiable sentence* $\mathrm{SNF}(\Gamma)$ *in* $\mathcal{L}_{ext}$ *can be constructed, where* $\mathcal{L}_{ext}$ *is an extension of* $\mathcal{L}$ *with new predicates, and* $\mathrm{SNF}(\Gamma)$ *has the following form:*

$$\mathrm{SNF}(\Gamma) := \forall xy.\Phi(x, y) \wedge \bigwedge_{i=1}^{m} \forall x \exists y.\Psi_i(x, y) \tag{3.26}$$

where $\Phi(x, y)$ *and* $\Psi_i(x, y)$ *are quantifier-free, such that for every* $\mathcal{L}$*-interpretation* $\omega \models \Gamma$*, there exists a unique* $\mathcal{L}_{ext}$*-interpretation* $\omega_{ext} \models \mathrm{SNF}(\Gamma)$*, and for every* $\mathcal{L}_{ext}$*-interpretation* $\omega_{ext} \models \mathrm{SNF}(\Gamma)$*,* $\omega_{ext} \downarrow \mathcal{L} \models \Gamma$*.*

In this section, we provide a proof for model counting in the presence of existential quantifiers. The key difference in our approach w.r.t [28] is that we make explicit use of the principle of inclusion-exclusion, and we will later generalize the same approach to counting quantifiers. We will first provide a corollary of the principle of inclusion-exclusion.

Corollary 3 ([43] section 4.2)**.** *Let Ω be a set of objects and let $\mathcal{S} = \{S_1, \ldots, S_m\}$ be a set of subsets of Ω. For every $\mathcal{Q} \subseteq \mathcal{S}$, let $N(\supseteq \mathcal{Q})$ be the count of objects in Ω that belong to all the subsets $S_i \in \mathcal{Q}$, i.e., $N(\supseteq \mathcal{Q}) = \left|\{\bigcap_{S_i \in Q} S_i\}\right|$. For every $0 \leq l \leq m$, let $s_l = \sum_{|\mathcal{Q}|=l} N(\supseteq \mathcal{Q})$ and let e_0 be count of objects that do not belong to any of the S_i in $\mathcal{S}$, then*

$$e_0 = \sum_{l=0}^{m} (-1)^l s_l \tag{3.27}$$

Theorem 11. *For an* FO^2 *formula in Scott's Normal Form as given in* (3.26)*, let $\Phi' = \forall xy.(\Phi(x,y) \wedge \bigwedge_{i=1}^{q} P_i(x) \to \neg\Psi_i(x,y))$ where P_i's are fresh unary predicates, then:*

$$\textsc{fomc}((3.26), n) = \sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle} (-1)^{\sum_i \boldsymbol{k}(P_i)} \textsc{fomc}(\Phi', \langle \boldsymbol{k}, \boldsymbol{h} \rangle) \tag{3.28}$$

Proof. Let Ω be the set of models of $\forall xy.\Phi(x,y)$ over the language of Φ and $\{\Psi_i\}$ (i.e., the language of Φ' excluding the predicates P_i) and on a domain Δ consisting of n elements. Let $\mathcal{S} = \{\Omega_{ci}\}_{c \in \Delta,\ 1 \leq i \leq q}$ be the set of subsets of Ω where Ω_{ci} is the set of ω such that $\omega \models \forall y.\neg\Psi_i(c,y)$. For every model ω of (3.26), $\omega \not\models \forall y \neg\Psi_i(c,y)$ for any pair of i and c i.e. ω is not in any Ω_{ci}. Also, for every $\omega \in \Omega$, if $\omega \notin \Omega_{ci}$ for any pair of i and c, then $\omega \models \exists y.\Psi_i(c,y)$ for all i and for all $c \in \Delta$ i.e., $\omega \models \bigwedge_{i=1}^{q} \forall x \exists y.\Psi_i(x,y)$. Hence, $\omega \models$ (3.26) if and only if $\omega \notin \Omega_{ci}$ for all c and i. Therefore, the count of models of (3.26) is equal to the count of models in Ω which do not belong to any Ω_{ci}. Hence, If we are able to compute s_l (as introduced in Corollary 3), then we could use Corollary 3 for computing cardinality of all the models which do not belong to any Ω_{ci} and hence $\textsc{fomc}((3.26), n)$.

For every $0 \leq l \leq n \cdot q$, let us define

$$\Phi'_l = \Phi' \wedge \sum_{i=1}^{q} |P_i| = l \tag{3.29}$$

We will now show that s_l is exactly given by $\textsc{fomc}((3.29), n)$.

Every model of Φ'_l is an extension of an $\omega \in \Omega$ that belongs to at least l elements in $\mathcal{S}$. In fact, for every model ω of $\forall xy.\Phi(x,y)$ i.e. $\omega \in \Omega$, if $\mathcal{Q}'$ is the set of elements of $\mathcal{S}$ that contain ω, then ω can be extended into a model of Φ'_l in $\binom{|\mathcal{Q}'|}{l}$ ways. Each such model can be obtained by choosing l elements in Q' and interpreting $P_i(c)$ to be true in the extended model, for each of the l chosen elements $\Omega_{ci} \in Q'$. On the other hand, recall that $s_l = \sum_{|\mathcal{Q}|=l} N(\supseteq Q)$. Hence, for any $\omega \in \Omega$ if $\mathcal{Q}'$ is the set of elements of $\mathcal{S}$ that contain ω, then there are $\binom{|\mathcal{Q}'|}{l}$ distinct subsets $\mathcal{Q} \subseteq \mathcal{Q}'$ such that $|\mathcal{Q}|= l$. Hence, we have that ω contributes $\binom{|\mathcal{Q}'|}{l}$ times to s_l. Therefore, we can conclude that

$$s_l = \text{FOMC}(\Phi'_l, n) = \sum_{|\mathcal{Q}|=l} N(\supseteq Q)$$

and by the principle of inclusion-exclusion as given in Corollary 3, we have that :

$$\begin{aligned}
\text{FOMC}((3.26), n) = e_0 &= \sum_{l=0}^{n\cdot q}(-1)^l s_l \\
&= \sum_{l=0}^{n\cdot q}(-1)^l \text{FOMC}(\Phi'_l, n) \\
&= \sum_{l=0}^{n\cdot q}(-1)^l \sum_{\langle \boldsymbol{k},\boldsymbol{h}\rangle \models \sum_i |P_i|=l} \text{FOMC}(\Phi', \langle \boldsymbol{k}, \boldsymbol{h}\rangle) \\
&= \sum_{\langle \boldsymbol{k},\boldsymbol{h}\rangle} (-1)^{\sum_i \boldsymbol{k}(P_i)} \text{FOMC}(\Phi', \langle \boldsymbol{k}, \boldsymbol{h}\rangle)
\end{aligned}$$

□

3.7 FOMC for Counting Quantifiers

Counting quantifiers are expressions of the form $\exists x^{\geq m}.\Psi(x)$, $\exists^{\leq m}x.\Psi(x)$, and $\exists^{=m}x.\Psi(x)$. The extension of FO^2 with such quantifiers is denoted by C^2 [44]. In this section, we show how FOMC in C^2 can be performed by exploiting the formula for FOMC in FO^2 with cardinality constraints. We assume that the counting quantifier $\exists^{\leq m}y.\Psi(y)$ is expanded to $\bigvee_{k=0}^{m} \exists^{=k}y.\Psi(y)$, and the quantifiers $\exists^{\geq m}y.\Psi(y)$ are first transformed to $\neg(\exists^{\leq m-1}y.\Psi(y))$ and then expanded. We are therefore left with quantifiers of the form $\exists^{=m}y.\Psi(y)$. Hence, any C^2 formula can be transformed into a formula of the form $\Phi_0 \wedge \bigwedge_{k=1}^{q} \forall x.(A_k(x) \leftrightarrow \exists^{=m_k}y.\Psi_k(x,y))$ that preserves FOMC, where[2] Φ_0 is a pure universal formula obtained by replacing every occurrence of the sub-formula $\exists^{=m_k}y.\Psi_k(y)$ with $A_k(x)$, where $A_k(x)$ is a fresh predicate. W.l.o.g, we can assume that $\Psi_k(x,y)$ is the atomic formula $R_k(x,y)$. We will now present a closed-form for FOMC

[2]We assume that Φ_0 contains no existential quantifiers as they can be transformed as described in Theorem 11.

of $\Phi_0 \wedge \bigwedge_k \forall x.(A_k(x) \leftrightarrow \exists^{=m_k} y.R_k(x,y))$. For the sake of notational convenience, we use $\Phi_{i..j}$ to denote $\bigwedge_{i \le s \le j} \Phi_s$ for any set of formulas $\{\Phi_s\}$.

Theorem 12. *Let Φ be the following C^2 formula :*

$$\Phi_0 \wedge \bigwedge_{k=1}^{q} \forall x.(A_k(x) \leftrightarrow \exists^{=m_k} y.R_k(x,y))$$

where Φ_0 is a pure universal formula in FO^2. Let us define the following formulas for each k, where $1 \le k \le q$:

$$\begin{aligned}
\Phi_1^k &= \textstyle\bigwedge_{i=1}^{m_k} \forall x \exists y. A_k(x) \vee B_k(x) \rightarrow f_{ki}(x,y)\\
\Phi_2^k &= \textstyle\bigwedge_{1 \le i < j \le m_k} \forall x \forall y. f_{ki}(x,y) \rightarrow \neg f_{kj}(x,y)\\
\Phi_3^k &= \textstyle\bigwedge_{i=1}^{m_k} \forall x \forall y. f_{ki}(x,y) \rightarrow R_k(x,y)\\
\Phi_4^k &= \forall x. B_k(x) \rightarrow \neg A_k(x)\\
\Phi_5^k &= \forall x \forall y. M_k(x,y) \leftrightarrow ((A_k(x) \vee B_k(x)) \wedge R_k(x,y))\\
\Phi_6^k &= |A_k| + |B_k| = |f_{k1}| = \cdots = |f_{km_k}| = \tfrac{|M_k|}{m_k}
\end{aligned}$$

where[3] *B_k, f_{ki} and M_k are fresh predicates. Then $\mathrm{FOMC}(\Phi, n)$ is given as:*

$$\sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle \models \bigwedge_k \Phi_6^k} \frac{(-1)^{\sum_k \boldsymbol{k}(B_k) + \sum_{k,i} \boldsymbol{k}(P_{ki})} \mathrm{FOMC}(\Phi', \langle \boldsymbol{k}, \boldsymbol{h} \rangle)}{\prod_k m_k!^{\boldsymbol{k}(A_k)}}$$

where Φ' is obtained by replacing each Φ_1^k with $\bigwedge_{i=1}^{m_k} \forall x \forall y. P_{ki}(x) \rightarrow \neg(A_k(x) \vee B_k(x) \rightarrow f_{ki}(x,y))$ in $\Phi_0 \wedge \bigwedge_k \Phi_{1..5}^k$ and P_{ki} are fresh unary predicates.

Lemma 9. *For a given interpretation ω, let A_k^ω and B_k^ω represent the set of constants c such that $\omega \models A_k(c)$ and $\omega \models B_k(c)$, respectively. If $\omega \models \Phi_0 \wedge \bigwedge_{k=1}^{q} \Phi_{1..6}^k$ then every $c \in A_k^\omega \cup B_k^\omega$ has exactly m_k R_k-successors i.e., $\omega \models \exists^{=m_k} y.R(c,y)$.*

Proof. If $c \in A_k^\omega \cup B_k^\omega$, then by Φ_1^k, c has an f_{ki}-successor for every $1 \le i \le m_k$. Φ_2^k implies that c has distinct f_{ki} and f_{kj} successor for any choice of i and j. Φ_3^k implies that any f_{ki}-successor of c is also an R_k-successor. Hence, c has at least m_k R_k-successors.

Axiom Φ_5^k implies that c has exactly as many R_k-successors as M_k-successors. Hence, c has at-least m_k M_k-successors. Furthermore, by Φ_4^k we have that A_k^ω and B_k^ω

[3] If Φ_0 is obtained after a transformation as described in Theorem 11, then we can add the term $\sum_g k(P_g)$ to the exponent of (-1), for the set of unary predicates $\{P_g\}$ introduced to deal with existential quantifiers. Also, any cardinality constraint on predicates of Φ_0 can be easily conjuncted and incorporated into $\wedge_k \Phi_6^k$.

are disjoint. Hence, using Φ_6^k, we can conclude that c has exactly m_k M_k-successors. Finally, using Φ_5^k we can conclude that c has exactly m_k R_k-successors. $\square$

Proof (of Theorem 12). First notice that every model ω of Φ can be extended to $\prod_k m_k!^{A_k^\omega}$ models of $\Phi_0 \wedge \bigwedge_k \Phi_{1..6}^k$ by interpreting B_k in the empty set, f_{ki} in the set of pairs $\langle c, d\rangle$ for $c \in A_k^\omega$ and d being the i-th R_k-successor of c (for some ordering of the R_k-successors) and M_k according to the definition given in Φ_k^5.

Let Ω the set of models of $\Phi_0 \wedge \bigwedge_{k=1}^q \Phi_{1..6}^k$ restricted to the language of Φ, M_k and f_{ki} (i.e., the language of $\Phi_0 \wedge \bigwedge_{k=1}^q \Phi_{1..6}^k$ excluding the predicates B_k) and on a domain C consisting of n elements.

Notice that Ω contains also the models that are not extensions of some model of Φ. Therefore, in the first part of the proof we count the number of extensions of models of Φ in Ω, and successively we will take care of the over-counting due to the multiple interpretations of f_{ki}'s.

Let $\mathcal{S} = \{\Omega_{ck}\}$ be the set of subsets of Ω such that if $\omega \in \Omega_{ck}$ then $\omega \models \neg A_k(c) \wedge \exists^{=m_k} y.R_k(c, y)$. Due to Lemma 9, if $\omega \in \Omega$ then $\omega \models \bigwedge_k \forall x.A_k(x) \rightarrow \exists^{=m_k} y.R_k(x, y)$. Hence, in order to count the models of Φ in Ω we only need to count the number of models in Ω that satisfy $\bigwedge_k \forall x \exists^{=m_k} y.R_k(x, y) \rightarrow A_k(x)$, equivalently, the number of models that belong to none of the Ω_{ck}. Hence, if we are able to evaluate s_l (as introduced in Corollary 3) then we can use Corollary 3 to count the set of models in Ω that satisfy Φ.

Let $\omega \in \Omega$. Let us define Φ_l for $l \geq 0$ as follows:

$$\Phi_l = \Phi_0 \wedge \bigwedge_k \Phi_{1..6}^k \wedge \left(\sum_k |B_k| = l\right) \tag{3.30}$$

Firstly, let $\mathcal{Q}'$ be the set of elements in $\mathcal{S}$ that contain ω. By Lemma 9, ω can be extended in $\binom{|\mathcal{Q}'|}{l}$ models of Φ_l. Each such extension can be achieved by choosing l elements in $\mathcal{Q}'$, and interpreting $B_k(c)$ to be true in the extended model iff Ω_{ck} is a part of the l chosen elements. On the other hand, recall that $s_l = \sum_{|\mathcal{Q}|=l} N(\supseteq \mathcal{Q})$. Every ω that is contained in all the elements of $\mathcal{Q}'$, contributes $\binom{|\mathcal{Q}'|}{l}$ to s_l. Hence, $s_l = \text{FOMC}(\Phi_l, n)$. Using inclusion-exclusion principle (corollary 3), we have that the number of models which do not belong to any of the Ω_{ck} are:

$$\sum_l (-1)^l s_l = \sum_l (-1)^l \text{FOMC}(\Phi_l, n) \tag{3.31}$$

Hence, we have the count of models of Φ in Ω. But notice that this is the count of the models of Φ in the language of $\Phi_0 \wedge \bigwedge_k \Phi^k_{1..6}$ excluding B_k, where there are the additional predicates $\{f_{ki}\}$. Since every interpretation with $|A^\omega_k| = r_k$ can be extended in $m_k!^{r_k}$ models of Φ due to the permutations of $\{f_{ki}\}_{i=1}^{m_k}$, to obtain FOMC on the language of Φ we have to take into account this over-counting[4]. This can be obtained by introducing a cardinality constraint $|A_k| = r_k$ for every A_k and dividing by $m_k!^{r_k}$ for each k and $r_1 ... r_q$ values. Giving the following expression for $\text{FOMC}(\Phi, n)$:

$$\sum_{l, r_k} (-1)^l \frac{\text{FOMC}(\Phi_l \wedge \bigwedge_k |A_k| = r_k, n)}{\prod_k m_k!^{r_k}} \tag{3.32}$$

Also notice that Φ^k_1 contains m_k existential quantifiers, to eliminate them we use the result of Theorem 11. We introduce m_k new unary predicates $P_{k1}, \ldots, P_{km_k}$ for each k, and replace each Φ^k_1 with $\bigwedge_i \forall x \forall y. P_{ki}(x) \rightarrow \neg(A_k(x) \vee B_k(x) \rightarrow f_{ki}(x, y))$. Hence, by Theorem 11 we have that $\text{FOMC}(\Phi, n)$ is equal to:

$$\sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle \models \bigwedge_k \Phi^k_6} \frac{(-1)^{\sum_k \boldsymbol{k}(B_k) + \sum_{k,i} \boldsymbol{k}(P_{ki})} \text{FOMC}(\Phi', \langle \boldsymbol{k}, \boldsymbol{h} \rangle)}{\prod_k m_k!^{\boldsymbol{k}(A_k)}}$$

where Φ' is the pure universal formula $\Phi_0 \wedge \bigwedge_{k=1}^q \Phi^k_{2..5} \wedge \bigwedge_{i,k} P_{ki}(x) \rightarrow \neg(A_k(x) \vee B_k(x) \rightarrow f_{ki}(x, y))$.

□

3.8 Weighted First-Order Model Counting

FOMC formulas introduced so far can be easily extended to weighted model counting by simply defining a positive real-valued weight functions $w(\langle \boldsymbol{k}, \boldsymbol{h} \rangle)$ and adding them as a multiplicative factor to $\text{FOMC}(\Phi, \langle \boldsymbol{k}, \boldsymbol{h} \rangle)$, in all the FOMC formulas. We first deal with symmetric-weight functions (as defined in Definition 16) and then introduce a new larger class of weight functions.

Theorem 13. *Given a* C^2 *sentence* Φ*, symmetric-WFOMC for* Φ *can be obtained from FOMC as follows:*

$$\text{WFOMC}(\Phi, n) = \sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle} w(\langle \boldsymbol{k}, \boldsymbol{h} \rangle) \cdot \text{FOMC}(\Phi, \langle \boldsymbol{k}, \boldsymbol{h} \rangle) \tag{3.33}$$

[4] Notice that M_k leads to no additional models of Φ as interpretations of M_k are uniquely determined by A_k and R_k by Φ^k_5.

where $w(\langle \boldsymbol{k}, \boldsymbol{h} \rangle)$ is defined as follows:

$$w(\langle \boldsymbol{k}, \boldsymbol{h} \rangle) = \prod_{P \in \mathcal{L}} w(P)^{\langle \boldsymbol{k}, \boldsymbol{h} \rangle (P)} \cdot \bar{w}(P)^{\langle \boldsymbol{k}, \boldsymbol{h} \rangle (\neg P)}$$

where $w(P)$ and $\bar{w}(P)$ are real valued weights on predicate P, and it's negation respectively.

