



Bridging Weighted First Order Model Counting and Graph Polynomials

Qipeng Kuang  

The University of Hong Kong, China

Ondřej Kuželka  

Czech Technical University in Prague, Czech Republic

Yuanhong Wang  

Jilin University, Changchun City, China

Yuyi Wang  

CRRC Zhuzhou Institute, China

Abstract

The Weighted First-Order Model Counting Problem (WFOMC) asks to compute the weighted sum of models of a given first-order logic sentence over a given domain. It can be solved in time polynomial in the domain size for sentences from the two-variable fragment with counting quantifiers, known as C^2 . This polynomial-time complexity is known to be retained when extending C^2 by one of the following axioms: linear order axiom, tree axiom, forest axiom, directed acyclic graph axiom or connectedness axiom. An interesting question remains as to which other axioms can be added to the first-order sentences in this way. We provide a new perspective on this problem by associating WFOMC with graph polynomials. Using WFOMC, we define Weak Connectedness Polynomial and Strong Connectedness Polynomials for first-order logic sentences. It turns out that these polynomials have the following interesting properties. First, they can be computed in polynomial time in the domain size for sentences from C^2 . Second, we can use them to solve WFOMC with all of the existing axioms known to be tractable as well as with new ones such as bipartiteness, strong connectedness, having k connected components, etc. Third, the well-known Tutte polynomial can be recovered as a special case of the Weak Connectedness Polynomial, and the Strict and Non-Strict Directed Chromatic Polynomials can be recovered from the Strong Connectedness Polynomials.

2012 ACM Subject Classification Theory of computation → Finite Model Theory

Keywords and phrases Weighted First-Order Model Counting, Axiom, Enumerative Combinatorics, Tutte Polynomial

Digital Object Identifier 10.4230/LIPIcs.CSL.2026.7

Related Version *Full Version*: <https://arxiv.org/abs/2407.11877>

Supplementary Material *Software (Source Code)*: <https://github.com/121719p/Polynomials-for-WFOMC> [12], archived at [swh:1:dir:e82fa8e26998cfb9e25e723b2c1182f17a82276e](https://www.swh.io/dir/e82fa8e26998cfb9e25e723b2c1182f17a82276e)

Funding *Ondřej Kuželka*: Czech Science Foundation project 24-11820S (“Automatic Combinatorialist”).

Yuanhong Wang: National Natural Science Foundation of China (No. 62506141).

Yuyi Wang: partially supported by the Natural Science Foundation of Hunan Province, China (Grant No. 2024JJ5128).

1 Introduction

Given a function-free first-order logic sentence Ψ and $n \in \mathbb{N}$, the *first-order model counting* (FOMC) problem asks to compute the number of models of Ψ over the domain $\{1, \dots, n\}$. FOMC can be used, among others, to solve various combinatorics problems on labeled



© Qipeng Kuang, Ondřej Kuželka, Yuanhong Wang, and Yuyi Wang; licensed under Creative Commons License CC-BY 4.0

34th EACSL Annual Conference on Computer Science Logic (CSL 2026).

Editors: Stefano Guerrini and Barbara König; Article No. 7; pp. 7:1–7:23

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

structures. For instance, the FOMC of the sentence $\Psi = \forall x \exists y (E(x, y) \vee E(y, x)) \wedge \forall x \neg E(x, x)$ corresponds to the number of n -vertex directed graphs with no isolated vertices and no loops. Similarly, the FOMC of $\Phi = \forall x \neg E(x, x) \wedge \forall x \forall y (E(x, y) \rightarrow E(y, x)) \wedge \forall x \exists_{=k} y E(x, y)$ equals the number of k -regular graphs on n vertices where $\exists_{=k} y$ is a counting quantifier restricting the number of y satisfying $E(x, y) = \top$ to be exactly k for each x .

The weighted variant of the FOMC problem, known as the *weighted first-order model counting* (WFOMC) problem, additionally expects a specification of positive and negative weights for predicates in the language, which are then used to assign weights to models. In WFOMC, the task is to compute the sum of the weights of the given sentence's models.¹ WFOMC has applications in Statistical Relational Learning (SRL) [5]. For instance, Markov Logic Networks (MLNs) [21], one of the most popular SRL formalisms, model relational data by a distribution over possible worlds, where the computation of the partition function (used in probabilistic inference in MLN) is reducible to WFOMC [29].

A key issue regarding WFOMC is its computational complexity. Generally, there is no hope of devising algorithms for WFOMC that would scale polynomially with the size of the first-order logic sentences under reasonable assumptions [2]. This is true even for very simple fragments such as the one-variable fragment FO^1 , which contains sentences with at most one logical variable. Therefore, most of the focus in this area has been on identifying fragments of first-order logic for which WFOMC can be computed in time polynomial in the size of the domain (but not necessarily in the size of the sentence). The term coined for this kind of tractability by [29] is *domain liftability* – it is an analog to data complexity [32] from database theory.

The first positive results regarding the tractability of WFOMC came from the two seminal papers [29, 31], which together established domain liftability of the two-variable fragment of first-order logic FO^2 . This was quickly complemented by a hardness result [2], showing that WFOMC for the three-variable fragment FO^3 is not domain-liftable (under plausible complexity-theoretic assumptions). However, this does not mean that the frontiers of tractability cannot be pushed beyond FO^2 . For instance, the two-variable fragment with counting quantifiers C^2 [8] (e.g., the above sentence of a k -regular graph) with cardinality constraints (i.e., the restrictions to the number of true ground atoms for relations), which is strictly more expressive than FO^2 , was shown to be domain-liftable in [14]. Other works [30, 33, 34] proved domain-liftability when attaching ground unary literals (as part of the input along with the domain size) to sentences from these fragments.

One strategy for discovering new tractable fragments is to add extra axioms to a fragment known to be tractable, which may or may not be finitely expressible in first-order logic. This approach was first explored in [13] where it was shown that adding a single functionality axiom to FO^2 results in a domain-liftable fragment.² Subsequent works extended domain liftability to the fragments $\text{C}^2 + \textit{Tree}$ [28] and $\text{C}^2 + \textit{LinearOrder}$ [24], which are obtained by adding to C^2 an axiom specifying that a distinguished relation R should correspond to an undirected tree or to a linear order, respectively. Recently, the work [17] added several new tractable classes: $\text{C}^2 + \textit{DAG}$, $\text{C}^2 + \textit{Connected}$, and $\text{C}^2 + \textit{Forest}$, which allow restricting a distinguished binary relation to be a directed acyclic graph or to be a connected undirected graph or to be a forest. Except for the linear-order and the functionality axiom, these axioms are not finitely expressible in first-order logic.

¹ Surprisingly, WFOMC is also essential for the unweighted FOMC problem since, for example, Skolemization for FOMC [31], which is used to eliminate existential quantifiers, relies on the usage of (negative) weights.

² This fragment is contained in C^2 , which can encode an arbitrary number of functionality constraints.

■ **Table 1** Axioms that we proved the domain-liftability of. Note that the first 4 axioms require the distinguished relations to be symmetric and irreflexive.

| Axiom | Description | Lifted by | First lifted by |
|------------------|--|------------|---------------------------------|
| $connected_k(R)$ | $G(R)$ has k connected components | Theorem 15 | new ($connected_1(R)$ by [17]) |
| $bipartite(R)$ | $G(R)$ is a bipartite graph | Theorem 16 | new |
| $tree(R)$ | $G(R)$ is a tree | Theorem 19 | [28] |
| $forest(R)$ | $G(R)$ is a forest | Theorem 20 | [17] |
| $SC(R)$ | $G(R)$ is strongly connected | Theorem 31 | new |
| $AC(R)$ | $G(R)$ is acyclic | Theorem 33 | [17] |
| $DT(R, Root)$ | $G(R)$ is a directed tree such that the root interprets the unary predicate $Root$ as true | Theorem 34 | [28] |
| $DF(R)$ | $G(R)$ is a directed graph such that each weakly connected component is a directed tree | Theorem 34 | [17] |
| $LO(R)$ | $G(R)$ represents a linear order | Theorem 35 | [24] |

To add the axioms discussed above to C^2 while preserving domain-liftability, existing works usually rely on directly translating specialized techniques from enumerative combinatorics. For instance, Kirchhoff’s matrix-tree theorem was used for the tree axiom in [28]. Similarly, ideas used in existing combinatorial proofs for counting connected graphs and acyclic graphs on n -vertices were adapted to the WFOMC setting in [17] to implement the connectedness and acyclicity axioms. This raises a natural question:

Can we define a general framework for adding axioms to C^2 that allows us to prove domain-liftability of new axioms and to efficiently compute the WFOMC of sentences with these axioms?

1.1 Our Contribution

In this paper, we take a route inspired by *graph polynomials* such as the Tutte polynomial [25] and directed chromatic polynomials [22]. Graph polynomials are algebraic expressions that encode various combinatorial properties of graphs, allowing us to capture information such as connectivity, colorability, and spanning substructures. In this work, we consider C^2 possibly extended by cardinality constraints and ground unary literals (which we will call C^2 with cardinality constraints for simplicity). We introduce similar polynomials for first-order logic sentences, capturing certain combinatorial properties of their models, and show that they can be computed efficiently for C^2 with cardinality constraints, which requires the introduction of new algorithmic techniques for computation of WFOMC. Using the new polynomials for WFOMC, we can tackle the problem of efficiently computing WFOMC of C^2 sentences with cardinality constraints and one of a number of non-trivial *axioms*, special constraints on the graph interpreting a distinguished binary relation R (denoted by $G(R)$). The main axioms for which we prove domain-liftability are listed in Table 1, involving several new axioms as well as all of the old ones.

