Weighted model counting beyond two-variable logic

Antti Kuusisto
University of Bremen
Germany

Carsten Lutz
University of Bremen
Germany

Abstract

It was recently shown by van den Broeck at al. that the symmetric weighted first-order model counting problem (WFOMC) for sentences of two-variable logic FO^2 is in polynomial time, while it is $\#P_1$-complete for some FO^3-sentences. We extend the result for FO^2 in two independent directions: to sentences of the form $\varphi \wedge \forall x \exists^{=1} y\, \psi(x, y)$ with φ and ψ formulated in FO^2 and to sentences of the uniform one-dimensional fragment U_1 of FO, a recently introduced extension of two-variable logic with the capacity to deal with relation symbols of all arities. We note that the former generalizes the extension of FO^2 with a functional relation symbol. We also identify a complete classification of first-order prefix classes according to whether WFOMC is in polynomial time or $\#P_1$-complete.

Keywords weighted model counting, tractability, two-variable logic, enumerative combinatorics

1 Introduction

The first-order model counting problem asks, given a sentence φ and a number n, how many models of φ of size n exist. (The domain of the models is taken to be $\{0, \ldots, n-1\}$.) The weighted variant of this problem adds weights to atomic facts $R^{\mathfrak{M}}(u_1, \ldots, u_k)$ of models $\mathfrak{M}$, the total weight of $\mathfrak{M}$ being the product of the atomic weights. The question is then what the sum of the weights of all models of φ of size n is. Following [12], we also admit weights of negative facts '*not* $R^{\mathfrak{M}}(u_1, \ldots, u_k)$'.

We investigate the symmetric weighted model counting problem of systems extending the two-variable fragment FO^2 of first-order logic FO. The word 'symmetric' indicates that each weight is determined by the relation symbol of the (positive or negative) fact and thus the weights can be specified by weight functions w and $\bar{w}$ that assign weights to each relation symbol occurring positively (w) or negatively ($\bar{w}$). We let WFOMC refer to the symmetric weighted first-order model counting problem, with $\mathrm{WFOMC}(\varphi, n, w, \bar{w})$ denoting the sum of the weights of models $\mathfrak{M} \models \varphi$ of size n according to the weight functions w and $\bar{w}$. We focus on studying the *data complexity* of WFOMC, that is, the complexity of determining $\mathrm{WFOMC}(\varphi, n, w, \bar{w})$ where n is the only input, given in unary, and with $\varphi, w, \bar{w}$ fixed.

The recent article [4] established the by now well-known result that the data complexity of WFOMC is in polynomial time for formulae of FO^2, while [3] demonstrated that the three-variable fragment FO^3 contains formulae for which the problem is $\#P_1$-complete. We note that the non-symmetric variant of the problem is known to be #P-complete for some FO^2-sentences [3].

Weighted model counting problems have a range of well-known applications. For example, as pointed out in [3], WFOMC problems occur in a natural way in *knowledge bases with soft constraints* and are especially prominent in the area of Markov logic [6]. For a recent comprehensive survey on these matters, see [5]. From the mathematical perspective, WFOMC offers a neat and general approach to *elementary enumerative combinatorics.* To give a simple illustration of this, consider $\mathrm{WFOMC}(\varphi, n, w, \bar{w})$ for the two-variable logic sentence $\varphi = \forall x \forall y (Rxy \rightarrow (Ryx \wedge x \neq y))$ with $w(R) = \bar{w}(R) = 1$. The sentence states that R encodes a simple undirected graph and thus $\mathrm{WFOMC}(\varphi, n, w, \bar{w}) = 2^{\binom{n}{2}}$, the number of graphs of order n (with the set n of vertices). Thus WFOMC provides a *logic-based way of classifying combinatorial problems.* For instance, the result for FO^2-properties from [4] shows that all these properties can be associated with tractable enumeration functions. For discussions of the links between weighted model counting, the spectrum problem and 0-1 laws, see [3].

In the current paper, we extend the result of [4] for FO^2 in two independent directions. We first consider FO^2 with a *functionality axiom*, that is, sentences of type $\varphi \wedge \forall x \exists^{=1} y\, \psi(x, y)$ with φ and ψ in FO^2. This extension is motivated, inter alia, by certain description logics with *functional roles* [1]. The connection of WFOMC to enumerative combinatorics also provides an important part of the motivation. Indeed, while FO^2 is a reasonable formalism for specifying properties of relations, adding functionality axioms allows us to also express properties of functions, possibly combined with relations. For example, applying WFOMC to the sentence $\forall x \neg Rxx \wedge \forall x \exists^{=1} y Rxy$ gives the number of functions that do not have a fixed point. While the extension of FO^2 with a functionality axiom might appear simple at first sight, showing that the data complexity of WFOMC remains in PTIME requires a rather different and much more involved approach than that for FO^2. Our proofs provide concrete and insightful aritmetic expressions for analysing the related weighted model counts. We note that the article [9] considers weighted model counting of an orthogonal extension of FO^2 which can express that some relations are functions.