Proof. The proof is a consequence of the observation that $\text{FOMC}(\Phi, \langle \boldsymbol{k}, \boldsymbol{h} \rangle)$ is the number of models of Φ that contain $\boldsymbol{k}(P)$ elements that satisfy P if P is unary, and $\langle \boldsymbol{k}, \boldsymbol{h} \rangle (P)$ pairs of elements that satisfy P, if P is binary. □

3.8.1 Expressing Count Distribution

Kuželka [37] introduced a strictly more expressive class of weight functions which also preserves domain liftability. These weight functions can express count distributions, which are defined as follows:

Definition 29 (Count distribution [37])**.** *Let $\Phi = \{\alpha_i, w_i\}_{i=1}^m$ be a Markov Logic Network defining a probability distribution $p_{\Phi,\Omega}$ over a set of possible worlds (we call them assignments) of a formula Ω. The count distribution of Φ is the distribution over m-dimensional vectors of non-negative integers $\boldsymbol{n}$ given by*

$$q_\Phi(\Omega, \boldsymbol{n}) = \sum_{\omega \models \Omega,\ \boldsymbol{n} = \boldsymbol{N}(\Phi, \omega)} p_{\Phi,\Omega}(\omega) \tag{3.34}$$

where $\boldsymbol{N}(\Phi, \omega) = (n_1, \ldots, n_m)$ and n_i is the number of grounding of α_i that are true in ω.

Kuželka [37] showed that count distributions can be modelled by Markov Logic Networks with complex weights. In the following, we prove that if each α_i is in FO^2, count distributions can be expressed by a $w(\langle \boldsymbol{k}, \boldsymbol{h} \rangle)$.

Theorem 14. *Every count distribution over a set of possible worlds of a formula Ω definable in FO^2 can be modelled with a weight function on $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$, by introducing m new predicates P_i and adding the axioms $P_i(x) \leftrightarrow \alpha_i(x)$ and $P_j(x, y) \leftrightarrow \alpha_j(x, y)$, if α_i and α_j has one and two free variables respectively and by defining:*

$$q_\Phi(\Omega, \boldsymbol{n}) = \frac{1}{Z} \sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle (P_i) = n_i} w(\langle \boldsymbol{k}, \boldsymbol{h} \rangle) \cdot \text{FOMC}(\Omega, \langle \boldsymbol{k}, \boldsymbol{h} \rangle) \tag{3.35}$$

where $Z = \text{WFOMC}(\Omega, w, n)$ is the partition function.

Proof. The proof is a simple consequence of the fact that all the models agreeing with a count statistic $\boldsymbol{n}$ can be counted using cardinality constraints which agree with $\boldsymbol{n}$. Any such cardinality constraint correspond to a specific set of $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$ vectors. Hence, we can express arbitrary probability distributions over count statistics by picking real valued weights for $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$ vector.

Since Ω is a FO^2 formula, then we can compute FOMC as follows:

$$\textsc{fomc}(\Omega, n) = \sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle} \textsc{fomc}(\Omega, \langle \boldsymbol{k}, \boldsymbol{h} \rangle)$$

Let us define $w\langle \boldsymbol{k}, \boldsymbol{h} \rangle$ for each $\langle \boldsymbol{k}, \boldsymbol{h} \rangle$ as follows:

$$w\langle \boldsymbol{k}, \boldsymbol{h} \rangle = \frac{1}{\textsc{fomc}(\Omega, \langle \boldsymbol{k}, \boldsymbol{h} \rangle)} \sum_{\substack{\omega \models \Omega \\ N(\alpha_1, \omega)_1 = \langle \boldsymbol{k}, \boldsymbol{h} \rangle (P_1) \\ \cdots \\ N(\alpha_m, \omega)_m = \langle \boldsymbol{k}, \boldsymbol{h} \rangle (P_m)}} p_{\Phi, \Omega}(\omega) \tag{3.36}$$

This definition implies that the partition function Z is equal to 1. Indeed:

$$\begin{aligned}
Z &= \textsc{wfomc}(\Omega, w, n) \\
&= \sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle} w(\langle \boldsymbol{k}, \boldsymbol{h} \rangle) \cdot \textsc{fomc}(\Omega, \langle \boldsymbol{k}, \boldsymbol{h} \rangle) \\
&= \sum_{\langle \boldsymbol{k}, \boldsymbol{h} \rangle} \sum_{\substack{\omega \models \Omega \\ N(\alpha_1, \omega)_1 = \langle \boldsymbol{k}, \boldsymbol{h} \rangle (P_1) \\ \cdots \\ N(\alpha_m, \omega)_m = \langle \boldsymbol{k}, \boldsymbol{h} \rangle (P_m)}} p_{\Phi, \Omega}(\omega) \\
&= \sum_{\omega \models \Omega} \sum_{\substack{\langle \boldsymbol{k}, \boldsymbol{h} \rangle \\ N(\alpha_1, \omega)_1 = \langle \boldsymbol{k}, \boldsymbol{h} \rangle (P_1) \\ \cdots \\ N(\alpha_m, \omega)_m = \langle \boldsymbol{k}, \boldsymbol{h} \rangle (P_m)}} p_{\Phi, \Omega}(\omega) \\
&= \sum_{\omega \models \Omega} p_{\Phi, \Omega}(\omega) \\
&= 1
\end{aligned}$$

Hence,

$$\begin{aligned}
q_\Phi(\Omega, \boldsymbol{n}) &= \sum_{\langle \boldsymbol{k},\boldsymbol{h}\rangle(P_i)=n_i} \text{FOMC}(\Omega, \langle \boldsymbol{k}, \boldsymbol{h}\rangle) \cdot w(\langle \boldsymbol{k}, \boldsymbol{h}\rangle) \\
&= \sum_{\langle \boldsymbol{k},\boldsymbol{h}\rangle(P_i)=n_i} \sum_{\substack{\omega\models\Omega \\ N(\alpha_1,\omega)_1=\langle \boldsymbol{k},\boldsymbol{h}\rangle(P_1) \\ \cdots \\ N(\alpha_m,\omega)_m=\langle \boldsymbol{k},\boldsymbol{h}\rangle(P_m)}} p_{\Phi,\Omega}(\omega) \\
&= \sum_{\substack{\omega\models\Omega \\ N(\alpha_1,\omega)_1=n_1 \\ \cdots \\ N(\alpha_m,\omega)_m=n_m}} p_{\Phi,\Omega}(\omega)
\end{aligned}$$

□

Example 14. *In the example proposed in [37], they model the distribution of a sequence of 4 coin tosses such that the probability of getting an odd number of heads is zero, whereas each event with even number of heads is equally likely. In order to model this distribution, we introduce a predicate* $H(x)$ *over a domain of* 4 *elements, we also define* Ω *as* $\top$. *This means that every model of this theory is a model of* Ω. *Notice that this distribution cannot be expressed using symmetric weights, as symmetric weights can only express binomial distribution for this language. But we can define weight function on* $\langle \boldsymbol{k}, \boldsymbol{h}\rangle$ *vector. In this case* $\boldsymbol{k} = (k_0, k_1)$ *such that* $k_0 + k_1 = 4$. *Since there are no binary predicates we can ignore* $\boldsymbol{h}$. *Intuitively,* k_0 *is the number of tosses which are not heads and* k_1 *is the number of tosses which are heads. If we define the weight function as* $w(k_0, k_1) = 1 + (-1)^{k_1}$. *Then by applying* (3.35) *we obtain the following probability distribution over the tosses:*

$$\begin{aligned}
q(\Omega, (4,0)) &= \frac{\binom{4}{4} \cdot (1+1)}{16} = \frac{1}{8} \\
q(\Omega, (3,1)) &= \frac{\binom{4}{3} \cdot (1-1)}{16} = 0 \\
q(\Omega, (2,2)) &= \frac{\binom{4}{2} \cdot (1+1)}{16} = \frac{3}{4} \\
q(\Omega, (1,3)) &= \frac{\binom{4}{1} \cdot (1-1)}{16} = 0 \\
q(\Omega, (0,4)) &= \frac{\binom{4}{0} \cdot (1+1)}{16} = \frac{1}{8}
\end{aligned}$$

which coincides with the distribution obtained by [37]. Notice, that such a distribution cannot be expressed through symmetric weight functions and obligates the use of a strictly more expressive class of weight functions.

We are able to capture count distributions without losing domain liftability. Furthermore, we do not introduce complex or even negative weights, making the relation between weight functions and probability rather intuitive.

3.9 Conclusion

In this chapter, we have presented a closed-form formula for FOMC of universally quantified formulas in FO^2 that can be computed in polynomial time w.r.t. domain cardinality. From this, we are able to derive a closed-form expression for FOMC in FO^2 formulas in Scott's Normal Form, extended with cardinality constraints and counting quantifiers. These extended formulas are also computable in polynomial time, and therefore they constitute lifted inference algorithms for C^2. All the formulas are extended to cope with weighted model counting in a simple way, admitting a larger class of weight functions than symmetric weight functions. All the results have been obtained using combinatorial principles, providing a uniform treatment to all these fragments.

Chapter 4

WFOMC with Acyclicity Constraints

The work presented in this chapter forms part of the following under review article, available at arxiv:

Sagar Malhotra and Luciano Serafini. Weighted First Order Model Counting with Directed Acyclic Graph Axioms. arXiv:2302.09830 [24]

4.1 Introduction

A large part of SRL is concerned with modelling, learning and inferring over large scale datasets. However, as we show in Section 2.2, modelling complex relationships is restricted by the complexity of WFOMC in models like MLNs. As demonstrated in the previous chapter this complexity can be overcome for the two-variable fragment of FOL extended with cardinality constraints and counting quantifiers. However, real-world data modelling requires much more expressivity. Furthermore, modelling requirements may not be even FOL-definable. One such example is *Directed Acyclic Graphs* (DAGs). DAGs are ubiquitous data structures, that appear in all kinds of applications. Citation networks, such as CiteSeer, Cora and PubMed, can be modeled as DAGs. Citation networks are acyclic because a paper cannot cite itself or cite a paper that cites it. In these networks, articles are represented as nodes, and the edges represent the citation relationships between them. A set of multiple genealogy trees that trace family relationships can also be represented as DAGs (with additional constraints i.e. being a forest), where nodes represent individuals and edges represent parent-child relationships.

Hence, SRL models that can express an Acyclicity constraints can significantly aid learning and inference tasks in many real-world datasets.

In this chapter, we show that WFOMC in C^2 expanded with a DAG axiom is domain liftable, allowing us to efficiently answer questions like:

"How many DAGs with exactly (or atleast or atmost) k sources and exactly m (or atleast or atmost) sinks exist ?"

Given the vastness of DAG applications, these results can allow modelling of many real-world scenarios in models like Markov Logic Networks e.g. citation networks [45]. Furthermore, since counting DAGs have become an important tool for learning Bayesian Networks and causal inference [46, 47], our results could potentially be exploited in these domains as well.

4.2 Related Works

Recent results have attempted to expand the domain-liftability of WFOMC in C^2 [19, 22] by expanding the logical language in different directions, such as the *linear order axiom* [48]. However, one of the most interesting such development has been extension of domain-liftability of C^2 with a *tree axiom* [49]. This development is interesting as a tree structure is not only outside the expressivity of domain-liftable fragments, but is inexpressible in any FOL language. This chapter expands the domain-liftability of C^2, with another constraint in-expressible in FOL, namely the *acyclicity constraint.*

4.3 Background

We assume background from Section 2.1, Section 2.2 and Section 2.7. We will revisit WFOMC in FO^2 and C^2 as presented in [28] and [19], respectively, as these approaches render the proofs in this chapter easier to formulate.

4.3.1 WFOMC

In WFOMC as defined in equation (2.2), we assume that the weight function w does not depend on individual domain constants, which implies that w assigns same weight to two interpretations which are isomorphic under the permutation of domain elements. Hence, for a domain Δ of size n, we can equivalently use $[n]$ as our domain. Furthermore, in this chapter we will focus only on symmetric weight functions as defined in Definition 16. We will also need to invoke modularity of WFOMC-preserving reductions.

Definition 30 ([36]). *A reduction* $(\Phi, w, \bar{w})$ *to* $(\Phi', w', \bar{w}')$ *is modular iff for any sentence* Λ*:*

$$\textsc{wfomc}(\Phi \wedge \Lambda, (w, \bar{w}), n) = \textsc{wfomc}(\Phi' \wedge \Lambda, (w', \bar{w}'), n)$$

Intuitively, modularity implies that the reduction procedure is sound under presence of other sentences Λ. And that any new sentence Λ does not invalidate the reduction.

For the rest of the chapter, whenever referring to weights, we intend symmetric weights. Hence, we will use $\textsc{wfomc}(\Phi, n)$ without explicitly mentioning the weights w and $\bar{w}$.

Revisiting WFOMC in FO^2

We now define some weight parameters associated with an FOL language. These parameters will be useful for treating WFOMC. Given an FO^2 language $\mathcal{L}$ and symmetric weight functions $(w, \bar{w})$, let $\mathcal{I}$ denotes the set of atoms in $\mathcal{L}$ containing only variables, i.e. not grounded. We then define the following two parameters for each 1-type $i(x)$ and 2-table $l(x, y)$:

$$w_i = \prod_{\substack{i(x) \models g \\ g \in \mathcal{I}}} w(pred(g)) \prod_{\substack{i(x) \models \neg g \\ g \in \mathcal{I}}} \bar{w}(pred(g))$$

and

$$v_l = \prod_{\substack{l(x,y) \models g \\ g \in \mathcal{I}}} w(pred(g)) \prod_{\substack{l(x,y) \models \neg g \\ g \in \mathcal{I}}} \bar{w}(pred(g)) \tag{4.1}$$

We will now present a slightly reformulated version of WFOMC for universally quantified FO^2 formulas, as presented in [28].

Theorem 15 (Beame et al. (2015) [28]). *Given a universally quantified FO*^{2} *formula* $\forall xy.\Phi(x, y)$*, interpreted over a domain* $[n]$*, then the weighted model count of the models* ω *such that* $\omega \models \forall xy.\Phi(x, y)$ *and* ω *has the 1-type cardinality* $\boldsymbol{k}$ *is given as:*

$$\textsc{wfomc}(\forall xy.\Phi(x, y), \boldsymbol{k}) = \binom{n}{\boldsymbol{k}} \prod_{i \in [u]} w_i^{k_i} \prod_{i \leq j \in [u]} r_{ij}^{\boldsymbol{k}(i,j)} \tag{4.2}$$

where $\boldsymbol{k}(i,j)$ *is defined as follows:*

$$\boldsymbol{k}(i,j) = \begin{cases} \frac{k_i(k_i-1)}{2} & \textit{if } i = j \\ k_i k_j & \textit{otherwise} \end{cases}$$

where we define $r_{ij} = \sum_{l \in [b]} n_{ijl} v_l$, *where* n_{ijl} *is* 1 *if* $ijl(x,y) \models \Phi(\{x,y\})$ *and* 0 *otherwise.*

Proof. In a given 1-type cardinality vector $\boldsymbol{k}$, k_i represents the number of constants c of 1-type i. Also a given constant realizes exactly one 1-type. Hence, for a given $\boldsymbol{k}$, we have $\binom{n}{\boldsymbol{k}}$ possible ways of realizing 1-types. A domain constant c realizing the i^{th} 1-type contributes a weight of w_i multiplicatively to weight of ω. Hence, for a given $\boldsymbol{k}$, the contribution due to 1-type realizations is $\prod_{i \in [u]} w_i^{k_i}$. Furthermore, in an interpretation $\omega \models \forall xy.\Phi(x,y)$, given a pair of constants c and d such that c is of 1-type i and d is of 1-type j, using Lemma 3, we have that (c,d) can realize the 2-table $l(c,d)$, only if $ijl(x,y) \models \Phi(x,y)$. Hence, in an arbitrary interpretation, the multiplicative weight contribution due to $l(x,y)$ realization is given by $n_{ijl}v_l$. Also using Lemma 2, each ordered pair of constants can realize exactly one and only one 2-table. Hence, the possible 2-table realization contributes a weight $r_{ij} = \sum_l n_{ijl}v_l$. Furthermore, given 1-type assignments to $i(c)$ and $j(d)$, the ordered pair (c,d) can realize 2-table independently of all other domain constants. Finally, There are $\boldsymbol{k}(i,j)$ possible such pairs, contributing a weight

$$\prod_{i \leq j \in [u]} r_{ij}^{\boldsymbol{k}(i,j)}$$

□

Clearly, equation (4.2) can be computed in polynomial time w.r.t domain cardinality. Furthermore, there are only polynomially many $\boldsymbol{k}$, in the size of the domain. Hence, WFOMC$(\forall xy.\Phi(x,y), n)$ given as:

$$\sum_{|\boldsymbol{k}|=n} \text{WFOMC}(\forall xy.\Phi(x,y), \boldsymbol{k})$$

can be computed in polynomial time w.r.t domain size n.

[36] show that any FOL formula with existential quantification can be modularly reduced to a WFOMC preserving universally quantified FO^2 formula, with additional new predicates and negative weights. Hence, showing that FO^2 is domain-liftable.

Revisiting WFOMC in C^2

[19] show that WFOMC in C^2 can be reduced to WFOMC in FO^2 with cardinality constraints. Furthermore, these reductions are independent of the domain cardinality. In order to prove domain liftability, the proof in [19], relies on Lagrange interpolation. In the following we provide an easier presentation of the proof.

Theorem 16 ([19], slightly reformulated)**.** *Let Φ be a first-order logic sentence. Let Γ be a arbitrary cardinality constraint. Then* $\textsc{wfomc}(\Phi \wedge \Gamma, \boldsymbol{k})$ *can be computed in polynomial time with respect to the domain cardinality, relative to the* $\textsc{wfomc}(\Phi, \boldsymbol{k})$ *oracle.*

Proof. We assume an FOL language $\mathcal{L}$, with r relational symbols $\{R_i/a_i\}_{i\in[r]}$. Given an interpretation ω, let $\boldsymbol{\mu} = \langle |R_1| \dots |R_r| \rangle$ be the vector containing cardinality of each predicate R_i in ω. Now, $\mathrm{W}(\omega)$ can be easily evaluated using the definition of symmetric weight functions (Definition 16). Furthermore, any two interpretations with same predicate cardinalities $\boldsymbol{\mu}$ as ω, have the same weight $\mathrm{W}(\omega)$. Hence, we use $\mathrm{W}_{\boldsymbol{\mu}}$ to denote the weight $\mathrm{W}(\omega)$.

Given an FOL formula Φ, let $A_{\boldsymbol{\mu}}$ be the number of models $\omega \models \Phi \wedge \boldsymbol{\mu}$. Clearly, the following holds:

$$\textsc{wfomc}(\Phi, \boldsymbol{k}) = \sum_{\boldsymbol{\mu}} A_{\mu} \mathrm{W}_{\boldsymbol{\mu}} \tag{4.3}$$

Each predicate R_i/a_i can be grounded to n^{a_i} ground atoms. Hence, there are $n^{\sum_{i\in[r]} a_i}$ possible values of $\boldsymbol{\mu}$, which are polynomial in n. Hence, if we evaluate $\textsc{wfomc}(\Phi, \boldsymbol{k})$ for $n^{\sum_{i\in[r]} a_i}$ different values of weight functions $(w, \bar{w})$, then we have a linear system of $n^{\sum_{i\in[r]} a_i}$ equations with $n^{\sum_{i\in[r]} a_i}$ variables (variables being A_{μ}), which can be solved in $O(n^{3\sum_{i\in[r]} a_i})$ time, using Gauss-elimination. Once we have all the $A_{\boldsymbol{\mu}}$, we can evaluate any cardinality constraint as follows:

$$\textsc{wfomc}(\Phi \wedge \Gamma, \boldsymbol{k}) = \sum_{\boldsymbol{\mu} \models \boldsymbol{\Gamma}} A_{\mu} \mathrm{W}_{\boldsymbol{\mu}} \tag{4.4}$$

where $\boldsymbol{\mu} \models \boldsymbol{\Gamma}$ represents the fact that the predicate cardinalities $\boldsymbol{\mu}$, satisfy the cardinality constraint Γ. Since, there area only polynomially many $\boldsymbol{\mu}$, equation (4.4) can be computed in polynomial time. ☐

Remark 5. *In equation* (4.4)*, we assume that $\boldsymbol{\mu} \models \Gamma$ can be checked in polynomial time wrt n. Which is a reasonable assumption for all our purposes.*

Remark 6. *In the proof presented above (and in [19]), the first-order definability of Φ is never invoked. This property has also been exploited for imposing cardinality constraints with tree axiom in [49].*

Theorem 16 extends domain-liftability of any sentence Φ to its domain liftability with cardinality constraints. We now move onto the results on domain-liftability of C^2.