► **Theorem 1.** *The fragment of C^2 with cardinality constraints and any single axiom in Table 1 is domain-liftable.*

■ **Table 2** New axioms by combining existing axioms. Note that the first 2 axioms require the distinguished relations to be symmetric and irreflexive.

| Axiom | Description | Combined axioms |
|------------------|---|--------------------------------|
| $bipartite_k(R)$ | $G(R)$ is a bipartite graph with k connected components | $connected_k \wedge bipartite$ |
| $forest_k(R)$ | $G(R)$ is a forest with k trees | $connected_k \wedge forest$ |
| $AC_k(R)$ | $G(R)$ is a DAG with k weakly connected components | $connected_k \wedge AC$ |
| $BiAC(R)$ | $G(R)$ is a DAG such that the underlying undirected graph is bipartite | $bipartite \wedge AC$ |
| $polytree(R)$ | $G(R)$ is a polytree, i.e., a DAG whose underlying undirected graph is a tree | $tree \wedge AC$ |
| $polyforest(R)$ | $G(R)$ is a polyforest, i.e., a DAG whose underlying undirected graph is a forest | $forest \wedge AC$ |

Moreover, we can show domain-liftability of combinations of axioms. Table 2 offers a comprehensive list of axioms obtained by combining the existing axioms, all of which are new axioms that have not been shown to be domain-liftable before.³

► **Theorem 2.** *The fragment of C^2 with cardinality constraints and any single axiom in Table 2 is domain-liftable.*

Besides using the idea of graph polynomials to solve several counting tasks efficiently, conversely, the polynomials for WFOMC bring new results on graph polynomials. We show that the Tutte polynomial [25], the strict and non-strict directed chromatic polynomials [22] can be recovered from our polynomials for WFOMC. This allows us to show that these graph polynomials can be computed in time polynomial in the number of vertices for any graph that can be encoded by a set of ground unary literals, cardinality constraints and a fixed C^2 sentence (the parameters of cardinality constraints and the number of ground unary literals are allowed to depend on the number of vertices). This result answers in the positive way the question in [28] whether calculating the Tutte polynomial of a block-structured graph (see its definition in Example 8) can be done in time polynomial in the number of vertices.

1.2 Related Work

We build on several lines of research. First, we build on the stream of results from lifted inference literature which established domain-liftability of several natural fragments of first-order logic [18, 3, 29, 6, 31, 2, 10, 14]. These works started with the seminal results showing domain-liftability of FO^2 [29, 31] and continued by showing domain-liftability of larger fragments such as S^2FO^2 and S^2RU [10], U_1 [13] or C^2 [14]. Second, we build on extensions of these works that added axioms not expressible in these fragments or even in first-order logic such as the tree axiom [28], forest axiom [17], linear order axiom [24], acyclicity axiom [17] and connectedness axiom [17]. In this work, we generalize all of these results and add new axioms to this list, showing their domain-liftability. Finally, our work builds on tools from enumerative combinatorics [23] and on graph polynomials [26]. The most well-known among

³ This would be difficult to achieve using existing techniques based on individual treatment of each axiom, as done in the literature.

graph polynomials is the Tutte polynomial [25] but there exist also extensions of it [1], with which we do not work directly in this paper. Computing the Tutte polynomial is known to be intractable in general [9], however, in this work we show that it can be computed in polynomial time for graphs that can be encoded by a fixed sentence from the first-order fragment C^2 , possibly with cardinality constraints and ground unary literals, both of which may depend on the number of vertices of the graph (unlike the C^2 sentence which needs to stay fixed).

2 Preliminaries

We introduce our notations and main technical concepts, and provide a brief overview of existing algorithms in this section. A brief introduction to graph polynomials is in Appendix B.

Throughout this paper, we denote the set of natural numbers from 1 to n as $[n]$. For a polynomial $f(x)$, the notation $[x^i]f$ represents the coefficient of the monomial x^i .

All vectors used in this paper are non-negative integer vectors. For a vector \mathbf{c} of length L , $|\mathbf{c}|$ denotes the sum of its elements, i.e., $\sum_{i=1}^L c_i$. When considering two vectors \mathbf{a} and \mathbf{b} of the same length L , the operation $\mathbf{a} + \mathbf{b}$ denotes the vector $(a_1 + b_1, a_2 + b_2, \dots, a_L + b_L)$, and the subtraction is defined similarly. For a vector \mathbf{a} and a non-negative integer n , $\binom{n}{\mathbf{a}}$ refers to $\binom{n}{a_1} \binom{n-a_1}{a_2} \dots \binom{n-a_1-\dots-a_{L-1}}{a_L} = \frac{n!}{a_1! a_2! \dots a_L!}$.

2.1 First-Order Logic

We consider the function-free (we allow constants but not any functions with arity greater than zero), finite domain fragment of first-order logic. An *atom* of arity k takes the form $P(x_1, \dots, x_k)$ where P/k is from a vocabulary of *predicates* (also called *relations*), and x_1, \dots, x_k are logical variables from a vocabulary of variables or constants from a vocabulary of constants.⁴ A *literal* is an atom or its negation. A *formula* is a literal or formed by connecting one or more formulas together using negation, conjunction, or disjunction. A formula may be optionally surrounded by one or more quantifiers of the form $\forall x$ or $\exists x$, where x is a logical variable. A logical variable in a formula is said to be *free* if it is not bound by any quantifier. A formula with no free variables is called a *sentence*.

A first-order logic formula in which all variables are substituted by constants in the domain is called *ground*. A *possible world* ω interprets each predicate in a sentence over a finite domain, represented by a set of ground literals. Sometimes we only care about the interpretation of a single predicate R in a possible world ω , which is denoted by ω_R . The satisfaction relation \models is defined in the usual way: $\omega \models \alpha$ means that the formula α is true in ω . The possible world ω is a *model* of a sentence Ψ if $\omega \models \Psi$. We denote the set of all models of a sentence Ψ over the domain $\{1, 2, \dots, n\}$ by $\mathcal{M}_{\Psi, n}$.

2.2 Weighted First Order Model Counting

The *weighted first-order model counting problem* (WFOMC) takes the input consisting of a first-order sentence Ψ , a domain size $n \in \mathbb{N}$, and a pair of weighting functions (w, \bar{w}) that both map all predicates in Ψ to real weights. Given a set \mathcal{L} of ground literals whose predicates are in Ψ , the weight of \mathcal{L} is defined as $W(\mathcal{L}, w, \bar{w}) := \prod_{l \in \mathcal{L}_T} w(\text{pred}(l)) \cdot \prod_{l \in \mathcal{L}_F} \bar{w}(\text{pred}(l))$, where \mathcal{L}_T (resp. \mathcal{L}_F) denotes the set of positive (resp. negative) ground literals in \mathcal{L} , and $\text{pred}(l)$ maps a literal l to its corresponding predicate name.

⁴ We restrict $k \geq 1$ for simplicity but $k \geq 0$ is tractable for this work as well.

► **Definition 3** (Weighted First Order Model Counting). *The WFOMC of a first-order sentence Ψ over a finite domain of size n under weighting functions (w, \bar{w}) is*

$$\text{WFOMC}(\Psi, n, w, \bar{w}) := \sum_{\mu \in \mathcal{M}_{\Psi, n}} W(\mu, w, \bar{w}).$$

Since these weightings are defined on the predicate level, all positive ground literals of the same predicate get the same weights, and so do all negative ground literals of the same predicate. For this reason, the notion of WFOMC defined here is also referred to as *symmetric WFOMC* [2].

► **Example 4.** Consider $\Psi = \forall x \exists y R(x, y)$, and let $w(R) = 1, \bar{w}(R) = 1$. Then $\text{WFOMC}(\Psi, n, w, \bar{w}) = (2^n - 1)^n$ because for each domain element $i \in [n]$, there are $2^n - 1$ ways to choose the truth assignment of $R(i, 1), R(i, 2), \dots, R(i, n)$ that make $\exists y R(i, y)$ satisfied, and there are n domain elements.

2.3 Data Complexity of WFOMC

We consider the *data complexity* of WFOMC: the complexity of computing $\text{WFOMC}(\Psi, n, w, \bar{w})$ when fixing the input sentence Ψ and weighting (w, \bar{w}) , and treating the domain size n as an input encoded in unary.

► **Definition 5** (Domain-liftability [29]). *A sentence, or class of sentences, that enjoys polynomial-time data complexity, i.e., the complexity is polynomial in the domain size, is said to be domain-liftable.*

Building on [29, 31], in [2, Appendix C], it was shown that any sentence with at most two logical variables (FO^2) is domain-liftable by devising a polynomial-time algorithm for computing $\text{WFOMC}(\Psi, n, w, \bar{w})$ for any FO^2 sentence Ψ and any weighting (w, \bar{w}) . The description of this algorithm is in Appendix A. This result also serves as the basis of a WFOMC algorithm for C^2 (the FO^2 sentence with *counting quantifiers*) together with *cardinality constraints* and *ground unary literals*.

► **Definition 6** (Counting quantifiers). *The counting quantifier $\exists_{=k}$ restricts that the number of assignments of the quantified variable satisfying the subsequent formula is exactly k . Counting quantifiers $\exists_{\leq k}$ and $\exists_{\geq k}$ are defined similarly.*

Counting quantifiers allow us to express concepts such as “each vertex in the undirected graph has exactly k edges”, i.e., k -regular graphs, by the sentence $\forall x \neg E(x, x) \wedge \forall x \forall y (E(x, y) \rightarrow E(y, x)) \wedge \forall x \exists_{=k} y E(x, y)$.

► **Definition 7** (Cardinality constraints). *The cardinality constraint is an expression of the form $|P| \bowtie k$, where P is a predicate and \bowtie is a comparison operator $\{<, \leq, =, \geq, >\}$. These constraints are imposed on the number of distinct positive ground literals of P in a model.*

For example, $|Eq| = n$ means that there are exactly n positive ground literals of Eq in a model, and the sentence $(\forall x Eq(x, x)) \wedge |Eq| = n$ along with a domain of size n encodes an identity relation Eq . Note that we allow k in Definition 7 to depend on the domain size (such as in the definition of Eq here where we used $|Eq| = n$).

Finally, we show the usage of ground unary literals defined in Section 2.1. For example, the sentence $\forall x \exists y (R(x, y) \rightarrow S(y)) \wedge S(1) \wedge \neg S(2)$ contains ground unary literals $S(1) \wedge \neg S(2)$ indicating that any interpretation of this sentence should satisfy that $S(1)$ is true and $S(2)$ is false. Another example is encoding a block-structured graph.