Theorem 17 (Kuželka (2021)[19]). *The fragment of first-order logic with two variables and counting quantifiers is domain-liftable.*

The key idea behind Theorem 17 is that WFOMC of a C^2 sentence Φ can be converted to a problem of WFOMC of an FO^2 sentence Φ' with cardinality constraints Γ on an extended vocabulary with additional weights for the new predicates (the new predicates are weighted 1 or -1). We refer the reader to [19], for the detailed treatment of Theorem 17. However, for our purposes it is important to note that this transformation is modular. The modularity of the WFOMC procedure as presented in [19], has also been exploited to demonstrate domain-liftability of C^2 extended with Tree axiom [49] and Linear Order axiom [48].

4.3.2 Counting Directed Acyclic Graphs

A *Directed Acyclic Graph (DAG)* is a directed graph such that starting from an arbitrary node i and traversing an arbitrary path along directed edges, we would never arrive at node i. We now present the derivation of a recursive formula for counting the number of DAGs.

Let the nodes be the set $[n]$ and let A_i be the set of DAGs on $[n]$ where node i has indegree zero. Since every DAG has at least one node with in-degree zero, we have that the total number of DAGs i.e. a_n is given as $|\bigcup_{i\in[n]} A_i|$. The number of DAGs such that all nodes in $J \subseteq [n]$ have in-degree zero is then given as $A_J := \bigcap_{j\in J} A_j$. Let us assume that $J = [m]$ for some $1 \leq m \leq n$. We now derive a method for computing $A_{[m]}$. We make the following three observations for deriving the formula for counting the DAGs in $A_{[m]}$.

- Observation 1. If $\omega \in A_{[m]}$, then there are no edges between the nodes in $[m]$, as otherwise a node in $[m]$ will have a non-zero in-degree. In other words, only directed edges from $[m]$ to $[\bar{m}]$ are allowed.

- Observation 2. If $\omega \in A_{[m]}$, then subgraph of ω restricted to $[\bar{m}]$ i.e. $\omega \downarrow [\bar{m}]$ is a DAG. And the subgraph of ω restricted to $[m]$ is just an empty graph, i.e., the set of isolated nodes $[m]$ with no edges between them.

- Observation 3. Given a DAG on $[\bar{m}]$, then it can be extended to $2^{m(n-m)}$ DAGs in $A_{[m]}$. This is because DAGs in $A_{[m]}$ have no edges between the nodes in $[m]$. They only have outgoing edges from $[m]$ to $[\bar{m}]$. For extending a given DAG on $[\bar{m}]$ to a DAG in $A_{[m]}$, we can either draw an out-going edge from $[m]$ to $[\bar{m}]$ or not. Giving us two choices for each pair of nodes in $[m] \times [\bar{m}]$. Hence, there are $2^{|[m]\times[\bar{m}]|} = 2^{m(n-m)}$ ways to extend a given DAG on $[\bar{m}]$ to a DAG in $A_{[m]}$.

The number of possible DAGs on $[\bar{m}]$ is a_{n-m}. Due to Observation 3, we have that $A_{[m]}$ has $2^{m(n-m)}a_{n-m}$ DAGs obtained by extending the DAGs on $[m]$. Furthermore, due to Observation 1 and Observation 2, these are all the possible DAGs in $A_{[m]}$. Hence, $|A_{[m]}|= 2^{m(n-m)}a_{n-m}$. Now, we can repeat this argument for any m sized subset of $[n]$. Hence, if $|J|= |J'|= m$ then $A_J = A_{J'} = 2^{m(n-m)}a_{n-m}$. Hence, using the principle of inclusion-exclusion as given in equation (2.20), we have that:

$$a_n = \sum_{m=1}^{n} (-1)^{m+1} \binom{n}{m} 2^{m(n-m)} a_{n-m} \tag{4.5}$$

Notice that replacing $n - m$ with l in equation (4.5), it can be equivalently written as:

$$a_n = \sum_{l=0}^{n-1} (-1)^{n-l+1} \binom{n}{l} 2^{l(n-l)} a_l \tag{4.6}$$

This change of variable allows us to write a bottom-up algorithm for counting DAGs, as given in Algorithm 1. Based on this algorithm we now show that counting DAGs can be performed in polynomial time with respect to the number of nodes n.

Proposition 1. *The number of labelled DAGs over n nodes can be computed in polynomial time.*

Proof. We define $a_0 = 1$ by convention and then by using equation (4.6) in Algorithm 1, we incrementally compute $a_1, a_2...$, saving each result in a list given by A. The for loop runs in time $O(n)$, and in each run in line 5, we perform other $O(n)$ operations. Hence, the algorithm runs in $O(n^2)$. □

4.4 WFOMC with DAG Axiom

In this section we extend the approach used for counting DAGs in equation (4.5) to WFOMC of FO^2 and C^2 formulas with a DAG Axiom. First, we formally define the

Algorithm 1 Number of DAG on n nodes

1: **Input**: n
2: **Output**: a_n
3: $A[0] \leftarrow 1$
4: **for** $i = 1$ to n **do**
5: $\quad A[i] \leftarrow \sum_{l=0}^{i-1}(-1)^{i-l+1}\binom{i}{l}2^{l(i-l)}A[l]$
6: **end for**
7: **return** $A[n]$

DAG axiom. We then present Proposition 2, Proposition 3 and Proposition 4, analogous to Observation 1, 2 and 3 respectively, as presented in the subsection 4.3.2. We then use principle of inclusion-exclusion to compute the WFOMC of universally quantified FO^2 formulas extended with a DAG axiom. And show our method to be domain-liftable. The proposed apporach is then extended to admit full FO^2, Cardinality constraints and C^2. We finally extend the DAG axiom, with additional unary predicates that represent sources and sinks of the DAG.

Definition 31. *Let Φ be a first-order logic sentence, possibly containing the binary relation R. An interpretation ω is a model of $\Psi = \Phi \wedge Acyclic(R)$ if and only if:*

- *ω is a model of Φ, and*
- *ω_R forms a Directed Acyclic Graph*

Definition 32. *Let $\Psi = \Phi \wedge Acyclic(R)$, where Φ is a first-order logic sentence, be interpreted over the domain $[n]$. Let $1 \leq m \leq n$. Then ω is a model of $\Psi_{[m]}$ if and only if ω is a model of Ψ on $[n]$ and the domain elements in $[m]$ have zero R-indegree.*

Notice that due to Definition 32, for the domain $[n]$, $\Psi_{[n]}$ is equivalent to $\Psi' = \Phi \wedge \neg R(x,y)$.

Proposition 2. *Let $\Psi = \forall xy.\Phi(x,y) \wedge Acyclic(R)$ and $\Psi' = \forall xy.\Phi(x,y) \wedge \neg R(x,y)$, where $\Phi(x,y)$ is quantifier-free, be interpreted over $[n]$. Let $1 \leq m \leq n$. If ω is a model of $\Psi_{[m]}$, then $\omega \downarrow [m] \models \Psi'$ and $\omega \downarrow [\bar{m}] \models \Psi$.*

Proof. We have that $\omega \models \Psi$. Hence, we have that:

$$\begin{aligned} \omega &\models \bigwedge_{(c,d)\in[n]^2} \Phi(c,d) \\ \Rightarrow \omega &\models \bigwedge_{(c,d)\in[m]^2} \Phi(c,d) \bigwedge_{(c,d)\in[\bar{m}]^2} \Phi(c,d) \\ &\bigwedge_{(c,d)\in[\bar{m}]\times[m]} \Phi(c,d) \bigwedge_{(c,d)\in[m]\times[\bar{m}]} \Phi(c,d) \end{aligned}$$

Since, $\omega \models \bigwedge_{(c,d)\in[m]^2} \Phi(c,d)$ and $\omega \models \bigwedge_{(c,d)\in[\bar{m}]^2} \Phi(c,d)$, we have that $\omega \downarrow [m] \models \forall xy.\Phi(x,y)$ and $\omega \downarrow [\bar{m}] \models \forall xy.\Phi(x,y)$. Now, since $[m]$ has zero R-indegree, it can only have outgoing R-edges to $[\bar{m}]$. Hence, we can infer that $\omega \downarrow [m] \models \forall xy.\neg R(x,y)$. Now, ω_R is a DAG, then so is $\omega_R \downarrow [\bar{m}]$. Hence, $\omega \downarrow [\bar{m}] \models \mathit{Acyclic}(R)$. Hence, $\omega \downarrow [m] \models \Psi'$ and $\omega \downarrow [\bar{m}] \models \Psi$. □

Proposition 3. *Let $\Psi = \forall xy.\Phi(x,y) \wedge \mathit{Acyclic}(R)$ and $\Psi' = \forall xy.\Phi(x,y) \wedge \neg R(x,y)$, where $\Phi(x,y)$ is quantifier-free, be interpreted over the domain $[n]$. Let ω' be a model of Ψ' on the domain $[m]$ and let ω'' be a model of Ψ on the domain $[\bar{m}]$. Then the number of extensions ω, of $\omega' \uplus \omega''$, such that $\omega \models \Psi_{[m]} \wedge \boldsymbol{k}$ is given as:*

$$\prod_{i,j\in[u]} n_{ij}^{k'_i \cdot k''_j} \tag{4.7}$$

where k'_i and k''_i are the number of domain constants realizing the i^{th} 1-type in ω' and ω'' respectively. We define n_{ijl} to be 1 if $ijl(x,y) \models \Phi(\{x,y\}) \wedge \neg R(y,x)$ and 0 otherwise and $n_{ij} = \sum_{l\in[b]} n_{ijl}$.

Proof. In order to obtain an interpretation $\omega \models \Psi_{[m]} \wedge \boldsymbol{k}$ on the domain $[n]$ from $\omega' \uplus \omega''$, we only need to extend $\omega' \uplus \omega''$ with interpretations of the ground-atoms containing $(c,d) \in [m] \times [\bar{m}]$. For a given pair $(c,d) \in [m] \times [\bar{m}]$, let $\omega' \models i(c)$ and $\omega'' \models j(d)$. Since ω is a model of $\forall xy.\Phi(x,y)$, we must have that $ijl(c,d) \models \Phi(\{c,d\})$. Furthermore, since we want that every domain element in $[m]$ has indegree zero, we cannot have $R(d,c)$. Hence, we must have that $ijl(c,d) \models \Phi(\{c,d\}) \wedge \neg R(d,c)$. Hence, the number of 2-tables that can be realized by (c,d) is given by n_{ij}. Since there are k'_i domain elements c realizing the i^{th} 1-type in ω' and k''_j domain elements d realizing the j^{th} 1-type in ω'', the number of extensions ω, of $\omega' \uplus \omega''$, such that $\omega \models \Psi_{[m]} \wedge \boldsymbol{k}$ is given by expression (4.7). □

Proposition 4. *Let $\Psi = \forall xy.\Phi(x,y) \wedge Acyclic(R)$ and $\Psi' = \forall xy.\Phi(x,y) \wedge \neg R(x,y)$, where $\Phi(x,y)$ is quantifier-free. Then:*

$$\textsc{wfomc}(\Psi_{[m]}, \boldsymbol{k}) = \sum_{\substack{\boldsymbol{k}=\boldsymbol{k}'+\boldsymbol{k}'' \\ |\boldsymbol{k}'|=m}} \prod_{i,j\in[u]} r_{ij}^{k'_i k''_j} \cdot \textsc{wfomc}(\Psi', \boldsymbol{k}') \cdot \textsc{wfomc}(\Psi, \boldsymbol{k}'') \tag{4.8}$$

where $\boldsymbol{k}' + \boldsymbol{k}''$ represents the element-wise sum of integer-vectors $\boldsymbol{k}'$ and $\boldsymbol{k}''$, such that $|\boldsymbol{k}'| = m$ and $|\boldsymbol{k}''| = |\boldsymbol{k}| - m$. Also, $n_{ij} = \sum_l n_{ijl} v_l$, where n_{ijl} is 1 if $ijl(x,y) \models \Phi(\{x,y\}) \wedge \neg R(y,x)$ and 0 otherwise.

Proof. The WFOMC of Ψ' on $[m]$, with 1-type cardinality vector $\boldsymbol{k}'$ is given as $\textsc{wfomc}(\Psi', \boldsymbol{k}')$. Similarly, the WFOMC of Ψ on $[\bar{m}]$, with 1-type cardinality vector $\boldsymbol{k}''$ is given as $\textsc{wfomc}(\Psi, \boldsymbol{k}'')$. Due to proposition 3, each pair of models counted in $\textsc{wfomc}(\Psi', \boldsymbol{k}')$ and $\textsc{wfomc}(\Psi, \boldsymbol{k}'')$, can be extended in $\prod_{i,j\in[u]} n_{ij}^{k'_i k''_j}$ ways to a model of $\Psi_{[m]} \wedge \boldsymbol{k}$. It is easy to see that the total multiplicative weight contribution of these extensions is given as $\prod_{i,j\in[u]} r_{ij}^{k'_i k''_j}$. The summation in (5.7) runs over all possible realizable 1-type cardinalities over $[m]$ and $[\bar{m}]$, represented by $\boldsymbol{k}'$ and $\boldsymbol{k}''$ respectively, such that they are consistent with $\boldsymbol{k}$, i.e. when $\boldsymbol{k} = \boldsymbol{k}' + \boldsymbol{k}''$. Hence, formula (5.7) gives us the WFOMC of the models ω, such that $\omega \downarrow [m] \models \Psi'$, $\omega \downarrow [\bar{m}] \models \Psi$ and $\omega \models \Psi \wedge \boldsymbol{k}$ where the domain constant in $[m]$ have zero R indegree. Due to proposition 2, we have that these are all the models such that $\omega \models \Psi \wedge \boldsymbol{k}$ and the domain constants in $[m]$ have zero R indegree. $\square$

Proposition 5. *The first order model count of the formula $\Psi = \forall xy.\Phi(x,y) \wedge Acyclic(R)$, where $\Phi(x,y)$ is quantifier-free, is given as:*

$$\textsc{wfomc}(\Psi, \boldsymbol{k}) = \sum_{m=1}^{|\boldsymbol{k}|} (-1)^{m+1} \binom{|\boldsymbol{k}|}{m} \textsc{wfomc}(\Psi_{[m]}, \boldsymbol{k}) \tag{4.9}$$

Proof. The proof idea is very similar to the case for counting DAGs as given in (4.5). Let the domain be $[n]$, hence $|\boldsymbol{k}| = n$. Let A_i be the set of models $\omega \models \Psi$, such that ω has 1-type cardinality $\boldsymbol{k}$ and the domain element i has zero R-indegree. Since, each DAG has atleast one node with zero R-indegree, our goal is to compute $\mathrm{W}(\cup_{i\in[n]} A_i)$. Let $J \subseteq [n]$ be an arbitrary set of domain constants. Let $A_J = \bigcap_{j\in J} A_j$ for an arbitrary subset J of $[n]$. Then using principle of inclusion-exclusion as given in equation (2.22), we have that:

$$\text{WFOMC}(\Psi, \boldsymbol{k}) = \sum_{\emptyset \neq J \subseteq [n]} (-1)^{|J|+1} \mathrm{W}(A_{[m]}) \tag{4.10}$$

Now, $A_{[m]}$ is the set of models such that domain elements in $[m]$ have zero R-indegree. Hence, $A_{[m]}$ are exactly the models of $\Psi_{[m]}$. Furthermore, notice that in Proposition 2, Proposition 3 and Proposition 4, $[m]$ can be replaced with any m-sized subset J of $[n]$. Hence, for all $J \subseteq [n]$, such that $|J|= m$, we have that $\mathrm{W}(A_J) = \text{WFOMC}(\Psi_{[m]}, \boldsymbol{k})$. Hence, equation (4.10) reduces to equation (4.9). □

We make a change of variable in equation (4.9) (similar to equation (4.6)), by replacing m with $|\boldsymbol{k}|-l$, to obtain the following equation:

$$\text{WFOMC}(\Psi, \boldsymbol{k}) = \sum_{l=0}^{|\boldsymbol{k}|-1} (-1)^{|\boldsymbol{k}|-l+1} \binom{|\boldsymbol{k}|}{l} \text{WFOMC}(\Psi_{[|\boldsymbol{k}|-l]}, \boldsymbol{k}) \tag{4.11}$$

We provide pseudocode for evaluating equation (4.11) in Algorithm 2, namely WFOMC-DAG. We now analyse how WFOMC-DAG works and show that it runs in polynomial time with respect to domain cardinality $|\boldsymbol{k}|= n$.

WFOMC-DAG takes as input $\Psi = \forall xy.\Phi(x, y) \wedge Acyclic(R)$ and $\boldsymbol{k}$ – where $\Phi(x, y)$ is a quantifier-free formula and $\boldsymbol{k}$ is a 1-type cardinality vector, such that $|\boldsymbol{k}|= n$ – and returns $\text{WFOMC}(\Psi, \boldsymbol{k})$. In line 3, an array A with u indices is initiated and $A[\mathbf{0}]$ is assigned the value 1, where $\mathbf{0}$ corresponds to the u dimensional zero vector. The for loop in line $5 - 7$ incrimentally computes $\text{WFOMC}(\Psi, \boldsymbol{p})$, where the loop runs over all u-dimensional integer vectors $\boldsymbol{p}$, such that $p_i \leq k_i$, in lexicographical order. The number of possible $\boldsymbol{p}$ vectors is atmost n^u. Hence, the for loop in line 5 runs at most n^u iterations. In line 6, we compute $\text{WFOMC}(\Psi, \boldsymbol{p})$ as given in equation (4.11). Also in line 6, the function $\overline{\text{WFOMC}}(\Psi_{[m]}, \boldsymbol{p})$ — that computes $\text{WFOMC}(\Psi_{[m]}, \boldsymbol{p})$ —is called at most $|\boldsymbol{p}|-1$ times, which is bounded above by n. $A[\boldsymbol{p}]$ stores the value $\text{WFOMC}(\Psi, \boldsymbol{p})$. Hence, as $\boldsymbol{p}$ increments in lexicographical order, $A[\boldsymbol{p}]$, stores the value of $\text{WFOMC}(\Psi_{[m]}, \boldsymbol{p})$. In the function $\overline{\text{WFOMC}}(\Psi_{[m]}, \boldsymbol{s})$, the number of iterations in the for loop is bounded above by n^{2u}. And $\text{WFOMC}(\Psi', \boldsymbol{s}')$ is an FO^2 WFOMC problem, again computable in polynomial time. Hence, the algorithm WFOMC-DAG runs in polynomial time w.r.t domain cardinality. Notice that since loop 5-7 runs in lexicographical order, the $A[\boldsymbol{s}'']$ required in the function $\overline{\text{WFOMC}}(\Psi_{[m]}, \boldsymbol{s})$ are always already stored in A. Now, there are only polynomially many $\boldsymbol{k}$ w.r.t domain cardinality. Hence, computing $\text{WFOMC}(\Psi, \boldsymbol{k})$ over

Algorithm 2 WFOMC-DAG

Input: $\Psi, \boldsymbol{k}$
Output: $\textsc{wfomc}(\Psi, \boldsymbol{k})$
$A[\mathbf{0}] \leftarrow 1$ ▷ A has u indices
▷ $\mathbf{0} = \langle 0, ..., 0\rangle$
for $\mathbf{0} < \boldsymbol{p} \leq \boldsymbol{k}$ where $\boldsymbol{p} \in \mathbb{N}_0^u$ **do** ▷ Lexical order
  $A[\boldsymbol{p}] \leftarrow \sum_{l=0}^{|\boldsymbol{p}|-1} (-1)^{|\boldsymbol{p}|-l+1} \binom{|\boldsymbol{p}|}{l} \overline{\textsc{wfomc}}(\Psi_{[|\boldsymbol{p}|-l]}, \boldsymbol{p})$
end for
return $A[\boldsymbol{k}]$
function $\overline{\textsc{wfomc}}(\Psi_{[m]}, \boldsymbol{s})$ ▷ Equation (5.7)
  $S = 0$
  for $\boldsymbol{s}' + \boldsymbol{s}'' = \boldsymbol{s}$ and $|\boldsymbol{s}'| = m$ **do**
    $S \leftarrow S + \prod_{i,j \in [u]} r_{ij}^{s'_i s''_j} \cdot \textsc{wfomc}(\Psi', \boldsymbol{s}') \cdot A[\boldsymbol{s}'']$
  end for
  return S
end function

all possible $\boldsymbol{k}$ values, we can compute $\textsc{wfomc}(\Psi, n)$ in polynomial time w.r.t domain cardinality. Furthermore, using the modular WFOMC preserving skolemization process as provided in [36], we can easily extend this result to the entire FO^2 fragment. Hence, leading to the following theorem:

Theorem 18. *Let $\Psi = \Phi \wedge Acyclic(R)$, where Φ is an FO^2 formula. Then* $\textsc{wfomc}(\Psi, n)$ *can be computed in polynomial time with respect to the domain cardinality.*

Using Theorem 16 and Remark 6, we can also extend domain-liftability of FO^2, with DAG axiom and cardinality constraints.

Theorem 19. *Let $\Psi = \Phi \wedge Acyclic(R)$, where Φ is an FO^2 formula, potentially also containing cardinality constraints. Then* $\textsc{wfomc}(\Psi, n)$ *can be computed in polynomial time with respect to the domain cardinality.*

Furthermore, since WFOMC of any C^2 formula can be modularly reduced to WFOMC of an FO^2 formula with cardinality constraints [19]. We also have the following theorem:

Theorem 20. *Let $\Psi = \Phi \wedge Acyclic(R)$, where Φ is an C^2 formula. Then* $\textsc{wfomc}(\Psi, n)$ *can be computed in polynomial time with respect to the domain cardinality.*

4.4.1 Source and Sink

Definition 33. *Let Φ be a first order sentence, possibly containing some binary relation R, a unary relation $Source$ and a unary relation $Sink$. Then a structure ω is a model of $\Psi = \Phi \land Acyclic(R, Source, Sink)$ if and only if:*

- *ω is a model of $\Phi \land Acyclic(R)$, and*
- *In the DAG represented by ω_R, the sources of the DAG are interpreted to be true in ω_{Source}.*
- *In the DAG represented by ω_R, the sinks of the DAG are interpreted to be true in ω_{Sink}.*

The $Source$ and the $Sink$ predicate can allow encodicng constraints like $\exists^{=k}x.Source(x)$ or $\exists^{=k}x.Sink(x)$.

Theorem 21. *Let $\Psi = \Phi \land Acyclic(R, Source, Sink)$, where Φ is a C^2 formula. Then* $\text{WFOMC}(\Psi, n)$ *can be computed in polynomial time with respect to the domain cardinality.*

Proof. The sentence Ψ can be equivalently written as:

$$\begin{aligned} &\Phi \land Acyclic(R) \\ &\land \forall x.Source(x) \leftrightarrow \neg\exists y.R(y, x) \\ &\land \forall x.Sink(x) \leftrightarrow \neg\exists y.R(x, y) \end{aligned} \tag{4.12}$$

which is a FO^2 sentence extended with DAG constraint. □

4.5 Conclusion

In this chapter we demonstrate the domain liftability of FO^2 and C^2 extended with a Directed Acyclic Graph Axiom. We then extend our results with Source and Sink predicates, which can allow additional constraints on the number of sources and sinks in a DAG. These results can potentially allow better modelling of datasets that naturally appear with a DAG structure [50]. In future, we aim at investigating successor, predecessor and ancestory constraints in FOL extended with DAG axioms.

Chapter 5

WFOMC with Connectivity Constraint

5.1 Introduction

A *connected graph* is a graph in which there is a path between any two vertices. A large set of structures/datasets in real-world are represented by connected graphs e.g. social networks [51], transportation networks [52], electrical networks [53] e.t.c. Furthermore, connectivity can also be a very safty-critical property to check [54]. Even though most of the aforementioned structures can be seen as relational data, analyzing these structures for connectivity is limited in most SRL models, as First-Order Logic and hence the language of most SRL models cannot express connectivity [14]. In this chapter we aim to make initial steps towards resolving this problem by expanding the class of domain-liftable languages with a connectivity constraint, i.e., a predicate in the language represents a connected graph.

5.2 Background

We assume background from Section 2.1, Section 2.2, Section 2.7 and Section 4.3. We also present the required combinatorial background on counting Connected Graphs.

5.2.1 Counting Connected Graphs

A *connected graph* on $[n]$ is a simple undirected graph such that for any pair of nodes i and j, there exists a path connecting the two nodes. In a given graph, a *connected component* is a subgraph that is not part of any larger connected subgraph. In a *rooted graph* one node is labeled in a special way to distinguish it from other nodes, the special node is called the *root* of the graph. Given a rooted graph, we call its connected component containing the root as the *rooted-connected component.*

Proposition 6. *The number of rooted graphs on $[n]$, such that the subgraph on $[m]$ is a rooted-connected component is given as:*

$$m \cdot c_m \cdot 2^{\binom{n-m}{2}} \tag{5.1}$$

where c_m is the number of connected graphs on $[m]$.

Proof. Let ω be a rooted graph such that $\omega \downarrow [m]$ forms a rooted-connected component. Since, $\omega \downarrow [m]$ is a connected component, there can be no edges between $[m]$ and $[\bar{m}]$. The number of possible connected graphs on $[m]$ is given by c_m. Also in ω any node in $[m]$ can be chosen to be the root. Hence, the number of ways in which $\omega \downarrow [m]$ can be a rooted-connected component is $m \cdot c_m$. Since $\omega \downarrow [m]$ is a connected-component, there can be no edges between $[m]$ and $[\bar{m}]$, and $\omega \downarrow [\bar{m}]$ can be any $n - m$ sized graph. Hence, $2^{\binom{n-m}{2}}$ subgraphs can be realized on $[\bar{m}]$. Since, subgraphs on $[m]$ and $[\bar{m}]$ are realized independently, the total number of graphs on $[n]$ such that $[m]$ is a rooted-connected component is given by expression (5.1). □

The arguments used in Proposition 6 can be repeated for any rooted graph on $[n]$, with a rooted-connected component of size m. Since, there are $\binom{n}{m}$ ways of choosing such subsets, we have the following Proposition.

Proposition 7. *The number of rooted graphs on $[n]$ with an m sized rooted-connected component is given as:*

$$\binom{n}{m} \cdot m \cdot c_m \cdot 2^{\binom{n-m}{2}} \tag{5.2}$$

where c_m is the number of connected graphs on m nodes.

Summing up equation (5.2) over all m, for $1 \leq m \leq n$. We get the following proposition.

Proposition 8. *Let c_m be the number of connected graphs on m nodes. Then the following holds:*

$$n \cdot 2^{\binom{n}{2}} = \sum_{m=1}^{n} \binom{n}{m} \cdot m \cdot c_m \cdot 2^{\binom{n-m}{2}} \tag{5.3}$$

Proof. Using Proposition 7, the RHS of equation (5.3) sums over the number of rooted graphs with a rooted-connected component of size m. But any rooted graph on $[n]$ consists of a rooted-connected component of some size m, where $1 \leq m \leq n$. Hence,

Algorithm 3 Number of connected graphs on $[n]$

1: **Input**: n
2: **Output**: c_n
3: $c[1] \leftarrow 1$ ▷ c is an array
4: **for** $i = 2$ to n **do**
5: $\quad c[i] \leftarrow 2^{\binom{i}{2}} - \frac{1}{i}\sum_{m=1}^{i-1}\binom{i}{m} \cdot m \cdot c[m] \cdot 2^{\binom{i-m}{2}}$
6: **end for**
7: **return** $c[n]$

the RHS is counting all possible rooted graphs on $[n]$, which is equal to $n \cdot 2^{\binom{n}{2}}$ i.e., the LHS. □

Clearly the RHS of equation (5.3) can be written as:

$$n \cdot 2^{\binom{n}{2}} = n \cdot c_n + \sum_{m=1}^{n-1}\binom{n}{m} \cdot m \cdot c_m \cdot 2^{\binom{n-m}{2}} \tag{5.4}$$

which gives us the following equation for counting connected graphs:

$$c_n = 2^{\binom{n}{2}} - \frac{1}{n}\sum_{m=1}^{n-1}\binom{n}{m} \cdot m \cdot c_m \cdot 2^{\binom{n-m}{2}} \tag{5.5}$$

Remark 7. *The number of connected graphs on one node, i.e. c_1, is vacuously equal to 1.*

Theorem 22. *The number of connected graphs on $[n]$ can be computed in polynomial time w.r.t n.*

Proof. Algorithm 4 takes in as input n the number of nodes and returns the number of connected graphs on $[n]$. The algorithm incrementally computes the number of connected graphs on i nodes, from i equal to 2 to i equal to n. Storing the number of connected graphs on i nodes in the i^{th} position of an array c. Hence, in each iteration of for loop on line 5, the $c[m]'s$ used in line 5 are already stored in the array c. Hence, the algorithm runs n iterations of the for loop and in each loop at most $O(n)$ operations are performed. Hence, the algorithm runs in $O(n^2)$.

□

5.3 WFOMC with Connectivity Axiom

In this section we extend the approach used for counting connected graphs in equation (5.5) to WFOMC of FO^2 and C^2 formulas with a connectivity axiom.

Definition 34. *Let R be a binary predicate, an interpretation ω is a model of $Connected(R)$ if*

- ω_R *forms a symmetric and antireflexive relation of R, and*
- ω_R *forms a connected graph*

Definition 35. *For any subset $C \subseteq [n]$, ω is a model of $Connected(R, C)$ if*

- $\omega_R \downarrow C$ *forms a symmetric and antireflexive relation of R, and*
- $\omega_R \downarrow C$ *is a connected component of ω_R*

Notice that $Connected(R)$ is equivalent to $Connected(R, [n])$.

Remark 8. *Since our goal is to compute WFOMC of $\Psi = \Phi \wedge Connected(R)$, which by definition entails that ω_R forms a symmetric and anti-reflexive relation of R. We can assume without loss of generality that:*

$$\Phi \models \forall x.\neg R(x, x) \wedge \forall xy.R(x, y) \rightarrow R(y, x)$$

.

Proposition 9. *Let $\Psi := \forall xy.\Phi(x, y) \wedge Connected(R)$ and let $\Psi_{[m]} := \forall xy.\Phi(x, y) \wedge Connected(R, [m])$ be two sentences interpreted over $[n]$. Let $1 \leq m \leq n$. If ω is a model of $\Psi_{[m]}$, then $\omega \downarrow [m] \models \Psi$ and $\omega \downarrow [\bar{m}] \models \forall xy.\Phi(x, y)$.*

Proof. We have that $\omega \models \Psi_{[m]}$. Hence, we have that:

$$\begin{aligned} & \omega \models \bigwedge_{(c,d)\in[n]^2} \Phi(c, d) \\ \Rightarrow\ & \omega \models \bigwedge_{(c,d)\in[m]^2} \Phi(c, d) \bigwedge_{(c,d)\in[\bar{m}]^2} \Phi(c, d) \\ & \bigwedge_{(c,d)\in[\bar{m}]\times[m]} \Phi(c, d) \bigwedge_{(c,d)\in[m]\times[\bar{m}]} \Phi(c, d) \end{aligned}$$

Since, $\omega \models \bigwedge_{(c,d)\in[m]^2} \Phi(c, d)$ and $\omega \models \bigwedge_{(c,d)\in[\bar{m}]^2} \Phi(c, d)$, we have that $\omega \downarrow [m] \models \forall xy.\Phi(x, y)$ and $\omega \downarrow [\bar{m}] \models \forall xy.\Phi(x, y)$. Since $[m]$ is a connected component in ω_R, we can infer that $\omega \downarrow [m] \models \Psi$. $\square$

Proposition 10. *Let* $\Psi := \forall xy.\Phi(x,y) \wedge Connected(R)$ *and let* $\Psi_{[m]} := \forall xy.\Phi(x,y) \wedge Connected(R,[m])$ *be two sentences interpreted over* $[n]$*, where* $\Phi(x,y)$ *is quantifier-free formula such that:*

$$\begin{aligned}\forall xy.\Phi(x,y) \models & \forall xy.\neg R(x,x) \wedge \\ & \forall xy.R(x,y) \rightarrow R(y,x)\end{aligned}$$

Let ω' *be a model of* Ψ *on the domain* $[m]$ *and let* ω'' *be a model of* $\forall xy.\Phi(x,y)$ *on the domain* $[\bar{m}]$*. Then the WFOMC of the extensions* ω *on* $[n]$*, of* $\omega' \uplus \omega''$*, such that* $\omega \models \Psi_{[m]} \wedge \boldsymbol{k}$ *is given as:*

$$\mathrm{W}(\omega') \cdot \mathrm{W}(\omega'') \cdot \prod_{i,j \in [u]} r_{ij}^{k'_i \cdot k''_j} \tag{5.6}$$

where k'_i *and* k''_j *are the number of domain constants realizing the* i^{th} *and* j^{th} *1-types in* ω' *and* ω'' *respectively, and* $r_{ij} = \sum_{l \in [b]} n_{ijl} v_l$*, where* n_{ijl} *is* 1 *if* $ijl(x,y) \models \Phi(\{x,y\}) \wedge \neg R(x,y)$ *and* 0 *otherwise and* v_l *is as defined in equation* (4.1)*.*

Proof. In order to obtain an interpretation $\omega \models \Psi_{[m]} \wedge \boldsymbol{k}$ on the domain $[n]$, from $\omega' \uplus \omega''$, we only need to extend $\omega' \uplus \omega''$ with interpretations of the ground-atoms containing $(c,d) \in [m] \times [\bar{m}]$. For a given pair $(c,d) \in [m] \times [\bar{m}]$, let $\omega' \models i(c)$ and $\omega'' \models j(d)$. Since ω is a model of $\forall xy.\Phi(x,y)$, we must have that $ijl(c,d) \models \Phi(\{c,d\})$. Furthermore, since we want that $\omega_R \downarrow [m]$ forms a connected component in ω, we cannot have $R(c,d)$. Hence, we must have that $ijl(c,d) \models \Phi(\{c,d\}) \wedge \neg R(c,d)$. Hence, (c,d) can realize the l^{th} 2-table only if $n_{ijl} = 1$. Hence, the multiplicative weight contribution to the weight of a given extension ω of $\omega' \uplus \omega''$, due to (c,d) realizing the l^{th} 2-table is given as $n_{ijl} v_l$. Furthermore, (c,d) realizes the 2-tables mutually-exclusively. Also, the weight contribution of the 2-table realizations of (c,d), $n_{ijl} v_l$, only depends on the 1-types of c and d and is independent of all other domain constants. Now, ω' and ω'' interpret completely independent set of ground atoms in any extension of $\omega' \uplus \omega''$, hence, their weights contribute independently as $\mathrm{W}(\omega')$ and $\mathrm{W}(\omega'')$ in any such extension. Finally, since there are k'_i domain elements c realizing the i^{th} 1-type in ω' and k''_j domain elements d realizing the j^{th} 1-type in ω'', the WFOMC of the extensions ω, of $\omega' \uplus \omega''$, such that $\omega \models \Psi_{[m]} \wedge \boldsymbol{k}$ is given by:

$$\mathrm{W}(\omega') \cdot \mathrm{W}(\omega'') \cdot \prod_{i,j \in [u]} \Big(\sum_{l \in [b]} n_{ijl} v_l\Big)^{k'_i \cdot k''_j}$$

Which can be equivalently written as expression (5.6). □

Proposition 11. *Let* $\Psi := \forall xy.\Phi(x,y) \wedge Connected(R)$ *and let* $\Psi_{[m]} := \forall xy.\Phi(x,y) \wedge Connected(R,[m])$ *be two sentences interpreted over* $[n]$*, where* $\Phi(x,y)$ *is quantifier-free formula such that:*

$$\begin{aligned}\forall xy.\Phi(x,y) \models &\forall x.\neg R(x,x) \wedge \\ &\forall xy.R(x,y) \rightarrow R(y,x)\end{aligned}$$

Then:

$$\begin{aligned}&\textsc{wfomc}(\Psi_{[m]}, \boldsymbol{k}) = \\ &\sum_{\substack{\boldsymbol{k}=\boldsymbol{k}'+\boldsymbol{k}''\\ |\boldsymbol{k}'|=m}} \prod_{i,j} r_{ij}^{k'_i k''_j} \textsc{wfomc}(\Psi, \boldsymbol{k}')\textsc{wfomc}(\forall xy.\Phi(x,y), \boldsymbol{k}'')\end{aligned} \tag{5.7}$$

where $\boldsymbol{k}' + \boldsymbol{k}''$ *represents the element-wise sum of integer-vectors* $\boldsymbol{k}'$ *and* $\boldsymbol{k}''$*, such that* $|\boldsymbol{k}'| = m$ *and* $|\boldsymbol{k}''| = |\boldsymbol{k}| - m$*. Also,* $r_{ij} = \sum_l n_{ijl} v_l$*, where* n_{ijl} *is* 1 *if* $ijl(x,y) \models \Phi(\{x,y\}) \wedge \neg R(x,y)$ *and* 0 *otherwise.*

Proof. Due to Proposition 10, each pair of models $\omega' \models \Psi \wedge \boldsymbol{k}'$ and $\omega'' \models \forall xy.\Phi(x,y) \wedge \boldsymbol{k}''$, interpreted on $[m]$ and $[\bar{m}]$ respectively, can be extended to multiple models ω of $\Psi_{[m]}$ on $[n]$. The weighted sum of all such extensions, for a given pair ω' and ω'' is given as expression (5.6). Furthermore, due to Proposition 9, if $\omega \models \Psi_{[m]}$, then $\omega \downarrow [m] \models \Psi$ and $\omega \downarrow [\bar{m}] \models \forall xy.\Phi(x,y)$. Hence, the WFOMC of interpretations $\omega \models \Psi_{[m]}$, such that $\omega \downarrow [m] \models \boldsymbol{k}'$ and $\omega \downarrow [\bar{m}] \models \boldsymbol{k}''$, for fixed $\boldsymbol{k}'$ and $\boldsymbol{k}''$, is given as:

$$\begin{aligned}&\sum_{\substack{\omega' \models \Psi \wedge \boldsymbol{k}'\\ \omega'' \models \forall xy.\Phi(x,y) \wedge \boldsymbol{k}''}} \prod_{i,j \in [u]} r_{ij}^{k'_i k''_j} \cdot \mathrm{W}(\omega') \cdot \mathrm{W}(\omega'') \\ &= \prod_{i,j \in [u]} r_{ij}^{k'_i k''_j} \cdot \sum_{\substack{\omega' \models \Psi \wedge \boldsymbol{k}'\\ \omega'' \models \forall xy.\Phi(x,y) \wedge \boldsymbol{k}''}} \mathrm{W}(\omega') \cdot \mathrm{W}(\omega'') \\ &= \prod_{i,j \in [u]} r_{ij}^{k'_i k''_j} \cdot \sum_{\omega' \models \Psi \wedge \boldsymbol{k}'} w(\omega') \cdot \sum_{\omega'' \models \forall xy.\Phi(x,y) \wedge \boldsymbol{k}''} \mathrm{W}(\omega'') \\ &= \prod_{i,j \in [u]} r_{ij}^{k'_i k''_j} \cdot \textsc{wfomc}(\Psi, \boldsymbol{k}') \cdot \textsc{wfomc}(\forall xy.\Phi(x,y), \boldsymbol{k}'')\end{aligned}$$

The WFOMC of Ψ on $[m]$, with 1-type cardinality vector $\boldsymbol{k}'$, is given as $\textsc{wfomc}(\Psi, \boldsymbol{k}')$. Similarly, the number models of $\forall xy.\Phi(x,y)$ on $[\bar{m}]$, with 1-type cardinality vector $\boldsymbol{k}''$ is given as $\textsc{wfomc}(\Psi, \boldsymbol{k}'')$. Due to Proposition 10, the multiplicative-weight contribution due to the extensions of each pair of models counted in $\textsc{wfomc}(\Psi, \boldsymbol{k}')$ and $\textsc{wfomc}(\Psi, \boldsymbol{k}'')$, to models the models of $\Psi_{[m]} \wedge \boldsymbol{k}$ is given as $\prod_{i,j\in[u]} r_{ij}^{k'_i k''_j}$. In order to compute WFOMC of $\Psi_{[m]} \wedge \boldsymbol{k}$, we sum over all possible $\boldsymbol{k}'$ and $\boldsymbol{k}''$, such that $\boldsymbol{k}' + \boldsymbol{k}'' = \boldsymbol{k}$ and $|\boldsymbol{k}'| = m$. Hence, giving us equation (5.7). can be extended in $\prod_{i,j\in[u]} n_{ij}^{k'_i k''_j}$ ways. The summation in (5.7) runs over all possible realizable 1-type cardinalities over $[m]$ and $[\bar{m}]$, represented by $\boldsymbol{k}'$ and $\boldsymbol{k}''$ respectively, such that they are consistent with $\boldsymbol{k}$, i.e. when $\boldsymbol{k} = \boldsymbol{k}' + \boldsymbol{k}''$. Hence, formula (5.7) gives us the total number of models ω, such that $\omega \downarrow [m] \models \Psi'$, $\omega \downarrow [\bar{m}] \models \Psi$ and $\omega \models \Psi \wedge \boldsymbol{k}$ where the domain constant in $[m]$ have zero R indegree. Due to proposition 2, we have that these are all the models such that $\omega \models \Psi \wedge \boldsymbol{k}$ and the domain constants in $[m]$ have zero R indegree. □

Due to modularity of the skolemization process for WFOMC [36], in Proposition 11, $\forall xy.\Phi(x, y)$ can be replaced by any FO^2 sentence.