► **Example 8.** A *block-structured graph* is an undirected graph $G = (V, E)$ where vertices are partitioned into a constant number of blocks, and for every two blocks V_i, V_j , either $\forall x \in V_i \forall y \in V_j, (x, y) \in E$ or $\forall x \in V_i \forall y \in V_j, (x, y) \notin E$. We can encode a block-structured graph in the following way: we introduce a fresh unary predicate $Block_i$ for each block V_i , and let

$$\begin{aligned} \Psi_B = & \bigwedge_{V_i, V_j \text{ are connected}} \forall x \forall y (Block_i(x) \wedge Block_j(y) \rightarrow E(x, y)) \\ & \wedge \bigwedge_{V_i, V_j \text{ are not connected}} \forall x \forall y (Block_i(x) \wedge Block_j(y) \rightarrow \neg E(x, y)) \\ & \wedge \bigwedge_{e \in V} \left(Block_{v(e)}(e) \wedge \bigwedge_{j \neq v(e)} \neg Block_j(e) \right), \end{aligned} \tag{1}$$

where $v(e)$ indicates the block that vertex e belongs to, and $Block_*(e)$ and $\neg Block_*(e)$ are ground unary literals for vertex e indicating whether e is in the specified block or not.

► **Remark 9.** We need to define the data complexity more carefully when the input sentence of WFOMC involves cardinality constraints and ground unary literals. In such instances, the C^2 sentence (including the parameters in counting quantifiers) in the input remains fixed, while cardinality constraints and ground unary literals are considered variable inputs encoded in binary.

The presence of counting quantifiers, cardinality constraints, and ground unary literals does not affect the domain-liftability of FO^2 . Using the techniques in [14] and [34, Appendix A], WFOMC of C^2 with cardinality constraints and ground unary literals can be reduced to WFOMC of FO^2 with symbolic weights⁵, and then solved by the algorithm for FO^2 in a straightforward manner.

► **Remark 10.** The techniques mentioned above apply to the methods in this paper. Therefore, in the rest of this paper we only present the algorithms for FO^2 and state the results in terms of C^2 with cardinality constraints for simplicity.

2.4 Axioms

While C^2 captures numerous graph structures, it falls short in expressing certain important graph properties, such as being a tree, a forest, and strongly connected. In fact, these structures cannot be finitely axiomatized by first-order logic [16].

Given a binary predicate R , an interpretation for R can be regarded as a directed graph where the domain is the vertex set and R is the edge set, which we call $G(R)$. If a possible world ω is specified, the interpretation of R in ω is a specific graph, denoted by $G(\omega_R)$.

An axiom is a special constraint on the interpretation of a binary predicate R that $G(R)$ should be a certain combinatorial structure. We often write an axiom as $axiom(R)$ where $axiom$ is an identifier of the axiom. For example, in what follows, we will discuss the *tree axiom* $tree(R)$, which requires $G(R)$ to be a tree, and the *forest axiom* $forest(R)$, which requires $G(R)$ to be a forest. For better illustration, we treat axioms as atomic formulas in a sentence, and use $\Psi \wedge axiom(R)$ to denote the sentence Ψ with the axiom $axiom(R)$ conjoined. For instance, $tree(E) \wedge \forall x (Leaf(x) \leftrightarrow \exists_{=1} y E(x, y))$ is a sentence that expresses the concept of trees with a leaf predicate $Leaf/1$.

⁵ The weighting functions w, \bar{w} map predicates to symbolic variables. For more details, see [14, Section 3.1].

3 Weak Connectedness Polynomial

In this section, we introduce our first polynomial, called *Weak Connectedness Polynomial* (WCP). The WCP is a univariate polynomial defined on a first-order logic sentence and a distinguished binary relation. Similarly to graph polynomials, the WCP can also capture various combinatorial properties of the models of the sentence w.r.t. the binary relation. We show that WCP can be computed in polynomial time in the domain size for sentences from the fragment C^2 with cardinality constraints. This property allows us to attach axioms of non-trivial combinatorial structures to the first-order logic sentence as well as to derive the Tutte polynomial for certain graphs. All the omitted proofs can be found in Appendix C.

3.1 Definition

We define WCP as a univariate polynomial whose point evaluation at every positive integer u is equal to a weighted first-order model count of a suitable first-order logic sentence parameterized by u .

► **Definition 11** (Weak Connectedness Polynomial). *Let Ψ be a first-order logic sentence possibly with cardinality constraints and ground unary literals, w, \bar{w} be two weighting functions, and R be a distinguished binary relation. The n -th WCP $f_n(u)$ of Ψ for the relation R is the univariate polynomial which satisfies:*

$$f_n(u; \Psi, w, \bar{w}, R) = \text{WFOMC}(\Psi_{R,u}, n, w, \bar{w})$$

for all positive integers u . Here, the sentence $\Psi_{R,u}$ is defined as:

$$\begin{aligned} \Psi_{R,u} = & \Psi \wedge \bigwedge_{i=1}^u \forall x \forall y (A_i(x) \wedge (R(x,y) \vee R(y,x)) \rightarrow A_i(y)) \\ & \wedge \bigwedge_{i=2}^u \forall x (A_i(x) \rightarrow A_{i-1}(x)), \end{aligned} \quad (2)$$

where A_1, \dots, A_u are fresh unary predicates and $w(A_i) = \bar{w}(A_i) = 1$.

Intuitively, by introducing u new predicates A_1, \dots, A_u , we make each *weakly* connected component in $G(R)$ contribute a factor of $u + 1$ to WFOMC. In the rest of the paper, we use $cc(\mu_R)$ to denote the number of weakly connected components in $G(\mu_R)$. The most important property of WCP is that it captures $cc(\mu_R)$, which is stated formally in the following proposition.

► **Proposition 12.** *For the sentence Ψ , the weighting functions w and \bar{w} and the binary relation R , it holds that*

$$f_n(u; \Psi, w, \bar{w}, R) = \sum_{\mu \in \mathcal{M}_{\Psi, n}} W(\mu, w, \bar{w}) \cdot (u + 1)^{cc(\mu_R)}.$$

Proof. By $\bigwedge_{i=2}^u \forall x (A_i(x) \rightarrow A_{i-1}(x))$, for each element e in the domain there must be an $i' \in [0, u]$ such that $A_1(e) = \dots = A_{i'}(e) = \top$ and $A_{i'+1}(e) = \dots = A_u(e) = \perp$. Meanwhile, elements whose corresponding vertices form a weakly connected component in $G(\mu_R)$ must have the same i' due to the first row of Equation (2).

Therefore, for any model μ of Ψ , each weakly connected component in $G(\mu_R)$ has $u + 1$ choices of i' regardless of the interpretations of other predicates. That is, μ replicates $u + 1$ times in $\Psi_{R,u}$, and the weight of each of these replicas is $W(\mu, w, \bar{w})$ due to $w(A_i) = \bar{w}(A_i) = 1$. Therefore, the WFOMC of $\Psi_{R,u}$ is $\sum_{\mu \in \mathcal{M}_{\Psi, \mu}} W(\mu, w, \bar{w}) \cdot (u + 1)^{cc(\mu_R)}$, which completes the proof. ◀

► **Proposition 13.** *The n -th WCP $f_n(u; \Psi, w, \bar{w}, R)$ always exists, and it is a unique univariate polynomial of degree at most n .*

Proof. The existence and the upper bound of the degree follow immediately from Proposition 12 since $cc(\mu_R) \leq n$. ◀

3.2 Fast Calculation

Before applying the polynomials in any context, we need to be able to compute them (or at least evaluate them at some points) efficiently. By Proposition 13, $n + 1$ evaluations are sufficient to interpolate the n -th WCP (e.g., by Lagrange Interpolation). Therefore, our task is to evaluate the WCP at $u = 0, 1, \dots, n$.

Note that by definition, evaluating the polynomial at a specific positive integer is a WFOMC query. However, such WFOMC is not readily solvable by existing algorithms (e.g., [2, 27]) even for the FO^2 fragment. In general, the runtime of these algorithms is exponential in the length of the sentence, while the length of the sentence $\Psi_{R,u}$ in the WFOMC query can be as large as the domain size n , hence these algorithms would not lead to polynomial runtime in the domain size. Luckily, by exploiting the structure of $\Psi_{R,u}$ carefully, the algorithm computing WFOMC for FO^2 can be adapted into a dynamic programming (DP) style algorithm that achieves an efficient calculation of the polynomials for C^2 sentences with cardinality constraints, which is elaborated in detail in Appendix C.

► **Proposition 14.** *Computing WCP for a given C^2 sentence with cardinality constraints, a domain of size n , a distinguished binary relation, and weighting functions can be completed in polynomial time in n .*

3.3 Applications

In this section, we show how WCP can be used for computing WFOMC of C^2 sentences with cardinality constraints and several combinatorial axioms in polynomial time, i.e. we show that the fragments together with each of these axioms are domain-liftable. We also show that WCP allows us to efficiently compute the Tutte polynomial of any graph that can be encoded by a set of ground unary literals, cardinality constraints and a fixed C^2 sentence, e.g., block-structured graphs.

Most of the axioms in this section require some binary relation R to be symmetric and irreflexive. In this case, we append $\forall x \forall y (R(x, y) \rightarrow R(y, x)) \wedge \forall x \neg R(x, x)$ to the input sentence Ψ , and interpret the graph $G(R)$ as a simple undirected graph. We denote the new sentence by Ψ_R .

3.3.1 Axioms: Existing, Generalized, and New

We first show how WFOMC with several existing and new combinatorial axioms can be computed using WCP.

k -connected-component axiom. As in Proposition 12, WCP associates the weight of each model with a term indicating the number of connected components in $G(R)$, therefore we can restrict the number of connected components of $G(R)$.

► **Theorem 15.** *The fragment of C^2 with cardinality constraints and a single k -connected-component axiom is domain-liftable. More concretely, for any C^2 sentence Ψ with cardinality constraints, a domain of size n , weighting functions w, \bar{w} and a distinguished binary predicate R , it holds*

$$\text{WFOMC}(\Psi \wedge \text{connected}_k(R), n, w, \bar{w}) = [u^k] f_n(u - 1; \Psi_R, w, \bar{w}, R).$$

7:10 Bridging Weighted First Order Model Counting and Graph Polynomials

Proof. Consider a C^2 sentence Ψ with cardinality constraints, a domain of size n , the weighting functions w, \bar{w} , and a distinguished binary predicate R . By Proposition 12, we have that $\text{WFOMC}(\Psi \wedge \text{connected}_k(R), n, w, \bar{w})$ is equal to $[u^k]f_n(u-1; \Psi_R, w, \bar{w}, R)$. ◀

Bipartite axiom. The *bipartite* axiom to the binary predicate R , denoted by $\text{bipartite}(R)$, requires R to be symmetric and $G(R)$ to be a bipartite graph⁶.

► **Theorem 16.** *The fragment of C^2 with cardinality constraints and a single bipartite axiom is domain-liftable.*

Tree and forest axioms. The *tree* axiom [28] to the binary predicate R (denoted by $\text{tree}(R)$) requires R to be symmetric and $G(R)$ to be a tree. The *forest* axiom [17] to the binary predicate R (denoted by $\text{forest}(R)$) requires R to be symmetric and $G(R)$ to be a forest. To handle these two axioms, we extend the ability of WCP to classify the models of a sentence by both the number of connected components in $G(R)$ and the number of edges in $G(R)$ by introducing a new variable to the weight of R .

► **Definition 17.** *With the notations in Definition 11, the extended WCP of Ψ is defined as a bivariate polynomial such that for all positive integers u and all real numbers v ,*

$$\hat{f}_n(u, v; \Psi, w, \bar{w}, R) = \text{WFOMC}(\Psi_{R,u}, n, w_{R,v}, \bar{w}),$$

where $w_{R,v}(R) = w(R) \cdot v$, and $w_{R,v}(P) = w(P)$ for any other predicate P .

► **Corollary 18** (Corollary from Propositions 12 and 14). *For the sentence Ψ containing the symmetric binary relation R , a domain of size n , and the weighting functions w, \bar{w} , we have*

$$\hat{f}_n(u, v; \Psi, w, \bar{w}, R) = \sum_{\mu \in \mathcal{M}_{\Psi, n}} W(\mu, w, \bar{w}) \cdot (u+1)^{cc(\mu_R)} v^{2e(\mu_R)},$$

where $cc(\mu_R)$ is the number of connected components of $G(\mu_R)$ and $e(\mu_R)$ is the number of undirected edges⁷ of $G(\mu_R)$. Furthermore, the extended WCP of Ψ always exists, and can be computed in time polynomial in n using the same techniques as in Proposition 14.

With the above corollary, we can show that the tree and forest axioms are domain-liftable.

► **Theorem 19.** *The fragment of C^2 with cardinality constraints and a single tree axiom is domain-liftable. Specifically,*

$$\text{WFOMC}(\Psi \wedge \text{tree}(R), n, w, \bar{w}) = [uv^{2(n-1)}] \hat{f}_n(u-1, v; \Psi_R, w, \bar{w}, R).$$

► **Theorem 20.** *The fragment of C^2 with cardinality constraints and a single forest axiom is domain-liftable. Specifically,*

$$\text{WFOMC}(\Psi \wedge \text{forest}(R), n, w, \bar{w}) = \sum_{i \in [n]} [u^i v^{2(n-i)}] \hat{f}_n(u-1, v; \Psi_R, w, \bar{w}, R).$$

► **Remark 21.** The tree axiom and the forest axiom have already been shown to be domain-liftable in [28] and [17] respectively. Nevertheless, our method provides a more flexible command of the graph structures. For example, it is easy to add further constraints on the number of trees in the forest by our method (we will discuss this in Section 5).

⁶ Note that bipartite graphs are not finitely axiomatizable in first-order logic by the compactness theorem.

⁷ Here, as R is symmetric, $G(\mu_R)$ is interpreted as an undirected graph, and the number of undirected edges is equal to half the cardinality of R in μ . It is where the number 2 in the exponent comes from.

The strong expressive power of C^2 fragment with cardinality constraints and the axioms above allows us to represent many interesting combinatorial problems as well as Statistical Relational Learning (SRL) problems.

► **Example 22.** In a social network, one may want to model the compactness in terms of a symmetric relation *friends* by the number of connected components in $G(\textit{friends})$. In this case, a real number d might be used to represent the degree of compactness: the larger d means that the network is more compact, i.e., the number of connected components in $G(\textit{friends})$ is smaller. This can be achieved by defining the weight of a possible world ω as $W(\omega, w, \bar{w}) \cdot \exp(-d \cdot cc(\omega_{\textit{friends}}))$. Then the reduced WFOMC of the inference problem is equal to $f_n(\exp(-d) - 1; \Psi, w, \bar{w}, \textit{friends})$, where Ψ and (w, \bar{w}) are the sentence and weighting functions modeling other aspects of the social network.

3.3.2 From WCP to Tutte Polynomial

We can further use the extended WCP in Definition 17 to obtain the well-known Tutte polynomial ([25], also see Appendix B for the formal definition) of undirected graphs that can be encoded by a set of ground unary literals, cardinality constraints and a fixed C^2 sentence. Note that the size of the C^2 sentence should be fixed to a constant independent of the graph size (i.e., the number of vertices) while the parameters for cardinality constraints and the size of ground unary literals are unbounded. The weighting functions are all set to $\mathbb{1}$ (i.e., a function always returns 1) in the rest of this section.

► **Theorem 23.** *Let S be an infinite set of undirected graphs. Suppose that there is a fixed C^2 sentence Ψ with a distinguished symmetric binary relation E such that for each graph $G \in S$ of size n , there is a C^2 sentence Ψ_G obtained by conjuncting Ψ with cardinality constraints and ground unary literals, and G is isomorphic to $G(\mu_E)$ for each model $\mu \in \mathcal{M}_{\Psi_G, n}$.⁸ Then the Tutte polynomial of any $G \in S$ of size n can be computed as follows in time polynomial in n :*

$$T(x, y) = \frac{\hat{f}_n((x-1)(y-1) - 1, \sqrt{y-1}; \Psi_T(G), \mathbb{1}, \mathbb{1}, R)}{(x-1)^{cc(\mu_E)}(y-1)^n \text{WFOMC}(\Psi_G, n, \mathbb{1}, \mathbb{1})},$$

where $\Psi_T(G) = \Psi_G \wedge \forall x \forall y ((R(x, y) \rightarrow E(x, y)) \wedge (R(x, y) \rightarrow R(y, x)))$.

Proof. Denote the edges of $G(\mu_E)$ by \mathcal{E} . By Corollary 18, we have

$$\begin{aligned} & \hat{f}_n((x-1)(y-1) - 1, \sqrt{y-1}; \Psi_T(G), \mathbb{1}, \mathbb{1}, R) \\ &= \sum_{\mu' \in \mathcal{M}_{\Psi_T(G), n}} (x-1)^{cc(\mu'_R)}(y-1)^{cc(\mu'_R)+e(\mu'_R)} \\ &= \text{WFOMC}(\Psi_G, n, \mathbb{1}, \mathbb{1}) \cdot \sum_{A \subseteq \mathcal{E}} (x-1)^{cc(A)}(y-1)^{cc(A)+|A|} \\ &= \text{WFOMC}(\Psi_G, n, \mathbb{1}, \mathbb{1}) \cdot (x-1)^{cc(\mu_E)}(y-1)^n \cdot T_G(x, y), \end{aligned}$$

where $T_G(x, y)$ is the Tutte polynomial of G defined in Appendix B.

⁸ For instance, [11] gave the characterization of such graphs when the sentence is restricted to be in C^2 . Note that in their work the counting parameters and the size of the sentence are unbounded while in our case they should be constant to apply the fast calculation of WCP.

7:12 Bridging Weighted First Order Model Counting and Graph Polynomials

By Corollary 18, computing $f_n((x-1)(y-1) - 1, \sqrt{y-1}; \Psi_T(G), \mathbb{1}, \mathbb{1}, R)$ is in polynomial time in n . Since the cardinality of S is infinite, the size of Ψ is independent of the size of G , thus $\text{WFOMC}(\Psi_G, n, \mathbb{1}, \mathbb{1})$ can be computed in time polynomial in n by the algorithm described in Appendix A as well. Therefore, we can obtain $T(x, y)$ in time polynomial in n . \blacktriangleleft

► **Example 24.** We can get the Tutte polynomial for a simple undirected complete graph of any size n . Let S be the set of all simple undirected complete graphs and define

$$\Psi = \forall x \neg E(x, x), \quad \Psi_G = \Psi \wedge (|E| = n(n-1))$$

for each $G \in S$ of size n .

Then Ψ_G has a unique model μ over the domain of size n such that $G(\mu_E)$ is identical to G . Therefore,

$$\frac{\hat{f}_n((x-1)(y-1) - 1, \sqrt{y-1}; \Psi_T(G), \mathbb{1}, \mathbb{1}, R)}{(x-1)(y-1)^n}$$

is the Tutte polynomial of the complete graph of size n , which can be computed in polynomial time in n by Theorem 23.

► **Example 25.** Consider the Tutte polynomial of a block-structured graph defined in Example 8. We first fix the number k of blocks and the connection among blocks. Let S be the set of block-structured graphs of n vertices such that each graph has k blocks and the fixed connection. Define similarly to Equation (1) the following sentences:

$$\begin{aligned} \Psi &= \bigwedge_{V_i, V_j \text{ are connected}} (Block_i(x) \wedge Block_j(y) \rightarrow E(x, y)) \\ &\quad \wedge \bigwedge_{V_i, V_j \text{ are not connected}} (Block_i(x) \wedge Block_j(y) \rightarrow \neg E(x, y)), \\ \Psi_G &= \Psi \wedge \bigwedge_{e \in V} \left(Block_{v(e)}(e) \wedge \bigwedge_{j \neq v(e)} \neg Block_j(e) \right) \end{aligned}$$

for each $G \in S$ of size n . Then, the Tutte polynomial of G is

$$\frac{\hat{f}_n((x-1)(y-1) - 1, \sqrt{y-1}; \Psi_T(G), \mathbb{1}, \mathbb{1}, R)}{(x-1)^{cc(G)}(y-1)^n}.$$

By Theorem 23, the Tutte polynomial of G can be computed in polynomial time in n . Applying the same proof to any constant k and any possible connection among blocks, we conclude that the Tutte polynomial of any block-structured graph can be computed in polynomial time in the size of the graph.