We will also use the notion of ω_R representing a rooted graph. This can be easily modelled in FOL by introducing a fresh new predicate $Root(x)$ and adding a conjunct to any sentence, where a root needs to exist w.r.t ω_R, a formula $\exists^{=1}x.Root(x)$. This can be modelled with cardinality constraints as $\exists x.Root(x) \wedge |Root| = 1$. However, this is rather tedious and we will use the notion of a graph and a connected component being rooted without this formal discription.

Proposition 12. *Let $\Psi := \Phi \wedge Connected(R)$ and let $\Psi_{[m]} := \Phi \wedge Connected(R, [m])$ be two sentences interpreted over $[n]$, where Φ is an FO^2 sentence such that:*

$$\begin{aligned}\Phi \models &\forall x.\neg R(x, x) \wedge \\ &\forall xy.R(x, y) \rightarrow R(y, x)\end{aligned}$$

Then the WFOMC of the models $\omega \models \Psi_{[m]}$, such that $\omega_R \downarrow [m]$ forms a rooted-connected component is given as:

$$m \cdot \textsc{wfomc}(\Psi_{[m]}, \boldsymbol{k}) \tag{5.8}$$

Proof. The proof idea is identical to the one of Proposition 6. Since, $\omega_R \downarrow [m]$ forms a rooted-connected component in $\omega \models \Psi_{[m]}$, we have m choices for selecting any node in $[m]$ as a root. Hence, the number of models of $\Psi_{[m]}$, where $[m]$ represents a rooted-connected component is given as expression (5.8). □

Proposition 13. *Let* $\Psi := \Phi \wedge Connected(R)$ *and let* $\Psi_{[m]} := \Phi \wedge Connected(R, [m])$ *be two sentences interpreted over* $[n]$*, where* Φ *is an* FO^2 *sentence such that:*

$$\begin{aligned}\Phi \models & \forall x. \neg R(x,x) \wedge \\ & \forall xy. R(x,y) \rightarrow R(y,x)\end{aligned}$$

Then the WFOMC of the models $\omega \models \Phi$ *with a rooted-connected component of size* m *is given as:*

$$\binom{n}{m} \cdot m \cdot \textsc{wfomc}(\Psi_{[m]}, \boldsymbol{k}) \tag{5.9}$$

Proof. The proof idea is identical to the one of Proposition 7. $\textsc{wfomc}(\Psi_{[m]}, \boldsymbol{k})$ is the same as $\textsc{wfomc}(\Psi_C, \boldsymbol{k})$, where $\Psi_C := \Phi(x,y) \wedge Connected(R, C)$, for any $C \subseteq [n]$, where $|C| = m$. Furthermore, there are $\binom{n}{m}$ ways of choosing C in $[n]$. And in each Ψ_C has $|C| = m$ choices for selecting a root in C. □

Proposition 14. *Let* $\Psi = \Phi \wedge Connected(R)$*, where* Φ *is an* FO^2 *sentence, such that:*

$$\begin{aligned}\Phi \models & \forall x. \neg R(x,x) \wedge \\ & \forall xy. R(x,y) \rightarrow R(y,x)\end{aligned}$$

then the following holds:

$$n \cdot \textsc{wfomc}(\Phi, \boldsymbol{k}) = \sum_{m=1}^{n} \binom{n}{m} \cdot m \cdot \textsc{wfomc}(\Psi_{[m]}, \boldsymbol{k}) \tag{5.10}$$

where $\Psi_{[m]} := \Phi \wedge Connected(R, [m])$*.*

Proof. The proof idea is identical to the one of Proposition 8. Using Proposition 13, the RHS of (5.10), sums over the WFOMC of all the models of Φ, where ω_R is a simple graph with a R-rooted-connected component of size m, where $1 \leq m \leq n$. But any model of Φ, where ω_R is a rooted graph, consists of some R-rooted-connected component of size m, where $1 \leq m \leq n$. Hence, the RHS of equation (5.10), computes the weighted sum of all models of Φ where ω_R is a rooted graph. But this is equal to n times the WFOMC of Φ, because we have n choices for assigning a root in each model of Φ. □

Algorithm 4 WFOMC-Connected

1: **Input**: $\Psi := \Phi \wedge \mathit{Connected}(R), \boldsymbol{k}$
2: **Output**: $\textsc{wfomc}(\Psi, \boldsymbol{k})$
3: $A[\mathbf{0}] \leftarrow 0$ ▷ A has u indices
4: ▷ $\mathbf{0} = \langle 0, ..., 0\rangle$
5: **for** $\mathbf{0} < \boldsymbol{p} \leq \boldsymbol{k}$ where $\boldsymbol{p} \in \mathbb{N}_0^u$ **do** ▷ $\leq$ is Lexical order
6:
7: $\quad A[\boldsymbol{p}] \leftarrow \textsc{wfomc}(\Phi, \boldsymbol{p}) - \frac{1}{|\boldsymbol{p}|}\sum_{m=1}^{|\boldsymbol{p}|-1}\binom{|\boldsymbol{p}|}{m} m \overline{\textsc{wfomc}}(\Psi_{[m]}, \boldsymbol{p})$
8: ▷ Equation (5.12)
9: **end for**
10: **return** $A[\boldsymbol{k}]$
11:
12: **function** $\overline{\textsc{wfomc}}(\Psi_{[m]}, \boldsymbol{s})$
13: $\quad S = 0$
14: $\quad$ **for** $\boldsymbol{s}' + \boldsymbol{s}'' = \boldsymbol{s}$ and $|\boldsymbol{s}'| = m$ **do**
15: $\qquad S \leftarrow S + \prod_{i,j \in [u]} r_{ij}^{s'_i s''_j} \cdot A[\boldsymbol{s}'] \cdot \textsc{wfomc}(\Phi, \boldsymbol{s}'')$
16: ▷ Equation (5.7)
17: $\quad$ **end for**
18: $\quad$ **return** S
19: **end function**

The RHS of equation (5.10) can be written as:

$$
\begin{aligned}
&n \cdot \textsc{wfomc}(\Phi, \boldsymbol{k}) = \\
&n \cdot \textsc{wfomc}(\Psi_{[n]}, \boldsymbol{k}) + \sum_{m=1}^{n-1}\binom{n}{m} \cdot m \cdot \textsc{wfomc}(\Psi_{[m]}, \boldsymbol{k})
\end{aligned}
\tag{5.11}
$$

Notice that $\Psi_{[n]}$ is equivalent to Ψ. Hence, we get the following:

$$
\begin{aligned}
&\textsc{wfomc}(\Psi, \boldsymbol{k}) = \\
&\textsc{wfomc}(\Phi, \boldsymbol{k}) - \frac{1}{n}\sum_{m=1}^{n-1}\binom{n}{m} \cdot m \cdot \textsc{wfomc}(\Psi_{[m]}, \boldsymbol{k})
\end{aligned}
\tag{5.12}
$$

WFOMC-Connected (4) takes as input $\Psi = \Phi \wedge \mathit{Connected}(R)$ and $\boldsymbol{k}$ – where Φ is an FO^2 formula and $\boldsymbol{k}$ is a 1-type cardinality vector, such that $|\boldsymbol{k}| = n$ – and returns $\textsc{wfomc}(\Psi, \boldsymbol{k})$. In line 3, an array A with u indices is initiated and $A[\mathbf{0}]$ is assigned the value 0, where $\mathbf{0}$ corresponds to the u dimensional zero vector. The for loop in line $5-9$ incrementally computes $\textsc{wfomc}(\Psi, \boldsymbol{p})$, using equation (5.12), where the loop runs over all u-dimensional integer vectors $\boldsymbol{p}$, such that $\boldsymbol{p} \leq \boldsymbol{k}$, where $\leq$ is the lexicographical order. The number of possible $\boldsymbol{p}$ vectors is at most n^u. Hence, the for loop in line 5 runs at most n^u iterations. In line 7, we compute $\text{WFOMC}(\Psi, \boldsymbol{p})$ as given in equation (5.12). Also in line 7, the function $\overline{\text{WFOMC}}(\Psi_{[m]}, \boldsymbol{p})$ — that computes $\text{WFOMC}(\Psi_{[m]}, \boldsymbol{p})$ —is called at most $|\boldsymbol{p}|-1$ times, which is bounded above by n. $A[\boldsymbol{p}]$ stores the value $\text{WFOMC}(\Psi, \boldsymbol{p})$. In the function $\overline{\text{WFOMC}}(\Psi_{[m]}, \boldsymbol{s})$, the number of iterations in the for loop is bounded above by n^{2u}. And $\text{WFOMC}(\Phi, \boldsymbol{s}'')$ is an FO^2 WFOMC problem, again computable in polynomial time. Hence, the algorithm WFOMC-Connected runs in polynomial time w.r.t domain cardinality. Notice that since loop 5-7 runs in lexicographical order, the $A[\boldsymbol{s}']$ required in the function $\overline{\text{WFOMC}}(\Psi_{[m]}, \boldsymbol{s})$ are always already stored in A. Now, there are only polynomially many $\boldsymbol{k}$ w.r.t domain cardinality. Hence, computing $\text{WFOMC}(\Psi, \boldsymbol{k})$ over all possible $\boldsymbol{k}$ values, we can compute $\text{WFOMC}(\Psi, n)$ in polynomial time w.r.t domain cardinality. Giving us the following theorem:

Furthermore, using the modular WFOMC preserving skolemization process as provided in [36], we can easily extend this result to the entire FO^2 fragment. Hence, leading to the following theorem:

Theorem 23. *Let $\Psi := \Phi \wedge Connected(R)$, where Φ is an FO^2 formula. Then* $\text{WFOMC}(\Psi, n)$ *can be computed in polynomial time with respect to the domain cardinality.*

Using Theorem 16 and Remark 6, we can also extend domain-liftability of FO^2, with Connected Graph Axiom and cardinality constraints.

Theorem 24. *Let $\Psi := \Phi \wedge Connected(R)$, where Φ is an FO^2 formula, potentially also containing cardinality constraints. Then* $\text{WFOMC}(\Psi, n)$ *can be computed in polynomial time with respect to the domain cardinality.*

Furthermore, since WFOMC of any C^2 formula can be modularly reduced to WFOMC of an FO^2 formula with cardinality constraints [19]. We also have the following theorem:

Theorem 25. *Let $\Psi := \Phi \wedge Connected(R)$, where Φ is an C^2 formula. Then* $\text{WFOMC}(\Psi, n)$ *can be computed in polynomial time with respect to the domain cardinality.*

5.4 Conclusion

In this chapter, we demonstrate the domain liftability of FO^2 and C^2 extended with a Connectivity Axiom. These results can potentially allow better modelling of datasets that naturally appear with connected structures. In future, we aim at investigating constraints like $k-$connectivity. On the application side, we believe that our work can be used in efficiently computing partition function in Markov Logic Networks and also in checking robustness of connectivity in real-world networks.

Chapter 6

Projectivity in MLNs

The work presented in this Chapter is based on the following publication:

Sagar Malhotra and Luciano Serafini. Projectivity in Markov Logic Networks. *In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022* [25]

6.1 Introduction

Statistical Relational Learning [9, 10] (SRL) is concerned with representing and learning probabilistic models over relational structures. Many works have observed that SRL frameworks exhibit unwanted behaviors over varying domain sizes [55, 18]. These behaviors make models learned from a fixed or a sub-sampled domain unreliable for inference over larger (or smaller) domains [18]. Drawing on the works of Shalizi and Rinaldo [15] on Exponential Random Graphs (ERGMs), Jaeger and Schulte [16] have recently introduced the notion of *projectivity* as a strong form of guarantee for good scaling behavior in SRL models. A projective model requires that the probability of any given query, over arbitrary m domain objects, is completely independent of the domain size.

Jaeger and Schulte [16] identify restrictive fragments of SRL models to be projective. But whether these fragments are complete characterization of projectivity, remains an open problem.

In this chapter, our goal is to characterize projectivity for a specific class of SRL models, namely Markov Logic Networks (MLNs) [21]. MLNs are amongst the most prominent template-based SRL models. An MLN is a Markov Random Field with features defined in terms of function-free weighted First Order Logic (FOL) formulae. Jaeger and Schutle [16] show that an MLN is projective if - any pair of atoms in each of its formulae share the same set of variables. We show that this characterization is not complete. Furthermore, we completely characterize projectivity for the class of MLNs with at most 2 variables in their formulae. Our charecterization leads to a parametric restriction that can be easily incorporated into any MLN learning algorithm. We also identify a special class of projective models, namely the Relational Block Models (RBMs). Any projective MLN in the two variable fragment can be expressed as an RBM. We show that the training data likelihood due to the maximum likelihood RBM is greater than or equal to the training data likelihood due to any other projective MLN in the two variable fragment. RBMs also admit consistent maximum likelihood estimation. Hence, RBMs are projective models that admit consistent and efficient learning from sub-sampled domains.

The chapter is organized as follows: We first contextualize our work w.r.t the related works in this domain. We then provide some background and notation on FOL and relational structures. We also elaborate on the fragment of FOL with at most two variables i.e. FO^2 and define the notion of FO^2 interpretations as multi-relational graphs. We also overview some results on Weighted First Order Model Counting. In the subsequent section, we provide a parametric representation for any MLN in the two variable fragment. We then dedicate a section to the main result of this chapter i.e. the necessary and sufficient conditions for an MLN in the two variable fragment to be projective. Based on the projectivity criterions we identify a special class of models namely Relational Block Models. We dedicate a complete section to RBMs and elaborate on their useful properties. We then move on to a formal comparison between the previous characterizations and the presented characterization of projectivity in MLNs. Finally, we discuss the consistency and efficiency aspects of learning for projective MLNs and RBMs.

6.2 Related Work

Projectivity has emerged as a formal notion of interest through multiple independent lines of works across ERGM and SRL literature. The key focus of these works have been analyzing [56, 55] or mitigating [17, 18] the effects of varying domain sizes on relational models. The major step in formalizing the notion of projectivity can be attributed to Shalizi and Rinaldo [15]. The authors both formalize and characterize the sufficient and necessary conditions for ERGMs to be projective. It is interesting to note that their projectivity criterion is strictly structural i.e. they put no restrictions on parameter values but rather inhibit the class of features that can be defined as sufficient statistics in ERGMs. In contrast our results w.r.t MLN are strictly parametric (which may correspond to non-trivial structural restrictions as well). With respect to SRL, the notion of projectivity was first formalized by Jaeger and Schulte [16], they show some restrictive fragments of SRL models to be projective. Jaeger and Schulte [57] significantly extend the scope of projective models by characterizing necessary and sufficient conditions for an arbitrary model on relational structures to be projective. Their characterization is expressed in terms of the so called AHK models. But as they conclude in [57], expressing AHK models in existing SRL frameworks remains an open problem. Hence, a complete characterization of projectivity in most SRL languages is still an open problem. Weitkamper [58] has shown that the characterization of projectivity provided by Jaeger and Schulte [16], for probabilistic logic programs under distribution semantics, is indeed complete. In this work, we will extend this characterization to the two variable fragment of Markov Logic Networks.

Another correlated problem to projectivity is learning from sub-sampled or smaller domains. In the relational setting projectivity is not a sufficient condition for consistent learning from sub-sampled domains [16]. Mittal et. al. have proposed a solution to this problem by introducing domain-size dependent scale-down factors [18] for MLN weights. Although empirically effective, the scale-down factors are not known to be a statistically sound solution. On the other hand, Kuzelka et. al. [59], provide a statistically sound approach to approximately obtain the correct distribution for a larger domain. But their approach requires estimating the relational marginal polytope for the larger domain and hence, offers no computational gains w.r.t learning from a sub-sampled domain. In this work, we will provide a statistically sound approach for efficiently estimating a special class of projective models (namely, RBM) from sub-sampled domains. We also show that our approach provides consistent parameter estimates in an efficient manner and is better than estimating any projective MLN in the two variable fragment (in terms of data likelihood maximisation).

6.3 Background

Interpretations as Multi-relational Graphs.

Given an FO^2 language $\mathcal{L}$ with interpretations defined over the domain $\Delta = [n]$, we can represent an interpretation $\omega \in \Omega^{(n)}$ as a multi-relational graph $(\boldsymbol{x}, \boldsymbol{y})$. This is achieved by defining $\boldsymbol{x} = (x_1, ..., x_n)$ such that $x_q = i$ if $\omega \models i(q)$ and by defining

$\boldsymbol{y} = (y_{12}, y_{13}, ...y_{qr}, ..., y_{n-1,n})$, where $q < r$, such that $y_{qr} = l$ if $\omega \models l(q,r)$. We also define $k_i = k_i(\boldsymbol{x}) = k_i(\omega) := |\{c \in \Delta : c \models i(c)\}|$, $h_l^{ij} = h_l^{ij}(\boldsymbol{y}) = h_l^{ij}(\omega) := |\{(c,d) \in \Delta^2 : \omega \models ijl(c,d)\}|$ and for any $D \subseteq \Delta^2$, $h_l^{ij}(D) = h_l^{ij}(\omega, D) := |\{(c,d) : \omega \models ijl(c,d) \text{ and } (c,d) \in D\}|$. Notice that $\sum_{i \leq j} \sum_{l \in [b]} h_l^{ij} = \binom{n}{2}$ and $\sum_{l \in [b]} h_l^{ij} = \boldsymbol{k}(i,j)$, where $\boldsymbol{k}(i,j)$ is defined in equation (3.16) . We use $(\boldsymbol{x}_{\mathrm{I}}, \boldsymbol{y}_{\mathrm{I}})$ to represent the multi-relational graph for $\omega \downarrow \mathrm{I}$. Throughout this chapter we will use an interpretation ω and it's multi-relational graph $(\boldsymbol{x}, \boldsymbol{y})$ interchangeably.