The above result answers positively the question in [28] whether calculating the Tutte polynomial of a block-structured graph is hard. However, we remark that this result cannot be extended to computing the Tutte polynomial or evaluating any point for general graphs since representing a general graph by a first-order sentence involves a list of ground binary literals, making the WFOMC task $\#P$ -hard [30]. Therefore we are consistent with the hardness of computing the Tutte polynomial of general graphs [9], while showing tractability for block-structured ones (with a constant number of blocks).

4 Strong Connectedness Polynomials

The polynomial WCP is insensitive to the orientation of the edges of $G(R)$, where R is the distinguished relation. Hence, with WCP we can only define axioms that do not depend on the orientation of the edges of the graph $G(R)$. In this section, we define two new polynomials, *Strict Strong Connectedness Polynomial* (SSCP) and *Non-strict Strong Connectedness Polynomial* (NSCP), that use the orientation of the edges as well. Both polynomials are collectively referred to as *Strong Connectedness Polynomials* (SCPs). We show that these new polynomials can be used to enforce several axioms, some of which have not been known to be domain-liftable before. All the omitted proofs can be found in the full version.

4.1 Definition

The definitions of SCPs are similar to WCP except that we need another sequence of unary predicates to characterize the *strong* connectedness in $G(R)$.

► **Definition 26** (Strong Connectedness Polynomials). *Let Ψ be a first-order logic sentence possibly with cardinality constraints and ground unary literals, w and \bar{w} be two weighting functions and R be a distinguished binary relation. The n -th Strict Strong Connectedness Polynomial (SSCP) $g_n(u, v)$ of Ψ for the relation R is the bivariate polynomial which satisfies:*

$$g_n(u, v; \Psi, w, \bar{w}, R) = \text{WFOMC}(\Psi_{R,u,v}, n, w, \bar{w})$$

for all positive integers u, v , where $\Psi_{R,u,v}$ is defined as:

$$\begin{aligned} \Psi_{R,u,v} = & \Psi \wedge \bigwedge_{i=1}^u \forall x \forall y (A_i(x) \wedge (R(x, y) \vee R(y, x)) \rightarrow A_i(y)) \wedge \bigwedge_{i=2}^u \forall x (A_i(x) \rightarrow A_{i-1}(x)) \\ & \wedge \bigwedge_{i=1}^{v-1} \forall x \forall y (B_i(x) \wedge R(x, y) \rightarrow B_{i+1}(y)) \wedge \forall x \forall y (R(x, y) \rightarrow \neg B_v(x) \wedge B_1(y)) \\ & \wedge \bigwedge_{i=2}^v \forall x (B_i(x) \rightarrow B_{i-1}(x)) \end{aligned}$$

where A_i 's and B_i 's are fresh unary predicates and $w(A_i) = \bar{w}(A_i) = w(B_i) = \bar{w}(B_i) = 1$.

The n -th Non-strict Strong Connectedness Polynomial (NSCP) $\bar{g}_n(u, v)$ of Ψ for the relation R is the bivariate polynomial which satisfies:

$$\bar{g}_n(u, v; \Psi, w, \bar{w}, R) = \text{WFOMC}(\bar{\Psi}_{R,u,v}, n, w, \bar{w})$$

for all positive integers u, v , where $\bar{\Psi}_{R,u,v}$ is defined as:

$$\begin{aligned} \bar{\Psi}_{R,u,v} = & \Psi \wedge \bigwedge_{i=1}^u \forall x \forall y (A_i(x) \wedge (R(x, y) \vee R(y, x)) \rightarrow A_i(y)) \wedge \bigwedge_{i=2}^u \forall x (A_i(x) \rightarrow A_{i-1}(x)) \\ & \wedge \bigwedge_{i=1}^v \forall x \forall y (B_i(x) \wedge R(x, y) \rightarrow B_i(y)) \wedge \bigwedge_{i=2}^v \forall x (B_i(x) \rightarrow B_{i-1}(x)) \end{aligned}$$

where A_i 's and B_i 's are fresh unary predicates and $w(A_i) = \bar{w}(A_i) = w(B_i) = \bar{w}(B_i) = 1$.

The predicates A_1, \dots, A_u treat R symmetrically and are defined similarly as those in WCP. As discussed in Proposition 12, we can imagine them as capturing the information about weakly connected components of $G(R)$. The predicates B_1, \dots, B_v capture the directionality of edges in $G(R)$, enabling further directed axioms on R . We give an interpretation of both SSCP and NSCP by the following proposition and present more properties and applications in Section 4.3.

► **Proposition 27.** *For the sentence Ψ , the weighting functions w and \bar{w} and the binary relation R , it holds that*

$$g_n(u, v; \Psi, w, \bar{w}, R) = \sum_{\mu \in \mathcal{M}_{\Psi, n}} W(\mu, w, \bar{w}) \cdot (u+1)^{cc(\mu_R)} \cdot \chi_{G(\mu_R)}(v+1),$$

$$\bar{g}_n(u, v; \Psi, w, \bar{w}, R) = \sum_{\mu \in \mathcal{M}_{\Psi, n}} W(\mu, w, \bar{w}) \cdot (u+1)^{cc(\mu_R)} \cdot \bar{\chi}_{G(\mu_R)}(v+1),$$

where $cc(\mu_R)$ is the number of weakly connected components of $G(\mu_R)$, and $\chi_{G(\mu_R)}$ and $\bar{\chi}_{G(\mu_R)}$ are the strict and non-strict directed chromatic polynomials ([22], also see Appendix B for the formal definitions) of $G(\mu_R)$ respectively.

► **Proposition 28.** *$g_n(u, v; \Psi, w, \bar{w}, R)$ and $\bar{g}_n(u, v; \Psi, w, \bar{w}, R)$ always exist, and they are unique bivariate polynomials in which both variables have a degree at most n .*

4.2 Fast Calculation

Similarly to WCP, we can efficiently compute both SCPs for C^2 sentences with cardinality constraints.

► **Proposition 29.** *Computing NSCP for a given C^2 sentence with cardinality constraints, a domain of size n , a distinguished binary relation, and weighting functions can be completed in polynomial time in n .*

► **Proposition 30.** *Computing SSCP for a given C^2 sentence with cardinality constraints, a domain of size n , a distinguished binary relation, and weighting functions can be completed in polynomial time in n .*

4.3 Applications

Now, we show how SCP can be used for axioms that are associated with the orientation of the edges in $G(R)$, and for computing the directed chromatic polynomial of certain graphs.

Strong connectedness axiom. The *strong connectedness* axiom to the binary predicate R , denoted by $SC(R)$, requires $G(R)$ to be a strongly connected directed graph.

► **Theorem 31.** *The fragment of C^2 with cardinality constraints and a single strong connectedness axiom is domain-liftable. More concretely, for any C^2 sentence Ψ with cardinality constraints, a domain of size n , weighting functions w, \bar{w} and a distinguished binary predicate R , it holds*

$$\text{WFOMC}(\Psi \wedge SC(R), n, w, \bar{w}) = -[u] \bar{g}_n(u-1, -2; \Psi, w, \bar{w}, R).$$

► **Example 32.** A tournament is an orientation of an undirected complete graph. We can further require $G(R)$ to be strongly connected by adding the *strongly connected tournament* axiom to the binary predicate R , denoted by $SCT(R)$. For any C^2 sentence Ψ with cardinality constraints, a domain of size n , weighting functions w, \bar{w} and a distinguished binary predicate R , the sentence

$$\begin{aligned} \Psi_{TN} = & \Psi \wedge \forall x (\neg R(x, x) \wedge Eq(x, x)) \wedge (|Eq| = n) \\ & \wedge \forall x \forall y (\neg Eq(x, y) \rightarrow (R(x, y) \vee R(y, x)) \wedge (\neg R(x, y) \vee \neg R(y, x))) \end{aligned} \quad (3)$$

restricts $G(R)$ to be a tournament. Then by Theorem 31, it holds that

$$\text{WFOMC}(\Psi \wedge SCT(R), n, w, \bar{w}) = -[u] \bar{g}_n(u-1, -2; \Psi_{TN}, w, \bar{w}, R).$$

Therefore, the fragment of C^2 with cardinality constraints and a single strongly connected tournament axiom is domain-liftable.

Acyclicity axiom. The *acyclicity* axiom to the binary predicate R , denoted by $AC(R)$, requires that $G(R)$ does not have any directed cycle, or equivalently, $G(R)$ is a directed acyclic graph (DAG).

► **Theorem 33.** *The fragment of C^2 with cardinality constraints and a single acyclicity axiom is domain-liftable. Specifically,*

$$\text{WFOMC}(\Psi \wedge AC(R), n, w, \bar{w}) = (-1)^n \cdot g_n(0, -2; \Psi, w, \bar{w}, R).$$

Directed tree and forest axioms. The *directed tree* axiom to the binary predicate R and the unary predicate $Root$, denoted by $DT(R, Root)$, requires that $G(R)$ is a DAG with a single source, and the single source interprets $Root$ as true. The *directed forest* axiom to the binary predicate R , denoted by $DF(R)$, requires that $G(R)$ is a DAG such that every weakly connected component is a directed tree. In other words, $G(R)$ is a DAG such that every vertex has at most one incoming edge.

► **Theorem 34.** *The fragment of C^2 with cardinality constraints and a single directed tree axiom or a single directed forest axiom is domain-liftable.*

Linear order axiom. The *linear order* axiom to the binary predicate R , denoted by $LO(R)$, requires that R represents a linear order. That is to say, $G(R)$ is an acyclic tournament with a self-loop for each vertex.

► **Theorem 35.** *The fragment of C^2 with cardinality constraints and a single linear order axiom is domain-liftable.*

► **Remark 36.** Though some of the axioms have been proven to be domain-liftable such as the acyclicity axiom in [17] and the linear order axiom in [24], by our approach we can further generalize these axioms to obtain more features such as restricting the number of weakly connected components of $G(R)$ of an acyclic graph. For example, if we require $G(R)$ to be the union of k disjoint directed acyclic graphs, we can obtain the WFOMC by $(-1)^n \cdot [u^k] g_n(u-1, -2; \Psi, w, \bar{w}, R)$.