Families of Probability Distributions and Projectivity.

We will be interested in probability distributions over the set of interpretations or equivalently their multi-relational graphs. A family of probability distributions $\{P^{(n)} : n \in \mathbb{N}\}$ specifies, for each finite domain of size n, a distribution $P^{(n)}$ on the possible n-world set $\Omega^{(n)}$ [57]. We will mostly work with the so-called exchangeable probability distributions [57] i.e. distributions where $P^{(n)}(\omega) = P^{(n)}(\omega')$ if ω and ω' are isomorphic. A distribution $P^{(n)}(\omega)$ over n-worlds induces a marginal probability distribution over m-worlds $\omega' \in \Omega^{(m)}$ as follows:

$$P^{(n)} \downarrow [m](\omega') = \sum_{\omega \in \Omega^{(n)} : \omega \downarrow [m] = \omega'} P^{(n)}(\omega)$$

Notice that due to exchangeability $P^{(n)} \downarrow \mathrm{I}$ is the same for all subsets I of size m, hence we can always assume any induced m-world to be $\omega \downarrow [m]$. We are now able to define projectivity as follows:

Definition 36 ([57]). *An exchangeable family of probability distributions is called projective if for all $m < n$:*

$$P^{(n)} \downarrow [m] = P^{(m)}$$

When dealing with probability distributions over multi-relational representation, we denote by $(\boldsymbol{X}, \boldsymbol{Y})$ the random vector where, $\boldsymbol{X} = (X_1, \ldots, X_n)$ and each X_i takes value in $[u]$; and $\boldsymbol{Y} = (Y_{12}, Y_{13}, \ldots, Y_{qr}, \ldots, Y_{n-1,n})$ where $q < r$ and Y_{qr} takes values in $[b]$.

6.4 A Parametric Normal Form for MLNs

A Markov Logic Network (MLN) Φ is defined by a set of weighted formulas $\{(\phi_i, a_i)\}_i$, where ϕ_i are quantifier free, function-free FOL formulas with weights $a_i \in \mathbb{R}$. An MLN

Φ induces a probability distribution over the set of possible worlds $\omega \in \Omega^{(n)}$:

$$P_{\Phi}^{(n)}(\omega) = \frac{1}{Z(n)} \exp\Big(\sum_{(\phi_i, a_i) \in \Phi} a_i . N(\phi_i, \omega) \Big)$$

where $N(\phi_i, \omega)$ represents the number of true groundings of ϕ_i in ω. The normalization constant $Z(n)$ is called the *partition function* that ensures that $P_{\Phi}^{(n)}$ is a probability distribution.

Theorem 26. *Any Markov Logic Network (MLN) $\Phi = \{(\phi_i, a_i)\}_i$ on a domain of size n, such that ϕ_i contains at-most two variables, can be expressed as follows:*

$$P_{\Phi}^{(n)}(\omega) = \frac{1}{Z(n)} \prod_{i \in [u]} s_i^{k_i} \prod_{\substack{i,j \in [u] \\ i \leq j}} \prod_{l \in [b]} (t_{ijl})^{h_l^{ij}} \tag{6.1}$$

where s_i and t_{ijl} are positive real numbers and k_i is $k_i(\omega)$ and h_l^{ij} is equal to $h_l^{ij}(\omega)$.

Proof. Let $\Phi = \{(\phi_i, a_i)\}_i$ be an MLN, such that ϕ_i contains at-most two variables. Firstly, every weighted formula $(\phi(x,y), a) \in \Phi$ that contains exactly two variables is replaced by two weighted formulas $(\phi(x,x), a)$ and $(\phi(x,y) \wedge (x \neq y), a)$. The MLN distribution $P_{\Phi}^{(n)}$ is invariant under this transformation. Hence, Φ can be equivalently written as $\{(\alpha_q(x), a_q)\}_q \cup \{(\beta_p(x,y), b_p)\}_p$, where $\{\alpha_q(x)\}_q$ is the set of formulas containing only the variable x and $\{\beta_p(x,y)\}_p$ is the set of formulas containing both the variables x and y. Notice that every $\beta_p(x,y)$ entails $x \neq y$.

Let us have $\omega \in \Omega^{(n)}$, where we have a domain constant c such that $\omega \models i(c)$. Now notice that the truth value of ground formulas $\{\alpha_q(c)\}_q$ in ω is completely determined by $i(c)$ irrespective of all other domain constants. Hence, the (multiplicative) weight contribution of $i(c)$ to the weight of ω can be given as $\exp(\sum_q a_q \mathbb{1}_{i(x) \models \alpha_q(x)})$. We define s_i as follows:

$$s_i = \exp(\sum_q a_q \mathbb{1}_{i(x) \models \alpha_q(x)}) \tag{6.2}$$

Clearly, this argument can be repeated for all the domain constants realizing any 1-type in $[u]$. Hence, the (multiplicative) weight contribution due to 1-types of all domain constants and equivalently due to the groundings of all unary formulas, is given as $\prod_{i \in [u]} s_i^{k_i}$.

We are now left with weight contributions due to the binary formulas, given by the set $\{(\beta_p(x,y), b_p)\}_p$. Due to the aforementioned transformation, each binary formula $\beta(x,y)$ contains a conjunct $(x \neq y)$. Hence, all groundings of $\beta(x,y)$ such that both

x and y are mapped to the same domain constants evaluate to false. Hence, we can assume that x and y are always mapped to distinct domain constants. Let us have an unordered pair of domain constants $\{c, d\}$ such that $\omega \models ijl(c, d)$. The truth value of any binary ground formula $\beta(c, d)$ and $\beta(d, c)$ is completely determined by $ijl(c, d)$ irrespective of all other domain constants. Hence, the multiplicative weight contribution due to the ground formulas $\{\beta_p(c, d)\}_p \cup \{\beta_p(d, c)\}_p$ is given as t_{ijl}, where t_{ijl} is defined as follows:

$$\exp\left(\sum_p b_p \mathbb{1}_{ijl(x,y) \models \beta_p(x,y)} + \sum_p b_p \mathbb{1}_{ijl(x,y) \models \beta_p(y,x)}\right) \tag{6.3}$$

Hence, the weight of an interpretation ω under the MLN Φ is given as

$$\prod_{i \in [u]} s_i^{k_i} \prod_{\substack{i,j \in [u] \\ i \leq j}} \prod_{l \in [b]} (t_{ijl})^{h_l^{ij}}$$

□

Definition 37. *Given an MLN in the parametric normal form given by equation* (6.1)*. Then f_{ij} is defined as $\sum_{l \in [b]} t_{ijl}$.*

We will now provide the parameterized version of the partition function $Z(n)$ due to Theorem 26.

Proposition 15. *Let Φ be an MLN in the form* (6.1)*, then the partition function $Z(n)$ is given as:*

$$Z(n) = \sum_{\boldsymbol{k}} \binom{n}{\boldsymbol{k}} \prod_{i \in [u]} s_i^{k_i} \prod_{\substack{i,j \in [u] \\ i \leq j}} (f_{ij})^{\boldsymbol{k}(i,j)} \tag{6.4}$$

where $\boldsymbol{k}(i, j)$ is defined in equation (3.16)*.*

Sketch. The proposition is a parameterized version of Theorem 7, where $\prod_{i \in [u]} s_i^{k_i}$ takes into account the weight contributions due to the 1-type realizations and f_{ij} is essentially a weighted version of n_{ij} i.e. given a pair of constants c and d such that they realize the i^{th} and the j^{th} 1-type respectively, then f_{ij} is the sum of the weights due to the 2-types realized by the extensions to the binary predicates containing both c and d.

□

6.5 Projectivity in Markov Logic Networks

We present the necessary and sufficient conditions for an MLN to be projective in the two variable fragment. The complete proofs are provided in the appendix.

Lemma 10 (Sufficiency)**.** *A Markov Logic Network in the two variable fragment is projective if all the f_{ij} have the same value i.e. $\forall i, j \in [u] : f_{ij} = F$, for some positive real number F.*

Sketch. The key idea of the proof is that if $\forall i, j \in [u] : f_{ij} = F$, then the partition function factorizes as $Z(n) = (F)^{\binom{n}{2}} \left(\sum_{i\in[u]} s_i \right)^n$. Now, defining $p_i = \frac{s_i}{\sum_i s_i}$ and $w_{ijl} = \frac{t_{ijl}}{F}$, allows us to re-define the MLN distribution (6.1) equivalently as follows:

$$P_{\Phi}^{(n)}(\omega) = \prod_{i\in[u]} p_i^{k_i} \prod_{\substack{i,j\in[u]\\ i\leq j}} \prod_{l\in[b]} w_{ijl}^{h_l^{ij}} \tag{6.5}$$

Here, $\sum_i p_i = 1$ and $\sum_l w_{ijl} = 1$. Hence, $P_{\Phi}^{(n)}(\omega)$ is essentially a (labeled) stochastic block model, which are known to be projective [15]. □

We will now prove that the aforementioned sufficient conditions are also necessary.

Lemma 11 (Necessary)**.** *If a Markov Logic network in the two variable fragment is projective then, all the f_{ij} have the same value i.e. $\forall i, j \in [u] : f_{ij} = F$, for some positive real number F.*

Sketch. We begin by writing the projectivity condition in the multi-relational representation, i.e. $P_{\Phi}^{(n+1)} \downarrow [n](\boldsymbol{X}' = \boldsymbol{x}', \boldsymbol{Y}' = \boldsymbol{y}')$ is equal to:

$$\sum_{\substack{\boldsymbol{x}_{[n]}=\boldsymbol{x}'\\ \boldsymbol{y}_{[n]}=\boldsymbol{y}'}} P_{\Phi}^{(n+1)}(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{Y} = \boldsymbol{y}) \tag{6.6}$$

Multiplying and dividing equation (6.6) by $Z(n)$ and using simple algebraic manipulations we get that for all $\boldsymbol{x}'$:

$$\frac{Z(n+1)}{Z(n)} = \sum_{i\in[u]} s_i \prod_{j\in[u]} f_{ij}^{k_j(\boldsymbol{x}')} \tag{6.7}$$

Now, the LHS of equation (6.7) is completely independent of $\boldsymbol{x}'$, whereas RHS is dependent on $\boldsymbol{x}'$. It can be shown that this is possible iff f_{ij} does not depend on $\boldsymbol{x}'$, which in turn is possible iff f_{ij} does not depend on i and j i.e. $\forall i, j \in [u] : f_{ij} = F$, for some positive real F. □

We are finally able to provide the following theorem.

Theorem 27. *A Markov Logic Network (MLN) $\Phi = \{(\phi_i, a_i)\}_i$, such that ϕ_i contains at-most two variables is projective if and only if all the f_{ij} (as given in Definition 37) have the same value i.e. $\forall i, j \in [u] : f_{ij} = F$, for some positive real number F.*

In the next section, we will show that the conditions in Theorem 27 correspond to a special type of probability distributions. We will characterize such distributions and then investigate their properties.

6.6 Relational Block Model

In this section we introduce the Relational Block Model (RBM). We show that any projective MLN in the two variable fragment can be expressed as an RBM. And any RBM can be expressed as a projective MLN. Furthermore, we show that an RBM is a unique characterization of a projective MLN in the two variable fragment.

Definition 38. *Let n be a positive integer (the number of domain constants), u be a positive integer (the number of 1-types), b be a positive integer (the number of 2-tables), $p = (p_1, ..., p_u)$ be a probability vector on $[u] = \{1, ..., u\}$ and $W = (w_{ijl}) \in [0, 1]^{u \times u \times b}$, where $w_{ijl} = w_{\underline{ijl}}$ (w_{ijl} is the conditional probability of domain elements (c, d) realizing the l^{th} 2-table, given $i(c)$ and $j(d)$). The multi-relational graph $(\boldsymbol{x}, \boldsymbol{y})$ is drawn under $\mathrm{RBM}(n, p, W)$ if $\boldsymbol{x}$ is an n-dimensional vector with i.i.d components distributed under p and $\boldsymbol{y}$ is a random vector with its component $y_{qr} = l$, where $l \in [b]$, with a probability $w_{x_q x_r l}$ independently of all other pair of domain constants.*

Thus, the probability distribution of $(\boldsymbol{x}, \boldsymbol{y})$ is defined as follows, where $\boldsymbol{x} \in [u]^n$ and $\boldsymbol{y} \in [b]^{\binom{n}{2}}$

$$
\begin{aligned}
P(\boldsymbol{X} = \boldsymbol{x}) &:= \prod_{q=1}^{n} p_{x_q} = \prod_{i=1}^{u} p_{x_i}^{k_i} \\
P(\boldsymbol{Y} = \boldsymbol{y} | \boldsymbol{X} = \boldsymbol{x}) &:= \prod_{1 \leq q < r \leq n} w_{x_q x_r y_{qr}} \\
&= \prod_{1 \leq i \leq j \leq u} \prod_{1 \leq l \leq b} (w_{ijl})^{h_l^{ij}}
\end{aligned}
$$

In the following example, we show how RBMs can model homophily.

Example 15 (Homophily). *Let us have an FO^2 language with a unary predicate C (representing a two colors) and a binary predicate R. We wish to model a distribution on simple undirected graphs i.e. models of the formula $\phi = \forall xy.\neg R(x,x) \wedge (R(x,y) \rightarrow R(y,x))$ such that same color nodes are more likely to have an edge. Due to ϕ the 1-types with $\neg R(x,x)$ as a conjunct have a probability zero. Hence, we can assume we have only two 1-types: $1(x) = C(x) \wedge \neg R(x,x)$ and $2(x) = \neg C(x) \wedge \neg R(x,x)$ (representing two possible colors for a given node). Similarly due to ϕ, we have only two 2-tables $1(x,y) : R(x,y) \wedge R(y,x)$ and $2(x,y) : \neg R(x,y) \wedge \neg R(y,x)$ (representing existence and non existence of edges). We can now easily define homophily by following parameterization of an RBM. $p_1 = p_2 = 0.5$ i.e. any node can have two colors with equal probability. Then we can define $w_{111} = 0.9$, $w_{112} = 0.1$, $w_{221} = 0.9$, $w_{222} = 0.1$, $w_{121} = 0.1$ and $w_{122} = 0.9$.*

Theorem 28. *Every projective Markov Logic Network in the two variable fragment can be expressed as an RBM.*

Proof. The proof follows from the sufficiency proof in Lemma 10. Notice that in the proof, we derive equation (6.5) (equivalently, equation (21) in the appendix), which is exactly the expression for RBM. Hence, any projective MLN can be converted to an RBM by defining p_i and w_{ijl} as follows:

$$p_i = \frac{s_i}{\sum_i s_i} \quad w_{ijl} = \frac{t_{ijl}}{\sum_l t_{ijl}} \tag{6.8}$$

□

Theorem 29. *Every RBM can be expressed as a projective MLN in the two variable fragment.*

Proof. Given an RBM as defined in definition 38 with parameters $\{p_i, w_{ijl}\}$, let us have a projective MLN Φ such that every 1-type $i(x)$ is a formula in the MLN with a weight $\log p_i$. Φ also has a weighted formula $ijl(x,y)$ for every 2-type, such that $i \leq j$. The weight for $ijl(x,y)$ is $\log(w_{ijl})$ if $ijl(x,y) \neq ijl(y,x)$, and is $0.5\log(w_{ijl})$ if $ijl(x,y) = ijl(y,x)$. It can be seen from definition of s_i (6.2) and t_{ijl} (6.3), that for Φ, $s_i = p_i$ and $t_{ijl} = w_{ijl}$. Hence, due to (6.1), we have that:

$$P_{\Phi}^{(n)}(\omega) = \frac{1}{Z(n)} \prod_{i \in [u]} p_i^{k_i} \prod_{\substack{i,j \in [u] \\ i \leq j}} \prod_{l \in [b]} (w_{ijl})^{h_l^{ij}} \tag{6.9}$$

In the MLN Φ, $\sum_i s_i = \sum_i p_i = 1$ and $\sum_l t_{ijl} = \sum_l w_{ijl} = f_{ij} = 1$. Hence, using Proposition 15, we have that $Z(n) = 1$. Hence, completing the proof.

□

Proposition 16. *Given two RBMs with probability distribution P' and P'' and parameters $\{p'_i, w'_{ijl}\}$ and $\{p''_i, w''_{ijl}\}$. If $P' = P''$, then, $p'_i = p''_i$ and $w'_{ijl} = w''_{ijl}$.*

Proof. The proposition is a consequence of the fact that the parameter p_i is marginal probability of an arbitrary constant c realizing the i^{th} 1-type and w_{ijl} is the conditional probability of an arbitrary pair of constants (c, d) realizing the l^{th} 2-table given $i(c)$ and $j(d)$. Hence, two RBMs that disagree on the p_i and w_{ijl} cannot assign the same probability mass to marginal probability of $i(c)$ and $ijl(c, d)$ and hence, cannot be the same distribution.

□

Corollary 4 (of Proposition 16)**.** *Given two projective MLNs Φ' and Φ'' such that they have the same probability distributions $P_{\Phi'}$ and $P_{\Phi''}$, with their respective RBMs parameterized by $\{p'_i, w'_{ijl}\}$ and $\{p''_i, w''_{ijl}\}$. Then we must have that $p'_i = p''_i$ and $w'_{ijl} = w''_{ijl}$.*

Hence, RBMs are a unique representation for projective MLNs in the two variable fragment.

6.7 Previous Characterizations of Projectivity

Jaeger and Schulte [16] show that an MLN is projective if it's formulae ϕ_i satisfy the property that any two atoms appearing in ϕ_i contain exactly the same variables. Such MLNs are also known as *σ-determinate* [60]. We now show that in the two variable fragment, Theorem 27 leads to a strictly more expressive class of MLNs.

Proposition 17. *Given an MLN $\Phi = \{\phi_i, a_i\}_i$ such that any two atoms appearing in ϕ_i contain exactly the same variables or equivalently that the MLN is $\sigma-$determinate. Then:*

$$\forall i, j, i', j' \in [u], \forall l \in [b] : t_{i'j'l} = t_{ijl} \tag{6.10}$$

Proof. We first write an equivalent MLN $\Phi' = \{\alpha_q(x), a_q\} \cup \{\beta_p(x, y), b_p\}$ as presented in proof of Theorem 26. Due to the conditions provided in the proposition, all the atoms in $\beta_p(x, y)$ contain both the variables x and y. Using the definition of t_{ijl} from (6.3), and the fact that none of the $\beta_p(x, y)$ have an atom with only one variable, we have that the value of t_{ijl} depends only on the l^{th} 2-table, irrespective of the 1-types i and j. This is because, none of the first order atoms in the i^{th} and the j^{th} 1-type appear in $\beta_p(x, y)$. Hence, t_{ijl} only depends on l, giving us equation (6.10). □

Proposition 17 is a stricter condition than Theorem 27. In the following, we prove that σ-determinate MLNs cannot express all the projective MLNs in the two variable fragment.

Theorem 30. *There exists a projective MLN in the two variable fragment which cannot be expressed as a* $\sigma-$*determinate MLN.*

Proof. Let us have a σ-determinate MLN Φ, since Φ is projective, we can create it's equivalent RBM (due to Theorem 29), say P. Let $\{p_i, w_{ijl}\}$ be the parameters of P. Due to equation (6.8) and Proposition 17, we have that $w_{ijl} = w_{i'j'l}$ for all i, j, i', j'. Due to existence of a projective MLN for every RBM (from Theorem 29), we can always create an MLN Φ' for which the RBM parameters $w_{ijl} \neq w_{i'j'l}$ for some i, j, i', j'. Since, RBMs uniquely characterize the probability distributions due to MLNs (from Corollary 4), Φ' can not be expressed as an MLN such that $w_{ijl} = w_{i'j'l}$. Hence, Φ' can not be expressed as a σ-determinate MLN.