From SSCP and NSCP to Directed Chromatic Polynomials. With the same idea of obtaining the Tutte polynomial for graphs encoded by a set of ground unary literals, cardinality constraints and a fixed C^2 sentence, we can recover the strict and non-strict directed chromatic polynomials ([22], also see Appendix B for the formal definitions) from SSCP and NSCP respectively of such sentences encoding directed graphs.

► **Corollary 37** (Corollary from Theorem 23). *Let S be an infinite set of directed graphs. Suppose that there is a fixed C^2 sentence Ψ with a distinguished binary relation E such that for each $D \in S$ of size n , there is a C^2 sentence Ψ_D obtained by conjuncting Ψ with cardinality constraints and ground unary literals, and D is isomorphic to $G(\mu_E)$ for each model $\mu \in \mathcal{M}_{\Psi_D, n}$. Then the strict and non-strict directed chromatic polynomials of any $D \in S$ of size n can be computed as follows in time polynomial in n :*

$$\chi_D(x) = \frac{g_n(0, x; \Psi_D, \mathbb{1}, \mathbb{1}, E)}{\text{WFOMC}(\Psi_D, n, \mathbb{1}, \mathbb{1})}, \quad \bar{\chi}_D(x) = \frac{\bar{g}_n(0, x; \Psi_D, \mathbb{1}, \mathbb{1}, E)}{\text{WFOMC}(\Psi_D, n, \mathbb{1}, \mathbb{1})},$$

where $\mathbb{1}$ is the weighting function that always returns 1.

5 Combinations of Axioms

An important advantage of the proposed approach over the prior works is that it can easily extend to new axioms by combining existing axioms. Adopting the methodology of existing research [28, 24, 17], one would typically delve into the literature on polytree enumeration [20] and attempt to adapt the techniques (if they exist) to the WFOMC problem. However, even if such techniques are available, integrating them into current WFOMC algorithms is not a trivial task. In contrast, our approach leverages a single polynomial to encapsulate the structural characteristics of the models.

► **Theorem 38.** *The fragment of C^2 with cardinality constraints and a single axiom in the following list is domain-liftable.*

- $\text{bipartite}_k(R) = \text{connected}_k(R) \wedge \text{bipartite}(R)$, requiring $G(R)$ to be a bipartite graph with k connected components;
- $\text{forest}_k(R) = \text{connected}_k(R) \wedge \text{forest}(R)$, requiring $G(R)$ to be a forest with k trees;
- $\text{AC}_k(R) = \text{connected}_k(R) \wedge \text{AC}(R)$, requiring $G(R)$ to be a DAG with k weakly connected components;
- $\text{BiAC}(R) = \text{bipartite}(R) \wedge \text{AC}(R)$, requiring $G(R)$ to be a DAG whose underlying undirected graph is bipartite;
- $\text{polytree}(R) = \text{tree}(R) \wedge \text{AC}(R)$, requiring $G(R)$ to be a polytree, i.e., a DAG whose underlying undirected graph is a tree;
- $\text{polyforest}(R) = \text{forest}(R) \wedge \text{AC}(R)$, requiring $G(R)$ to be a polyforest, i.e., a DAG whose underlying undirected graph is a forest.

Note that in the combination of axioms involving $\text{AC}(R)$, we do not require R to be symmetric.

References

- 1 Jordan Awan and Olivier Bernardi. Tutte polynomials for directed graphs. *Journal of Combinatorial Theory, Series B*, 140:192–247, 2020. doi:10.1016/j.jctb.2019.05.006.
- 2 Paul Beame, Guy Van den Broeck, Eric Gribkoff, and Dan Suciu. Symmetric weighted first-order model counting. In *PODS*, pages 313–328. ACM, 2015. doi:10.1145/2745754.2745760.

- 3 Rodrigo de Salvo Braz, Eyal Amir, and Dan Roth. Lifted first-order probabilistic inference. In *IJCAI*, pages 1319–1325. Professional Book Center, 2005. URL: <http://ijcai.org/Proceedings/05/Papers/1548.pdf>.
- 4 Reinhard Diestel, Alexander Schrijver, and Paul Seymour. Graph theory. *Oberwolfach Reports*, 4(2):887–944, 2008.
- 5 Lise Getoor and Ben Taskar. *Introduction to statistical relational learning*, volume 1. MIT press Cambridge, 2007.
- 6 Vibhav Gogate and Pedro M. Domingos. Probabilistic theorem proving. In *UAI*, pages 256–265. AUA Press, 2011. URL: https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=2263&proceeding_id=27.
- 7 Erich Grädel, Phokion G. Kolaitis, and Moshe Y. Vardi. On the decision problem for two-variable first-order logic. *Bull. Symb. Log.*, 3(1):53–69, 1997. doi:10.2307/421196.
- 8 Erich Grädel, Martin Otto, and Eric Rosen. Two-variable logic with counting is decidable. In *LICS*, pages 306–317. IEEE Computer Society, 1997. doi:10.1109/LICS.1997.614957.
- 9 François Jaeger, Dirk L Vertigan, and Dominic JA Welsh. On the computational complexity of the jones and tutte polynomials. *Mathematical Proceedings of the Cambridge Philosophical Society*, 108(1):35–53, 1990.
- 10 Seyed Mehran Kazemi, Angelika Kimmig, Guy Van den Broeck, and David Poole. New liftable classes for first-order probabilistic inference. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, pages 3117–3125, 2016. URL: <https://proceedings.neurips.cc/paper/2016/hash/c88d8d0a6097754525e02c2246d8d27f-Abstract.html>.
- 11 Sandra Kiefer, Pascal Schweitzer, and Erkal Selman. Graphs identified by logics with counting. *ACM Trans. Comput. Log.*, 23(1):1:1–1:31, 2022. doi:10.1145/3417515.
- 12 Qipeng Kuang. 121719p/Polynomials-for-WFOMC. Software, swHId: swH:1:dir:e82fa8e26998cfb9e25e723b2c1182f17a82276e (visited on 2026-02-02). URL: <https://github.com/121719p/Polynomials-for-WFOMC>, doi:10.4230/artifacts.25483.
- 13 Antti Kuusisto and Carsten Lutz. Weighted model counting beyond two-variable logic. In *LICS*, pages 619–628. ACM, 2018. doi:10.1145/3209108.3209168.
- 14 Ondrej Kuželka. Weighted first-order model counting in the two-variable fragment with counting quantifiers. *J. Artif. Intell. Res.*, 70:1281–1307, 2021. doi:10.1613/jair.1.12320.
- 15 Michel Las Vergnas. Convexity in oriented matroids. *J. Comb. Theory, Ser. B*, 29(2):231–243, 1980. doi:10.1016/0095-8956(80)90082-9.
- 16 Leonid Libkin. *Elements of finite model theory*, volume 41. Springer, 2004. doi:10.1007/978-3-662-07003-1.
- 17 Sagar Malhotra, Davide Bizzaro, and Luciano Serafini. Lifted inference beyond first-order logic. *Artif. Intell.*, 342:104310, 2025. doi:10.1016/j.artint.2025.104310.
- 18 David Poole. First-order probabilistic inference. In *IJCAI*, pages 985–991. Morgan Kaufmann, 2003. URL: <http://ijcai.org/Proceedings/03/Papers/142.pdf>.
- 19 Ronald C Read. An introduction to chromatic polynomials. *Journal of combinatorial theory*, 4(1):52–71, 1968.
- 20 George Rebane and Judea Pearl. The recovery of causal poly-trees from statistical data. In *UAI*, pages 175–182. Elsevier, 1987. URL: https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1813&proceeding_id=1003.
- 21 Matthew Richardson and Pedro M. Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, 2006. doi:10.1007/s10994-006-5833-1.
- 22 Richard P Stanley. A chromatic-like polynomial for ordered sets. In *Proc. Second Chapel Hill Conf. on Combinatorial Mathematics and its Applications (Univ. North Carolina, Chapel Hill, NC, 1970)*, Univ. North Carolina, Chapel Hill, NC, pages 421–427, 1970.
- 23 Richard P Stanley. *What is enumerative combinatorics?* Springer, 1986.
- 24 Jan Tóth and Ondřej Kuželka. Lifted inference with linear order axiom. In *AAAI*, pages 12295–12304. AAAI Press, 2023. doi:10.1609/aaai.v37i10.26449.

- 25 William Thomas Tutte. A contribution to the theory of chromatic polynomials. *Canadian journal of mathematics*, 6:80–91, 1954.
- 26 WT Tutte. Graph-polynomials. *Advances in Applied Mathematics*, 32(1-2):5–9, 2004. doi:10.1016/S0196-8858(03)00041-1.
- 27 Timothy van Bremen and Ondrej Kuzelka. Faster lifting for two-variable logic using cell graphs. In *UAI*, volume 161 of *Proceedings of Machine Learning Research*, pages 1393–1402. AUAI Press, 2021. URL: <https://proceedings.mlr.press/v161/bremen21a.html>.
- 28 Timothy van Bremen and Ondrej Kuzelka. Lifted inference with tree axioms. *Artif. Intell.*, 324:103997, 2023. doi:10.1016/j.artint.2023.103997.
- 29 Guy Van den Broeck. On the completeness of first-order knowledge compilation for lifted probabilistic inference. In *NIPS*, pages 1386–1394, 2011. URL: <https://proceedings.neurips.cc/paper/2011/hash/846c260d715e5b854ffad5f70a516c88-Abstract.html>.
- 30 Guy Van den Broeck and Jesse Davis. Conditioning in first-order knowledge compilation and lifted probabilistic inference. In *AAAI*, pages 1961–1967. AAAI Press, 2012. doi:10.1609/aaai.v26i1.8404.
- 31 Guy Van den Broeck, Wannes Meert, and Adnan Darwiche. Skolemization for weighted first-order model counting. In *KR*. AAAI Press, 2014. URL: <http://www.aaai.org/ocs/index.php/KR/KR14/paper/view/8012>.
- 32 Moshe Y. Vardi. The complexity of relational query languages (extended abstract). In *STOC*, pages 137–146. ACM, 1982. doi:10.1145/800070.802186.
- 33 Yuanhong Wang, Juhua Pu, Yuyi Wang, and Ondrej Kuzelka. On exact sampling in the two-variable fragment of first-order logic. In *LICS*, pages 1–13. IEEE, 2023. doi:10.1109/LICS56636.2023.10175742.
- 34 Yuanhong Wang, Juhua Pu, Yuyi Wang, and Ondrej Kuzelka. Lifted algorithms for symmetric weighted first-order model sampling. *Artif. Intell.*, 331:104114, 2024. doi:10.1016/j.artint.2024.104114.

A Domain-Liftability of FO^2

We briefly describe the algorithm of computing WFOMC for FO^2 , as some of its concepts are important for designing algorithms for computing our polynomials efficiently. We mostly follow the notation from [27]. Any sentence in FO^2 can be reduced to the prenex normal form with only universal quantifiers (i.e., $\forall x \forall y \psi(x, y)$, where $\psi(x, y)$ is a quantifier-free formula) by the normalization in [7] and the technique of eliminating existential quantifiers in [31]. The reduction respects the WFOMC value, and hence in what follows, we assume that the input sentence is in the form $\forall x \forall y \psi(x, y)$. Two notions are needed to present the algorithm.

► **Definition 39** (1-type). *A 1-type of a first-order sentence Ψ is a maximally consistent set⁹ of literals formed from atoms in Ψ using only a single variable x .*

For example, $\Psi = \forall x \forall y (F(x) \wedge G(x, y))$ has four 1-types: $F(x) \wedge G(x, x)$, $F(x) \wedge \neg G(x, x)$, $\neg F(x) \wedge G(x, x)$ and $\neg F(x) \wedge \neg G(x, x)$. Intuitively, a 1-type interprets unary and reflexive binary predicates for a single domain element.

The *weighted model counting problem* (WMC) is defined similarly but the input formula is a ground quantifier-free formula.

⁹ A set of literals is maximally consistent if it is consistent (does not contain both a literal and its negation) and cannot be extended to a larger consistent set.

► **Definition 40** (Weighted Model Counting). *The WMC of a ground quantifier-free formula Φ under weighting functions (w, \bar{w}) is $\text{WMC}(\Phi, w, \bar{w}) := \sum_{\mu \in \mathcal{M}_\Phi} W(\mu, w, \bar{w})$, where \mathcal{M}_Φ denotes the set of all models of Φ .*

► **Example 41.** The WMC of a ground quantifier-free formula $\Phi = R(a, b) \rightarrow S(a)$ under $w(R) = 2, \bar{w}(R) = w(S) = \bar{w}(S) = 1$ is

$$W(\{R(a, b), S(a)\}, w, \bar{w}) + W(\{\neg R(a, b), S(a)\}, w, \bar{w}) + W(\{\neg R(a, b), \neg S(a)\}, w, \bar{w}) = 4.$$

Let $C_1(x), C_2(x), \dots, C_p(x)$ be all the 1-types of Ψ . When we consider domain-liftability, the sentence Ψ is fixed and therefore p is a constant. We partition domain elements into 1-types by enumerating the *configuration* $\mathbf{c} = (c_1, c_2, \dots, c_p)$, a vector indicating the number of elements partitioned to each 1-type, since in the symmetric WFOMC only the configuration matters. Then we can rewrite Ψ in the form of

$$\Psi = \bigwedge_{1 \leq i < j \leq p} \forall x \in C_i \forall y \in C_j (\psi(x, y) \wedge \psi(y, x)) \wedge \bigwedge_{1 \leq i \leq p} \forall x \in C_i \forall y \in C_i \psi(x, y),$$

where $\psi(x, y)$ is a quantifier-free formula.

Since we know the truth values of the unary and reflexive binary atoms of each 1-type, we may simplify the body of each conjunct by substituting every unary and reflexive binary literal with true or false as appropriate. Denote by $\psi_i(x, y)$ the simplification of $\psi(x, y)$ when both x and y belong to the same 1-type C_i , and by $\psi_{ij}(x, y)$ the simplification of $\psi(x, y) \wedge \psi(y, x)$ when x and y belong to C_i and C_j respectively. We then have:

$$\Psi = \bigwedge_{1 \leq i < j \leq p} \forall x \in C_i \forall y \in C_j \psi_{ij}(x, y) \wedge \bigwedge_{1 \leq i \leq p} \forall x \in C_i \forall y \in C_i \psi_i(x, y).$$

An important property here is that each conjunct is independent of others since no two conjuncts involve the same ground binary literal. Let

$$\begin{aligned} r_{i,j} &= \text{WMC}(\psi_{i,j}(a, b), w, \bar{w}), \\ s_i &= \text{WMC}(\psi_i(a, b) \wedge \psi_i(b, a), w, \bar{w}) \end{aligned}$$

be the *mutual* and *internal* 1-type coefficient of Ψ respectively. Then we have

$$\text{WFOMC}(\Psi, n, w, \bar{w}) = \sum_{|\mathbf{c}|=n} \binom{n}{\mathbf{c}} \prod_{1 \leq i \leq p} W(C_i, w, \bar{w})^{c_i} \cdot s_i^{\binom{c_i}{2}} \prod_{1 \leq i < j \leq p} r_{i,j}^{c_i c_j}, \quad (4)$$

which can be calculated in polynomial time in n .

B Graph Polynomial

We give a brief introduction to graph polynomials which are a class of polynomials defined on graphs that carry combinatorial meaning.

Tutte Polynomial. The Tutte polynomial [25] is one of the most well-known graph polynomials for undirected graphs. Consider an undirected graph $G = (V, E)$, where V is the set of vertices and E is the set of edges. The Tutte polynomial for the graph G is a polynomial in two variables x and y defined as

$$T_G(x, y) = \sum_{A \subseteq E} (x-1)^{cc(A) - cc(E)} (y-1)^{cc(A) + |A| - |V|},$$

where $cc(A)$ is the number of connected components considering only edges in A .

The Tutte polynomial holds significant combinatorial meaning. Evaluating the polynomial at specific points reveals various combinatorial properties. For example, $T_G(1, 1)$ counts the number of spanning forests of G (specifically, the number of spanning trees if G is connected), and $T_G(0, 2)$ counts the strongly connected orientations of G [15]. Substituting a concrete value for only one variable yields another graph polynomial of G . For example, $T_G(x, 0)$ corresponds to the chromatic polynomial [19] of G ; $T_G(0, y)$ corresponds to the flow polynomial [4] of G .

The computation of the Tutte polynomial for an arbitrary graph is known to be computationally challenging. Even the evaluation of a specific point of the polynomial is known to be #P-hard, with only a limited number of points being computationally tractable [9].

Directed Chromatic Polynomial. The idea of the chromatic polynomial of a directed graph appeared in [22], although it was originally stated in terms of ordered sets. Given a directed graph $D = (V, E)$ where $n = |V|$, let $\chi_D(x)$ be the number of ways of coloring each vertex in D by one of the colors $\{1, 2, \dots, x\}$ such that if there is an edge from u to v , the color of u is smaller than the color of v . We have the following lemma to state that $\chi_D(x)$ is a polynomial of x .

► **Lemma 42** ([22]). *For a directed graph D of size n , let $\chi_D^*(i)$ be the number of surjective colorings of vertices using i colors (i.e., each color should be used at least once) such that if there is an edge from u to v , the color of u is smaller than the color of v . It holds that*

$$\chi_D(x) = \sum_{i=1}^n \binom{x}{i} \chi_D^*(i) = \sum_{i=1}^n \frac{x(x-1)\cdots(x-i+1)}{i!} \chi_D^*(i),$$

where $\binom{x}{i} = 0$ if $i > x$.

► **Corollary 43.** $\chi_D(x)$ is a polynomial of x of degree n .

We call $\chi_D(x)$ the strict directed chromatic polynomial of D . A variant of it is the non-strict directed chromatic polynomial $\bar{\chi}_D(x)$ where for each positive integer x , $\bar{\chi}_D(x)$ equals the number of colorings of the vertices in D by one of the x colors $\{1, 2, \dots, x\}$ such that if there is an edge from u to v , the color of u is smaller than or equal to the color of v . Similarly to $\chi_D(x)$, $\bar{\chi}_D(x)$ is also a polynomial of x .

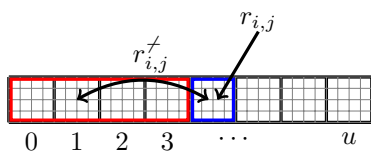
► **Corollary 44.** $\bar{\chi}_D(x)$ is a polynomial of x of degree n .

The relation between $\chi_D(x)$ and $\bar{\chi}_D(x)$ is given by the following lemma.

► **Lemma 45** ([22, 1]). *Let D be a directed graph and $\text{acyc}(D)$ be the directed acyclic graph obtained from D by condensing each cycle into a vertex. Denote the number of vertices in $\text{acyc}(D)$ by $|V(\text{acyc}(D))|$. We have*

$$\chi_D(x) = \begin{cases} (-1)^n \bar{\chi}_D(-x), & D \text{ is acyclic,} \\ 0, & \text{otherwise,} \end{cases}$$

$$\bar{\chi}_D(x) = (-1)^{|V(\text{acyc}(D))|} \chi_{\text{acyc}(D)}(-x).$$



■ **Figure 1** The 1-type structure of $\Psi_{R,u}$. Each tiny gray box represents an original 1-type of Ψ . The bold gray cells (each containing 4×4 boxes) refer to the replica of the original 1-types in each layer. The DP algorithm proceeds from left (cells $0, \dots, 3$) to right (cell 4).