$\square$

In the following example, we provide an MLN which cannot be written as a $\sigma-$ determinate MLN.

Example 16. *Let us have a binary predicate* R. *We have only two 1-types* $R(x,x)$ *(say* $1(x)$*) and* $\neg R(x,x)$ *(say* $2(x)$*) and four 2-tables,* $R(x,y) \land R(y,x)$ *(say* $1(x,y)$*),* $R(x,y) \land \neg R(y,x)$ *(say* $2(x,y)$*),* $\neg R(x,y) \land R(y,x)$ *(say* $3(x,y)$*) and* $\neg R(x,y) \land \neg R(y,x)$ *(say* $4(x,y)$*). An MLN* Φ*, with the following 2-types as weighted formulas, cannot be expressed as a* $\sigma-$*determinate MLN:*

$$
\begin{array}{cc}
111(x,y) : \log 7 & 114(x,y) : \log 4 \\
124(x,y) : \log 64 & 221(x,y) : \log 8
\end{array}
$$

In parametric normal form, $t_{111} = \exp(2\log 7)$, $t_{114} = \exp(2\log 4)$, $t_{124} = \exp(\log 64)$ *and* $t_{221} = \exp(2\log 8)$. *All the other* t_{ijl}*, such that* $ijl(x,y)$ *is not a dual of* $111(x,y)$, $114(x,y)$, $124(x,y)$ *or* $221(x,y)$*, are equal to* $\exp(0)$ *i.e. 1. It can be verified that* $f_{ij} = 67$ *for all* $i, j \in [2]$*, hence, this MLN is projective due to Theorem 27. Using Theorem 28, we can express this distribution as an RBM, such that* $w_{111} = \frac{7^2}{67}$ *and* $w_{114} = \frac{4^2}{67}$. *If* $w_{111} \neq w_{114}$ *then necessarily* $t_{111} \neq t_{114}$ *(as* w_{ijl} *is defined as* $\frac{t_{ijl}}{f_{ij}}$ *and* f_{ij} *is the same for all* i, j *in* Φ *and in any equivalent MLN, due to Theorem 27). Due to uniqueness of RBM parameters for any set of projective MLNs expressing the same distribution (Corollary 4), we have that in all MLNs equivalent to* Φ, $t_{111} \neq t_{114}$. *Hence,* *using Proposition 17, we have that any MLN expressing the same distribution as Φ cannot be expressed as a $\sigma-$determinate MLN.*

6.8 Maximum Likelihood Learning

In a learning setting, for an MLN $\{\phi_i, a_i\}$ in the two variable fragment, we are interested in estimating the set of parameters $\boldsymbol{\theta} = \{a_i\}$ that maximize the likelihood of a training example such that the learnt MLN is projective. As analyzed in [61, 59], we will focus on the scenario where only a single possible world $\omega \in \Omega^{(n)}$ is observed. We estimate $\boldsymbol{\theta}$ by maximizing the likelihood

$$L^{(n)}(\boldsymbol{\theta}|\omega) = P_{\boldsymbol{\theta}}^{(n)}(\omega) \tag{6.11}$$

Notice that although every projective MLN can be equivalently defined as an RBM, the maximum likelihood parameter estimate for an RBM is not the same as the parameter estimate for an MLN such that it is projective.

We will now provide, the maximum likelihood estimator for an RBM.

Proposition 18. *Given a training example $\omega \in \Omega^{(n)}$, the maximum likelihood parameter estimate for an RBM is given as, $p_i = \frac{k_i}{n}$ and $w_{ijl} = \frac{h_l^{ij}}{\boldsymbol{k}(i,j)}$.*

Proposition 18 can be derived by maximizing the log likelihood due to the distribution given in Definition 38.

We will now see how maximum likelihood parameter estimate can be obtained for an MLN such that the MLN is projective.

Given an MLN $\{\phi_i, a_i\}_i$ in the two variable fragment, where $\boldsymbol{\theta} = \{a_i\}_i$ are unknown parameters to be estimated, due to Theorem 26, we can define $s_i(\boldsymbol{\theta})$ and $t_{ijl}(\boldsymbol{\theta})$, such that the likelihood is given as:

$$L(\boldsymbol{\theta}|\omega) = \frac{1}{Z(n)} \prod_{i\in[u]} s_i(\boldsymbol{\theta})^{k_i} \prod_{\substack{i,j\in[u]\\ i\leq j}} \prod_{l\in[b]} (t_{ijl}(\boldsymbol{\theta}))^{h_l^{ij}} \tag{6.12}$$

Defining $F(\boldsymbol{\theta})$ as $\sum_l t_{i'j'l}(\boldsymbol{\theta})$ for some fixed i' and j', the maximum likelihood parameter estimates such that the estimated MLN is projective, can be then obtained by solving the following optimization problem:

$$\begin{aligned} maximize_{\boldsymbol{\theta}} : \Big[& \sum_{i \in [u]} k_i \log s_i(\boldsymbol{\theta}) + \sum_{\substack{i,j \in [u] \\ i \leq j}} \sum_{l \in [b]} h_l^{ij} \log t_{ijl}(\boldsymbol{\theta}) \\ & - n \log \big(\sum_{i \in [u]} s_i(\boldsymbol{\theta}) \big) - \binom{n}{2} \log F(\boldsymbol{\theta}) \Big] \\ subject \quad to : \quad & \forall i, j \in [u] : f_{ij}(\boldsymbol{\theta}) = F(\boldsymbol{\theta}) \end{aligned} \tag{6.13}$$

Notice that due to factorization of $Z(n)$ under projectivity (see Lemma 10), $-n \log \big(\sum_{i \in [u]} s_i(\boldsymbol{\theta}) \big) - \binom{n}{2} \log F(\boldsymbol{\theta})$ represents $-\log(Z(n))$. The above optimization can be solved through any conventional optimization algorithm. It can be seen that this problem has a much lesser overhead as far as computing $\log(Z(n))$ is concerned. But the additional constraints may counter act this gain. Furthermore, in many cases it may happen that no non-zero weights exist that satisfy the constraints and in that case the problem will return zero weights for the MLN formulas.

Theorem 31. *Given a training example $\omega \in \Omega^{(n)}$, then there is no parameterization for any projective MLN in the two variable fragment that has a higher likelihood for ω than the maximum likelihood RBM for ω.*

Proof. Let L be the likelihood of ω due to the maximum likelihood RBM. Let L' be the likelihood of ω due to a projective MLN Φ, such that $L' > L$. Now, due to Theorem 28, Φ can be expressed as an RBM. Hence, we can have an RBM such that the likelihood of ω is L', but $L' > L$ which is a contradiction. Hence, we cannot have a projective MLN that gives a higher likelihood to ω than the maximum likelihood RBM. □

Theorem 31 shows us that if a data source is known to be projective (i.e. we know that marginals in the data will be independent of the domain at large) then in terms of likelihood, specially in the case of large relational datasets, we are better off in using an RBM than an expert defined MLN. This can also be argued from efficiency point of view as RBMs admit much more efficient parameter estimates.

We will now move on to the question: *are parameters learned on a domain of size n, also good for modelling domain of a different size m ?* This question is an abstraction of many real world problems, for example, learning over relational data in presence of incomplete information [62], modelling a social network from only sub-sampled populations [63], modelling progression of a disease in a population by only testing a small set of individuals [64] etc.

Jaeger and Schulte [16] formalized the afore mentioned notions in the following two criterions:

$$E_\omega[\operatorname*{argmax}_{\boldsymbol{\theta}} \log L^{(m)}(\boldsymbol{\theta}|\omega')] = \operatorname*{argmax}_{\boldsymbol{\theta}} \log L^{(n)}(\boldsymbol{\theta}|\omega) \tag{6.14}$$

$$\operatorname*{argmax}_{\boldsymbol{\theta}} E_\omega[\log L^{(m)}(\boldsymbol{\theta}|\omega')] = \operatorname*{argmax}_{\boldsymbol{\theta}} \log L^{(n)}(\boldsymbol{\theta}|\omega) \tag{6.15}$$

It is easy to see, by law of large numbers, that RBMs satisfy both these criterions. On the other hand the same can not be said about the maximum likelihood estimates for projective MLNs as described in (6.13).

6.9 Conclusion

In this work, we have characterized the class of projective MLNs in the two-variable fragment. We have also identified a special class of models, namely Relational Block Model. We show that the maximum likelihood RBM maximizes the training data likelihood w.r.t to any projective MLN in the two-variable fragment. Furthermore, RBMs admit consistent parameter learning from sub-sampled domains, potentially allowing them to scale to very large datasets, especially in situations where the test data size is not known or changes over time.

From an applications point of view, the superiority of RBMs in terms of training likelihood maximisation and consistent parameter learning can potentially make them a better choice over an expert defined MLN, especially when training set is large and the test domain size is unknown or varies over time. We plan to investigate such capabilities of RBMs and projective MLNs in future work, especially in comparison to models like Adaptive MLNs [17] and Domain Size Aware MLNs [18].

On the theoretical front, the imposed independence structure due to projectivity clearly resembles the AHK models proposed in [57]. In future works, we aim at investigating this resemblance and generalizing our work to capture complete projectivity criterion for all the MLNs.

6.10 Appendix to Chapter 6

A.1 : Lemma 10 [Sufficiency]

Lemma 10 (Sufficiency). *A Markov Logic Network in the two variable fragment is projective if all the f_{ij} have the same value i.e. $\forall i, j \in [u] : f_{ij} = F$, for some positive real number F.*

Proof. Let $\forall i, j \in [u] : f_{ij} = F$. Hence, due to Proposition 15, we have:

$$Z(n) = \sum_{\boldsymbol{k}} \binom{n}{\boldsymbol{k}} \prod_{i \in [u]} s_i^{k_i} \prod_{\substack{i,j \in [u] \\ i \leq j}} (F)^{\boldsymbol{k}(i,j)} \tag{6.16}$$

$$= \sum_{\boldsymbol{k}} \binom{n}{\boldsymbol{k}} \prod_{i \in [u]} s_i^{k_i} (F)^{\binom{n}{2}} = F^{\binom{n}{2}} \Big(\sum_{i \in [u]} s_i \Big)^n \tag{6.17}$$

Let $p_i = \left(\frac{s_i}{\sum_i s_i} \right)$ and $w_{ijl} = \left(\frac{t_{ijl}}{F} \right)$. Hence,

$$P_\Phi^{(n)}(\omega) = \frac{1}{(\sum_{i \in [u]} s_i)^n (F)^{\binom{n}{2}}} \prod_{i \in [u]} s_i^{k_i} \prod_{\substack{i,j \in [u] \\ i \leq j}} \prod_{l \in [b]} (t_{ijl})^{h_l^{ij}}$$

$$= \prod_{i \in [u]} \Big(\frac{s_i}{\sum_{i \in [u]} s_i} \Big)^{k_i} \prod_{\substack{i,j \in [u] \\ i \leq j}} \prod_{l \in [b]} (\frac{t_{ijl}}{F})^{h_l^{ij}}$$

$$= \prod_{i \in [u]} p_i^{k_i} \prod_{\substack{i,j \in [u] \\ i \leq j}} \prod_{l \in [b]} w_{ijl}^{h_l^{ij}}$$

Using the multi-relational representation, $P_\Phi^{(n)}(\omega)$ can be equivalently expressed as:

$$P_\Phi^{(n)}(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{Y} = \boldsymbol{y}) = \prod_{q \in [n]} p_{x_q} \prod_{\substack{q,r \in [n] \\ q < r}} w_{x_q x_r y_{qr}} \tag{6.18}$$

Let $(\boldsymbol{X}', \boldsymbol{Y}')$ be the random vector containing X_q and $Y_{p,q}$ with $p < q \in [m]$. Clearly, our goal is to show that

$$P_\Phi^{(n)} \downarrow [m](\boldsymbol{X}' = \boldsymbol{x}', \boldsymbol{Y}' = \boldsymbol{y}') = P_\Phi^{(m)}(\boldsymbol{X}' = \boldsymbol{x}', \boldsymbol{Y}' = \boldsymbol{y}')$$

Now, the marginal distribution over the m-worlds $(\boldsymbol{X}', \boldsymbol{Y}')$, due to $P^{(n)}_{\Phi}(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{Y} = \boldsymbol{y})$ can be expressed as:

$$
\begin{aligned}
P^{(n)}_{\Phi} \downarrow [m](\boldsymbol{X}' = \boldsymbol{x}', \boldsymbol{Y}' = \boldsymbol{y}') &= \sum_{\substack{\boldsymbol{x}_{[m]}=\boldsymbol{x}' \\ \boldsymbol{y}_{[m]}=\boldsymbol{y}'}} P^{(n)}_{\Phi}(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{Y} = \boldsymbol{y}) \\
&= \sum_{\substack{\boldsymbol{x}_{[m]}=\boldsymbol{x}' \\ \boldsymbol{y}_{[m]}=\boldsymbol{y}'}} \prod_{q\in[n]} p_{x_q} \prod_{\substack{q,r\in[n] \\ q<r}} w_{x_q x_r y_{qr}} \\
&= \prod_{q\in[m]} p_{x_q} \prod_{\substack{q,r\in[m] \\ q<r}} w_{x_q x_r y_{qr}} \times \left(\sum_{\substack{\boldsymbol{x}_{[\overline{m}]} \\ \boldsymbol{y}_{[\overline{m}]}}} \prod_{q\in[\overline{m}]} p_{x_q} \prod_{\substack{q,r\in[\overline{m}] \\ q<r}} w_{x_q x_r y_{qr}} \prod_{\substack{q\in[m] \\ r\in[\overline{m}]}} w_{x_q x_r y_{qr}} \right) \\
&= \prod_{i\in[u]} p_i^{k_i(\boldsymbol{x}')} \prod_{\substack{i,j\in[u] \\ i\leq j}} \prod_{l\in[b]} w_{ijl}^{h^{ij}_l(\boldsymbol{y}')} \\
&\times \left(\sum_{\substack{\boldsymbol{x}_{[\overline{m}]} \\ \boldsymbol{y}_{[\overline{m}]}}} \prod_{i\in[u]} p_i^{k_i(\boldsymbol{x}_{[\overline{m}]})} \prod_{\substack{i,j\in[u] \\ i\leq j}} \prod_{l\in[b]} w_{ijl}^{h^{ij}_l(\boldsymbol{y}_{[\overline{m}]})} \prod_{\substack{i,j\in[u] \\ i\leq j}} \prod_{l\in[b]} w_{ijl}^{h^{ij}_l([m]\otimes[\overline{m}])} \right)
\end{aligned}
$$

where $A \otimes B = A \times B \cup B \times A$. Notice that $\prod_{i\in[u]} p_i^{k_i(\boldsymbol{x}')} \prod_{\substack{i,j\in[u] \\ i\leq j}} \prod_{l\in[b]} w_{ijl}^{h^{ij}_l(\boldsymbol{y}')}$ is $P^{(m)}_{\Phi}(\boldsymbol{X}' = \boldsymbol{x}', \boldsymbol{Y}' = \boldsymbol{y}')$. Hence, in order to complete the proof, we will now show that for any $\boldsymbol{x}'$:

$$
\sum_{\substack{\boldsymbol{x}_{[\overline{m}]} \\ \boldsymbol{y}_{[\overline{m}]}}} \prod_{i\in[u]} p_i^{k_i(\boldsymbol{x}_{[\overline{m}]})} \prod_{\substack{i,j\in[u] \\ i\leq j}} \prod_{l\in[b]} w_{ijl}^{h^{ij}_l(\boldsymbol{y}_{[\overline{m}]})} \prod_{\substack{i,j\in[u] \\ i\leq j}} \prod_{l\in[b]} w_{ijl}^{h^{ij}_l([m]\otimes[\overline{m}])} = 1 \tag{6.19}
$$

The LHS of equation (6.19) can be written as:

$$
\sum_{\sum \boldsymbol{k}=n-m} \binom{n-m}{\boldsymbol{k}} \prod_{i\in[u]} p_i^{k_i} \prod_{\substack{i,j\in[u] \\ i\leq j}} (\sum_l w_{ijl})^{\boldsymbol{k}(i,j)} \prod_{\substack{i,j\in[u] \\ i\leq j}} (\sum_l w_{ijl})^{k_i(\boldsymbol{x}')\times k_j} \tag{6.20}
$$

By definition, for any $i, j \in [u]$, $\sum_l w_{ijl} = 1$, and $\sum_i p_i = 1$. Hence, expression (6.20) can be written as:

$$
\sum_{\sum \boldsymbol{k}=n-m} \binom{n-m}{\boldsymbol{k}} \prod_{i\in[u]} p_i^{k_i} = \left(\sum_i p_i\right)^{n-m} = 1
$$

Hence, completing the proof. □

A.2 : Lemma 11 [Necessary]

Lemma 11 (Necessary)**.** *If a Markov Logic network in the two variable fragment is projective then, all the f_{ij} have the same value i.e. $\forall i, j \in [u] : f_{ij} = F$, for some positive real number F.*

Proof. Let us have a markov logic network Φ over a domain $[n+1]$. Let $\boldsymbol{X}$ and $\boldsymbol{Y}$ be random vectors representing multi-relational graphs on the domain $[n+1]$. Let $\boldsymbol{X'}$ and $\boldsymbol{Y'}$ be random vectors representing multi-relational graphs on the domain $[n]$. Then :

$$P_{\Phi}^{(n+1)} \downarrow [n](\boldsymbol{X'} = \boldsymbol{x'}, \boldsymbol{Y'} = \boldsymbol{y'}) = \sum_{\substack{\boldsymbol{x}_{[n]}=\boldsymbol{x'} \\ \boldsymbol{y}_{[n]}=\boldsymbol{y'}}} P_{\Phi}^{(n+1)}(\boldsymbol{X} = \boldsymbol{x}, \boldsymbol{Y} = \boldsymbol{y})$$

$$= \sum_{\substack{\boldsymbol{x}_{[n]}=\boldsymbol{x'} \\ \boldsymbol{y}_{[n]}=\boldsymbol{y'}}} \frac{1}{Z(n+1)} \prod_{q \in [n+1]} s_{x_q} \prod_{\substack{q,r \in [n+1] \\ q<r}} t_{x_q x_r y_{qr}}$$

$$= \frac{1}{Z(n)} \prod_{q \in [n]} s_{x_q} \prod_{\substack{q,r \in [n] \\ q<r}} t_{x_q x_r y_{qr}} \frac{Z(n)}{Z(n+1)} \sum_{\substack{x_{n+1} \\ y_{q,n+1}}} s_{x_{n+1}} \prod_{q \in [n]} t_{x_q x_{n+1} y_{q,n+1}}$$

$$= P_{\Phi}^{(n)}(\boldsymbol{X'} = \boldsymbol{x'}, \boldsymbol{Y'} = \boldsymbol{y'}) \frac{Z(n)}{Z(n+1)} \sum_{\substack{x_{n+1} \\ y_{q,n+1}}} s_{x_{n+1}} \prod_{q \in [n]} t_{x_q x_{n+1} y_{q,n+1}}$$

Due to projectivity we have that:

$$P_{\Phi}^{(n)}(\boldsymbol{X'} = \boldsymbol{x'}, \boldsymbol{Y'} = \boldsymbol{y'}) = P_{\Phi}^{(n+1)} \downarrow [n](\boldsymbol{X'} = \boldsymbol{x'}, \boldsymbol{Y'} = \boldsymbol{y'})$$

Hence,

$$\frac{Z(n+1)}{Z(n)} = \sum_{\substack{x_{n+1} \\ y_{q,n+1}}} s_{x_{n+1}} \prod_{q \in [n]} t_{x_q x_{n+1} y_{q,n+1}}$$

which can be equivalently written as:

$$\frac{Z(n+1)}{Z(n)} = \sum_{i \in [u]} s_i \prod_{j \in [u]} \Big(\sum_{l \in [b]} t_{jil} \Big)^{k_j(\boldsymbol{x'})}$$

Now, $\sum_{l \in [b]} t_{jiv} = f_{ji} = f_{ij}$. Hence:

$$\frac{Z(n+1)}{Z(n)} = \sum_{i \in [u]} s_i \prod_{j \in [u]} f_{ij}^{k_j(\boldsymbol{x}')}$$

Hence, for any choice of the domain size m and for any choice of m-worlds $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{x}', \boldsymbol{y}')$, we have that:

$$\sum_{i \in [u]} s_i \prod_{j \in [u]} f_{ij}^{k_j(\boldsymbol{x})} = \sum_{i \in [u]} s_i \prod_{j \in [u]} f_{ij}^{k_j(\boldsymbol{x}')} \tag{6.21}$$

which implies [1] that:

$$\forall i, j, i', j' \in [u] : f_{ij} = f_{i'j'}$$

Hence, completing the proof. □

A.3 : Auxiliary Lemmas for Lemma 11

In proof of Lemma 11 we argue that, for any choice of the domain size m and for any choice of m-worlds $(\boldsymbol{x}, \boldsymbol{y})$ and $(\boldsymbol{x}', \boldsymbol{y}')$, we have that:

$$\sum_{i \in [u]} s_i \prod_{j \in [u]} f_{ij}^{k_j(\boldsymbol{x})} = \sum_{i \in [u]} s_i \prod_{j \in [u]} f_{ij}^{k_j(\boldsymbol{x}')} \tag{6.22}$$

This implies that:

$$\forall i, j, i', j' \in [u] : f_{ij} = f_{i'j'} \tag{6.23}$$

We will first infer a slightly stricter equation from (6.22). $\boldsymbol{x}$ and $\boldsymbol{x}'$ can have any 1-type cardinalities, say $\boldsymbol{k} = \langle k_1(\boldsymbol{x})...k_u(\boldsymbol{x})\rangle = \langle k_1...k_u\rangle$ and $\boldsymbol{k}' = \langle k_1(\boldsymbol{x}')...k_u(\boldsymbol{x}')\rangle = \langle k'_1...k'_u\rangle$ respectively, such that $\sum_{i \in [u]} k_i = \sum_{i \in [u]} k'_i = m$. Hence, we can conclude that, for all $\boldsymbol{k}$ and $\boldsymbol{k}'$ such that $\sum_{i \in [u]} k_i = \sum_{i \in [u]} k'_i$, we have that:

$$\sum_{i \in [u]} s_i \prod_{j \in [u]} f_{ij}^{k_j} = \sum_{i \in [u]} s_i \prod_{j \in [u]} f_{ij}^{k'_j} \tag{6.24}$$

Hence, our goal is to prove that (6.24) implies (6.23). We formally prove this statement in Lemma 13. Before proving Lemma 13, we will need to prove the following auxiliary lemma.