C Omitted Proofs in Section 3

Proof of Proposition 14. Let Ψ and R be the input FO^2 sentence and the binary relation that WCP is defined on. Consider the point evaluation of the WCP at u . Let $C_1(x), C_2(x), \dots, C_p(x)$ be the valid 1-types¹⁰ of Ψ . Due to the definition of $\Psi_{R,u}$, the valid 1-types of $\Psi_{R,u}$ must have the form $C_i(x) \wedge C_{i'}^A(x)$, where $C_{i'}^A(x)$ is defined as

$$C_{i'}^A(x) = A_1(x) \wedge A_2(x) \wedge \dots \wedge A_{i'}(x) \wedge \neg A_{i'+1}(x) \wedge \dots \wedge \neg A_u(x)$$

for $i' \in [u]$, and

$$C_0^A(x) = \neg A_1(x) \wedge \neg A_2(x) \wedge \dots \wedge \neg A_u(x).$$

We use the tuple (i, i') to index the 1-type $C_i(x) \wedge C_{i'}^A(x)$. Then we have the following observation.

► **Observation 46.** Let $r_{(i,i'),(j,j')}$ be the mutual 1-type coefficient (defined in Appendix A) of $\Psi_{R,u}$. It holds that

$$r_{(i,i'),(j,j')} = \begin{cases} r_{i,j} = \text{WMC}(\psi_{i,j}(a, b), w, \bar{w}), & i' = j', \\ r_{i,j}^{\neq} = \text{WMC}(\psi_{i,j}(a, b) \wedge \neg R(a, b) \wedge \neg R(b, a), w, \bar{w}), & i' \neq j', \end{cases}$$

where $\psi_{i,j}(x, y)$ is the simplified formula of Ψ defined in Appendix A.

This observation provides an important insight into the structure of the new 1-types (as shown in Figure 1): we can imagine that the new 1-types of $\Psi_{R,u}$ form $u + 1$ layers, with each layer containing a replica of the original 1-types of Ψ . If two elements a, b fall in 1-types in the same layer, $R(a, b)$ can be either true or false. But if a and b are located in 1-types in different layers, $R(a, b)$ as well as $R(b, a)$ must be false due to the definition of $\Psi_{R,u}$ in Equation (2).

Therefore, we can perform a “blocked” DP, where each block consists of the 1-types in the same layer. We can write the new 1-type configuration as

$$c_{(1,1)}, \dots, c_{(p,1)}, c_{(1,2)}, \dots, c_{(p,2)}, \dots, c_{(1,u)}, \dots, c_{(p,u)}$$

where $c_{(i,i')}$ is the number of elements in the 1-type (i, i') . For any $\hat{u} \in [u]$, we define $h_{\hat{u},(c_1, c_2, \dots, c_p)}$ to be the weights of 1-types $(1, 1), \dots, (p, 1), \dots, (1, \hat{u}), \dots, (p, \hat{u})$ such that

¹⁰We call a 1-type *valid* if it is satisfiable in some model of the sentence.

$c_i = \sum_{j=1}^{\hat{u}} c_{(i,j)}$ for each $i \in [p]$, that is,

$$h_{\hat{u},(c_1,c_2,\dots,c_p)} = \sum_{c_{(1,1)}+\dots+c_{(1,\hat{u})}=c_1} \cdots \sum_{c_{(p,1)}+\dots+c_{(p,\hat{u})}=c_p} \frac{(\sum_{i \in [p]} c_i)!}{\prod_{i \in [p], i' \in [\hat{u}]} c_{(i,i')}!} \prod_{i \in [p], i' \in [\hat{u}]} W(C_i, w, \bar{w})^{c_{(i,i')}} \cdot s_i^{\binom{c_{(i,i')}}{2}} \prod_{\substack{i,j \in [p], i',j' \in [\hat{u}], \\ (i,i') < (j,j')}} (r_{(i,i'),(j,j')})^{c_{(i,i')}c_{(j,j')}} ,$$

where $(i, i') < (j, j')$ means $i < j$ or $i = j \wedge i' < j'$.

For simplicity, we use \mathbf{c} to denote (c_1, c_2, \dots, c_p) . Then the new term $h_{\hat{u},\mathbf{c}}$ can be also computed by a DP algorithm:

$$h_{\hat{u},\mathbf{c}} = \sum_{\bar{\mathbf{c}} + \mathbf{c}^* = \mathbf{c}} \binom{|\mathbf{c}|}{|\mathbf{c}^*|} \cdot h_{\hat{u}-1,\bar{\mathbf{c}}} \cdot w_{in}(\mathbf{c}^*) \cdot w_{cross}(\bar{\mathbf{c}}, \mathbf{c}^*), \quad (5)$$

where recall that $\bar{\mathbf{c}} + \mathbf{c}^* = \mathbf{c}$ means that $\bar{c}_i + c_i^* = c_i$ for each $i \in [p]$,

$$w_{in}(\mathbf{c}^*) = \binom{|\mathbf{c}^*|}{\mathbf{c}^*} \prod_{i=1}^p W(C_i, w, \bar{w})^{c_i^*} \cdot s_i^{\binom{c_i^*}{2}} \prod_{1 \leq i < j \leq p} (r_{i,j})^{c_i^* c_j^*}$$

is the weight within layer \hat{u} , and

$$w_{cross}(\bar{\mathbf{c}}, \mathbf{c}^*) = \prod_{i=1}^p \prod_{j=1}^p (r_{i,j}^{\neq})^{\bar{c}_i c_j^*}$$

is the weight between the layer \hat{u} and the layers before \hat{u} . The initial values are $h_{0,\mathbf{c}} = w_{in}(\mathbf{c})$ for each \mathbf{c} . Then the final WFOMC of $\Psi_{R,u}$ can be obtained by:

$$\text{WFOMC}(\Psi_{R,u}, n, w, \bar{w}) = \sum_{|\mathbf{c}|=n} h_{u,\mathbf{c}}.$$

The new DP algorithm is summarized in Algorithm 1, where `interpolate_1d(...)` takes $n + 1$ points as input and returns the corresponding polynomial of degree n .

■ **Algorithm 1** Computing the WCP.

Input: A sentence Ψ , weighting functions w, \bar{w} , the relation R , and an integer n
Output: $f_n(u; \Psi, w, \bar{w}, R)$

- 1 $h_{0,\mathbf{c}} \leftarrow w_{in}(\mathbf{c})$ **for each** $|\mathbf{c}| \leq n$ $\triangleright \mathbf{c} = (c_1, \dots, c_p)$
- 2 $evaluation_pairs \leftarrow \{(0, \sum_{|\mathbf{c}|=n} h_{0,\mathbf{c}})\}$
- 3 **for** $u \leftarrow 1$ **to** n **do**
- 4 **foreach** \mathbf{c} such that $|\mathbf{c}| \leq n$ **do**
- 5 $h_{u,\mathbf{c}} \leftarrow \sum_{\bar{\mathbf{c}} + \mathbf{c}^* = \mathbf{c}} \binom{|\mathbf{c}|}{|\mathbf{c}^*|} \cdot h_{u-1,\bar{\mathbf{c}}} \cdot w_{in}(\mathbf{c}^*) \cdot w_{cross}(\bar{\mathbf{c}}, \mathbf{c}^*)$
- 6 $evaluation_pairs \leftarrow evaluation_pairs \cup \{(u, \sum_{|\mathbf{c}|=n} h_{u,\mathbf{c}})\}$
- 7 $f \leftarrow interpolate_1d(evaluation_pairs)$
- 8 **return** f

Moreover, any WFOMC problem for C^2 sentences with cardinality constraints can be reduced to WFOMC of an FO^2 sentence, which can be solved in polynomial time in n by the algorithm presented above. The interpolation step from $n + 1$ point evaluations to the polynomial can also be done in polynomial time in n , completing the proof. ◀

Proof of Theorem 16. Consider a C^2 sentence Ψ with cardinality constraints, a domain of size n weighting functions w, \bar{w} , and a distinguished binary predicate R . We need another two predicates to represent the bipartite graph. Define

$$\begin{aligned} \Psi_b = & \Psi_R \wedge \forall x ((P_1(x) \vee P_2(x)) \wedge (\neg P_1(x) \vee \neg P_2(x))) \\ & \wedge \forall x \forall y (P_1(x) \wedge P_1(y) \vee P_2(x) \wedge P_2(y) \rightarrow \neg R(x, y)), \end{aligned}$$

where $w(P_1) = \bar{w}(P_1) = w(P_2) = \bar{w}(P_2) = 1$. The predicates P_1, P_2 divide domain elements into two groups to ensure that $G(R)$ is a bipartite graph. However, overcounting occurs since for every connected component in the bipartite graph, we have two choices of assigning vertices to P_1 and P_2 . Hence, for each connected component, we need to multiply the WFOMC by a factor $\frac{1}{2}$ to its weight, which can be done using WCP:

$$\begin{aligned} \text{WFOMC}(\Psi \wedge \text{bipartite}(R), n, w, \bar{w}) &= \sum_{\mu \in \mathcal{M}_{\Psi_b, n}} W(\mu, w, \bar{w}) \cdot \left(\frac{1}{2}\right)^{cc(\mu_R)} \\ &= f_n\left(-\frac{1}{2}; \Psi_b, w, \bar{w}, R\right). \quad \blacktriangleleft \end{aligned}$$

Proof of Theorem 19. Consider a C^2 sentence Ψ with cardinality constraints, a domain of size n , weighting functions w, \bar{w} and a distinguished binary predicate R . By definition, the tree axiom requires that $G(R)$ should be connected and the number of edges in $G(R)$ should be $n - 1$. By Corollary 18, we have

$$\text{WFOMC}(\Psi \wedge \text{tree}(R), n, w, \bar{w}) = [uv^{2(n-1)}] \hat{f}_n(u-1, v; \Psi_R, w, \bar{w}, R). \quad \blacktriangleleft$$

Proof of Theorem 20. Consider a C^2 sentence Ψ with cardinality constraints, a domain of size n , weighting functions w, \bar{w} and a distinguished binary predicate R . By definition, the forest axiom requires the number of edges in $G(R)$ to be equal to the difference between the number of vertices in $G(R)$ and the number of connected components in $G(R)$. By Corollary 18, we have

$$\text{WFOMC}(\Psi \wedge \text{forest}(R), n, w, \bar{w}) = \sum_{i \in [n]} [u^i v^{2(n-i)}] \hat{f}_n(u-1, v; \Psi_R, w, \bar{w}, R). \quad \blacktriangleleft$$