[1] For a rigorous proof of why this is true, see Lemma 12 and Lemma 13 in Appendix A.3

Lemma 12. *Let $(x_i)_{i=1}^m$,$(y_i)_{i=1}^m$ and $(a_i)_{i=1}^m$ be tuples of positive non-zero reals. If for all positive integers n:*

$$\sum_{i=1}^{m} a_i x_i^n = \sum_{i=1}^{m} a_i y_i^n \tag{6.25}$$

then the set of entries in $(x_i)_{i=1}^m$ and the set of entries in $(y_i)_{i=1}^m$ are the same.

Proof. Let $\{u_i\}_{i=1}^p$ and $\{v_i\}_{i=1}^q$ be the set of unique entries in $(x_i)_{i=1}^m$ and $(y_i)_{i=1}^m$ respectively. Also, without loss of generality, we may assume an ordering such that $u_1 > u_2 > ... > u_p$ and $v_1 > v_2 > ... > v_q$ and also that $q \geq p$. We can rewrite (6.25) as:

$$\forall n \in \mathbb{Z}^+ : \sum_{i=1}^{p} c_i u_i^n = \sum_{i=1}^{q} d_i v_i^n \tag{6.26}$$

As n grows the leading term on LHS is $c_1 u_1^n$ and on the RHS is $d_1 v_1^n$. Hence, it must be :

$$\forall n \in \mathbb{Z}^+ : c_1 u_1^n = d_1 v_1^n$$

Since, u_1, v_1, c_1 and d_1 are non-zero positive reals, we can conclude that $u_1 = v_1$ and $c_1 = d_1$. Hence, we may subtract $c_1 u_1^n$ from both sides in (6.26) to get :

$$\forall n \in \mathbb{Z}^+ : \sum_{i=2}^{p} c_i u_i^n = \sum_{i=2}^{q} d_i v_i^n \tag{6.27}$$

We may now repeat the aforementioned argument and infer that $u_2 = v_2$ and $c_2 = d_2$. Furthermore, repeating this argument p times, we can infer that $\{u_i\}_{i=1}^p = \{v_i\}_{i=1}^p$, leaving us with $0 = \sum_{i=q-p+1}^{p} d_i v_i^n$, which is a contradiction as d_i and v_i are positive reals. Hence, we must have that $p = q$. Hence, we have that $\{u_i\}_{i=1}^p = \{v_i\}_{i=1}^q$. Hence, completing the proof. □

Since, $f_{ij} = f_{ji}$, we can see $\{f_{ij}\}$ as a symmetric $u \times u$ matrix (f_{ij}) in $\mathbb{R}_{>0}^{u \times u}$. Hence, the statement that equation (6.24) implies equation (6.23) can be formally written as the following Lemma.

Lemma 13. *Let $S = (f_{ij}) \in \mathbb{R}_{>0}^{u \times u}$ be a symmetric matrix and let $(s_i)_{i=1}^u \in \mathbb{R}_{>0}^u$. If for all $\boldsymbol{k} = \langle k_1, ..., k_u \rangle$ and $\boldsymbol{k'} = \langle k'_1, ..., k'_u \rangle$ such that $k_i, k'_i \in \mathbb{Z}^+$ and $\sum_{i=1}^u k_i = \sum_{i=1}^u k'_i$, we have that:*

$$\sum_{i=1}^{u} s_i \prod_{j \in [u]} f_{ij}^{k_j} = \sum_{i=1}^{u} s_i \prod_{j \in [u]} f_{ij}^{k'_j} \tag{6.28}$$

then

$$\forall i, j, i', j' : f_{ij} = f_{i'j'}$$

Proof. Let $\boldsymbol{k}$ be such that $k_p = n$, let $k_i = 0$ for all $i \neq p$. Let $\boldsymbol{k'}$ be such that $k'_q = n$ and $k'_i = 0$ for all $i \neq q$. Then due to (6.28), we have that:

$$\forall n \in \mathbb{Z}^+ : \sum_{i=1}^{u} s_i(f_{ip})^n = \sum_{i=1}^{u} s_i(f_{iq})^n \tag{6.29}$$

Hence, due to Lemma 12, we have that the entries in $(f_{ip})_{i=1}^u$ and $(f_{iq})_{i=1}^u$ form the same set. A similar argument can be repeated for any pair of columns. Hence, all columns in S have the same set of entries, we denote the set of such entries as U.

Let $n = uk$ where $k \in \mathbb{Z}^+$. Let $\boldsymbol{k}$ be such that $k_i = k$ for all $i \in [u]$. Let $\boldsymbol{k'}$ be such that $k'_q = n$ and $k'_i = 0$ for all $i \neq q$. Then due to (6.28), we have that:

$$\forall k \in \mathbb{Z}^+ : \sum_{i=1}^{u} s_i \prod_{p\in[u]} f_{ip}^k = \sum_{i=1}^{u} s_i(f_{iq})^{uk}$$
$$\forall k \in \mathbb{Z}^+ : \sum_{i=1}^{u} s_i(\prod_{p\in[u]} f_{ip})^k = \sum_{i=1}^{u} s_i(f_{iq}^u)^k$$

As k grows the leading term on left-hand side and right-hand side must agree for the equality to hold. Let $c_{i'}(\prod_{p\in[u]} f_{i'p})^k$ and $d_{i''}(f_{i''q}^u)^k$ be the leading terms on RHS and LHS respectively. Hence,

$$\forall k \in \mathbb{Z}^+ : c_{i'}(\prod_{p\in[u]} f_{i'p})^k = d_{i''}(f_{i''q}^u)^k \tag{6.30}$$

Using Lemma 12, we have that $\prod_{p\in[u]} f_{i'p} = f_{i''q}^u$. Now, $f_{i''q}$ has to be equal to the maximum term in U, say m. Also, $\prod_{p\in[u]} f_{i'p}$ is a product of all the terms in the p^{th} matrix column of S. Since, each matrix column has the same set of terms U, we have that $\prod_{p\in[u]} f_{i'p} \leq m^u$. But due to (6.30), we have that, $\prod_{p\in[u]} f_{i'p} = m^u$, which is possible iff:

$$\forall i, j, i', j' : f_{ij} = f_{i'j'}$$

□

References

[1] Zoé Christoff and Jens Ulrik Hansen. A logic for diffusion in social networks. *Journal of Applied Logic*, 13(1):48–77, 2015.

[2] Nikita A Sakhanenko and David J Galas. Markov logic networks in the analysis of genetic data. *J. Comput. Biol.*, 17(11):1491–1508, November 2010.

[3] Jue Wang and Pedro Domingos. Hybrid markov logic networks. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, page 1106–1111. AAAI Press, 2008.

[4] Eduardo Massad, Neli R.S. Ortega, Lacio C. Barros, and Cludio J. Struchiner. *Fuzzy Logic in Action: Applications in Epidemiology and Beyond.* Springer Publishing Company, Incorporated, 1st edition, 2009.

[5] Mathias Verbeke, Vincent Van Asch, Roser Morante, Paolo Frasconi, Walter Daelemans, and Luc De Raedt. A statistical relational learning approach to identifying evidence based medicine categories. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, page 579–589, USA, 2012. Association for Computational Linguistics.

[6] Julia Hoxha and Achim Rettinger. First-order probabilistic model for hybrid recommendations. In *Proceedings of the 2013 12th International Conference on Machine Learning and Applications - Volume 02*, ICMLA '13, page 133–139, USA, 2013. IEEE Computer Society.

[7] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014.

[8] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 3 edition, 2010.

[9] Lise Getoor and Ben Taskar. *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning).* The MIT Press, 2007.

[10] Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2016.

[11] Dan Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1):273–302, 1996.

[12] L.G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

[13] Haim Gaifman. On local and non-local properties. In J. Stern, editor, *Proceedings of the Herbrand Symposium*, volume 107 of *Studies in Logic and the Foundations of Mathematics*, pages 105–135. Elsevier, 1982.

[14] Leonid Libkin. *Elements of Finite Model Theory*. Springer, August 2004.

[15] Cosma Rohilla Shalizi and Alessandro Rinaldo. Consistency under sampling of exponential random graph models. *Annals of statistics*, 41 2:508–535, 2013.

[16] Manfred Jaeger and Oliver Schulte. Inference, learning, and population size: Projectivity for SRL models. *CoRR*, abs/1807.00564, 2018.

[17] Dominik Jain, Andreas Barthels, and Michael Beetz. Adaptive markov logic networks: Learning statistical relational models with dynamic parameters. In Helder Coelho, Rudi Studer, and Michael J. Wooldridge, editors, *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 937–942. IOS Press, 2010.

[18] Happy Mittal, Ayush Bhardwaj, Vibhav Gogate, and Parag Singla. Domain-size aware markov logic networks. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 3216–3224. PMLR, 2019.

[19] Ondrej Kuzelka. Weighted first-order model counting in the two-variable fragment with counting quantifiers. *J. Artif. Intell. Res.*, 70:1281–1307, 2021.

[20] Paul W. Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137, 1983.

[21] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[22] Sagar Malhotra and Luciano Serafini. Weighted model counting in fo2 with cardinality constraints and counting quantifiers: A closed form formula. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(5):5817–5824, Jun. 2022.

[23] Sagar Malhotra and Luciano Serafini. A combinatorial approach to weighted model counting in the two-variable fragment with cardinality constraints. In *AIxIA 2021 – Advances in Artificial Intelligence: 20th International Conference of the Italian Association for Artificial Intelligence, Virtual Event, December 1–3, 2021, Revised Selected Papers*, page 137–152, Berlin, Heidelberg, 2021. Springer-Verlag.

[24] Sagar Malhotra and Luciano Serafini. Weighted first order model counting with directed acyclic graph axioms, 2023.

[25] Sagar Malhotra and Luciano Serafini. On projectivity in markov logic networks. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part V*, page 223–238, Berlin, Heidelberg, 2023. Springer-Verlag.

[26] Timothy Hinrichs and Michael Genesereth. Herbrand logic. *LG-2006-02, Stanford Reports*, 2009.

[27] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[28] Paul Beame, Guy Van den Broeck, Eric Gribkoff, and Dan Suciu. Symmetric weighted first-order model counting. In Tova Milo and Diego Calvanese, editors, *Proceedings of the 34th ACM Symposium on Principles of Database Systems, PODS 2015, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 313–328. ACM, 2015.

[29] Guy Van den Broeck. On the completeness of first-order knowledge compilation for lifted probabilistic inference. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, volume 24, pages 1386–1394. Curran Associates, Inc., 2011.

[30] R. A. Fisher. On the Mathematical Foundations of Theoretical Statistics. *Philosophical Transactions of the Royal Society of London Series A*, 222:309–368, January 1922.

[31] Guy Van den Broeck, Nima Taghipour, Wannes Meert, Jesse Davis, and Luc De Raedt. Lifted probabilistic inference by first-order knowledge compilation. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 2178–2185. AAAI Press/International Joint Conferences on Artificial Intelligence, IJCAI/AAAI, 2011.

[32] Ondrej Kuzelka and Vyacheslav Kungurtsev. Lifted weight learning of markov logic networks revisited, 2019.

[33] Seyed Mehran Kazemi, Angelika Kimmig, Guy Van den Broeck, and David Poole. New liftable classes for first-order probabilistic inference. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3117–3125, 2016.

[34] Seyed Mehran Kazemi, Angelika Kimmig, Guy Van den Broeck, and David Poole. Domain recursion for lifted inference with existential quantifiers. *CoRR*, abs/1707.07763:1386–1394, 2017.

[35] Antti Kuusisto and Carsten Lutz. Weighted model counting beyond two-variable logic. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 619–628. ACM, 2018.

[36] Guy Van den Broeck, Wannes Meert, and Adnan Darwiche. Skolemization for weighted first-order model counting. In Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014, Vienna, Austria, July 20-24, 2014*. AAAI Press, 2014.

[37] Ondrej Kuzelka. Complex markov logic networks: Expressivity and liftability. In Ryan P. Adams and Vibhav Gogate, editors, *Proceedings of the Thirty-Sixth Conference on Uncertainty in Artificial Intelligence, UAI 2020, virtual online, August 3-6, 2020*, volume 124 of *Proceedings of Machine Learning Research*, pages 729–738. PMLR, AUAI Press, 2020.

[38] Vibhav Gogate and Pedro M. Domingos. Probabilistic theorem proving. In Fábio Gagliardi Cozman and Avi Pfeffer, editors, *UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011*, pages 256–265. AUAI Press, 2011.

[39] Manfred Jaeger and Guy Van den Broeck. Liftability of probabilistic inference: Upper and lower bounds, 2012-08-18.

[40] MANFRED JAEGER. Lower complexity bounds for lifted inference. *Theory and Practice of Logic Programming*, 15(2):246–263, may 2014.

[41] Antti Kuusisto. On the uniform one-dimensional fragment. In Maurizio Lenzerini and Rafael Peñaloza, editors, *Proceedings of the 29th International Workshop on Description Logics, Cape Town, South Africa, April 22-25, 2016*, volume 1577 of *CEUR Workshop Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, CEUR-WS.org, 2016.

[42] D. Scott. A decision method for validity of sentences in two variables. *Journal of Symbolic Logic*, 27(377):377, 1962.

[43] Herbert S Wilf. *Generatingfunctionology*. CRC press, 2005.

[44] Erich Gradel, Martin Otto, and Eric Rosen. Two-variable logic with counting is decidable. In *Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science*, pages 306–317. IEEE, 1997.

[45] James R. Clough, Jamie Gollings, Tamar V. Loach, and Tim S. Evans. Transitive reduction of citation networks. *Journal of Complex Networks*, 3(2):189–203, 09 2014.

[46] Bertran Steinsky. Enumeration of labelled chain graphs and labelled essential directed acyclic graphs. *Discrete Mathematics*, 270(1):267–278, 2003.

[47] Robert Ganian, Thekla Hamm, and Topi Talvitie. An efficient algorithm for counting markov equivalent dags. *Artificial Intelligence*, 304:103648, 2022.

[48] J'anos T. T'oth and Ondřej Kuzelka. Lifted inference with linear order axiom. *ArXiv*, abs/2211.01164, 2022.

[49] Timothy van Bremen and Ondřej Kuželka. Lifted Inference with Tree Axioms. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*, pages 599–608, 11 2021.

[50] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.

[51] Patrick Doreian. On the connectivity of social networks. *The Journal of Mathematical Sociology*, 3(2):245–258, 1974.

[52] Sabyasachee Mishra, Timothy F Welch, and Manoj K Jha. Performance indicators for public transit connectivity in multi-modal transportation networks. *Transportation Research Part A: Policy and Practice*, 46(7):1066–1085, 2012.

[53] Roger Paredes, L. Dueñas-Osorio, Kuldeep Meel, and Moshe Vardi. Principled network reliability approximation: A counting-based approach. *Reliability Engineering and System Safety*, 191, 04 2019.

[54] Ettore Bompard, Di Wu, and Fei Xue. Structural vulnerability of power systems: A topological approach. *Lancet*, 81:1334–1340, 07 2011.

[55] David Poole, David Buchman, Seyed Mehran Kazemi, Kristian Kersting, and Sriraam Natarajan. Population size extrapolation in relational probabilistic modelling. In Umberto Straccia and Andrea Calì, editors, *Scalable Uncertainty Management - 8th International Conference, SUM 2014, Oxford, UK, September 15-17, 2014. Proceedings*, volume 8720 of *Lecture Notes in Computer Science*, pages 292–305. Springer, 2014.

[56] TOM A. B. SNIJDERS. Conditional marginalization for exponential random graph models. *The Journal of Mathematical Sociology*, 34(4):239–252, 2010.

[57] Manfred Jaeger and Oliver Schulte. A complete characterization of projectivity for statistical relational models. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4283–4290. ijcai.org, 2020.

[58] Felix Q. Weitkämper. An asymptotic analysis of probabilistic logic programming, with implications for expressing projective families of distributions. *Theory Pract. Log. Program.*, 21(6):802–817, 2021.

[59] Ondrej Kuzelka, Vyacheslav Kungurtsev, and Yuyi Wang. Lifted weight learning of markov logic networks (revisited one more time). In Manfred Jaeger and Thomas Dyhre Nielsen, editors, *International Conference on Probabilistic Graphical Models, PGM 2020, 23-25 September 2020, Aalborg, Hotel Comwell Rebild Bakker, Skørping, Denmark*, volume 138 of *Proceedings of Machine Learning Research*, pages 269–280. PMLR, 2020.

[60] Parag Singla and Pedro M. Domingos. Markov logic in infinite domains. pages 368–375, 2007.

[61] Rongjing Xiang and Jennifer Neville. Relational learning with one network: An asymptotic analysis. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pages 779–788. JMLR.org, 2011.

[62] Gueorgi Kossinets. Effects of missing data in social networks. *Soc. Networks*, 28(3):247–268, 2006.

[63] Mark S. Handcock and Krista J. Gile. Modeling social networks from sampled data. *The Annals of Applied Statistics*, 4(1):5 – 25, 2010.

[64] Sundara Rajan Srinivasavaradhan, Pavlos Nikolopoulos, Christina Fragouli, and Suhas Diggavi. Dynamic group testing to control and monitor disease progression in a population, 2021.