We also show that the data complexity of WFOMC remains in PTIME for sentences of the *uniform one-dimensional fragment* U_1. This is a recently introduced [8, 10] extension of FO^2 that preserves NEXPTIME-completeness of the satisfiability problem while admitting more than two variables and thus being able to speak about relations of all arities in a meaningful way. The fragment U_1 is obtained from FO by restricting quantification to blocks of existential (universal) quantifiers that leave at most one variable free, a restriction referred to as the *one-dimensionality* condition. Additionally, a *uniformity condition* is imposed: if $k, n \geq 2$, then a Boolean combination of atoms $Rx_1 \ldots x_k$ and $Sy_1 \ldots y_n$ is allowed only if the sets $\{x_1, \ldots, x_k\}$ and $\{y_1, \ldots, y_n\}$ of variables are equal. Boolean combinations of formulae with at most one free variable

LICS '18, July 9–12, 2018, Oxford, United Kingdom

ACM ISBN 978-1-4503-5583-4/18/07…$15.00
https://doi.org/10.1145/3209108.3209168

can be formed freely, and the use of equality is unrestricted. It is shown in [8] that lifting either of these conditions—in a minimal way—leads to undecidability. For a survey of the basic properties of U_1 and its relation to modal and description logics, see [14].

What makes weighted model counting for U_1 attractive in relation to applications is the ability of U_1 to express interesting properties of relations of all arities, thereby banishing one of the main weaknesses of FO^2. This is especially well justified from the points of view of database theory and of knowledge representation with formalisms such as Markov logic, which are among the main application areas of WFOMC. We note that U_1 is significantly more expressive than FO^2 already in restriction to models with at most binary relations [14].

We also identify a complete classification of first-order prefix classes according to whether the sentences of the particular class have polynomial time WFOMC or whether some sentence of the class has a $\#P_1$-complete WFOMC. This classification, whose proof makes significant use of the results and techniques from [3, 4], is remarkably simple: $\#P_1$-hardness arises precisely for the classes with more than two quantifiers, independently of the quantifier pattern.

2 Preliminaries

The natural numbers are denoted by $\mathbb{N}$ and positive integers by $\mathbb{Z}_+$. As usual, we often identify $n \in \mathbb{N}$ with the set $\{ k \in \mathbb{N} \mid k < n\}$. We define $[n] := \{1, \dots, n\}$ for each $n \in \mathbb{Z}_+$ and $[0] := \emptyset$. The domain of a function f is denoted by $dom(f)$. The function f is *involutive* if $f(f(x)) = x$ for all $x \in dom(f)$ and *anti-involutive* if $f(f(x)) \neq x$ for all $x \in dom(f)$. Two functions f and g are *nowhere inverses* if $f(g(x)) \neq x$ and $g(f(y)) \neq y$ for all $x \in dom(g)$, $y \in dom(f)$. We use the standard notation $\binom{n}{n_1,\dots,n_m}$ for multinomial coefficients.

We study (fragments of) first-order logic FO over relational vocabularies; constant and function symbols are not allowed. The identity symbol '=' and the Boolean constants $\bot, \top$ are *not* considered relation symbols; they are a logical symbols included in FO. We allow nullary relation symbols in FO with the usual syntax and semantics. The vocabulary of a formula φ is denoted by $voc(\varphi)$.

We let VAR $:= \{v_0, v_1, \dots\}$ denote a fixed, countably infinite set of variable symbols. We mainly use meta-variables x, y, z, etc., in order to refer to symbols in VAR. Note that for example x and y may denote the same variable, while v_i and v_j are different if $i \neq j$.

The domain of a model $\mathfrak{M}$ is denoted by $dom(\mathfrak{M})$. In the case $A \subseteq (dom(\mathfrak{M}))^k$, we let $(\mathfrak{M}, A)$ denote the expansion of $\mathfrak{M}$ obtained by adding the k-ary relation A to $\mathfrak{M}$. We mostly do not differentiate between relations and relation symbols explicitly when the distinction is clear from the context. Relational models decompose into *facts* and *negative facts* in the usual way: if R is a k-ary relation symbol of a model $\mathfrak{M}$ and $Ru_1 \dots u_k$ holds for some elements $u_1, \dots, u_k$ of $\mathfrak{M}$, then $Ru_1 \dots u_k$ is a positive fact of $\mathfrak{M}$, and if $Ru_1 \dots u_k$ does not hold in $\mathfrak{M}$, then $Ru_1 \dots u_k$ is a negative fact of $\mathfrak{M}$. We denote the positive (respectively, negative) facts of $\mathfrak{M}$ by $F^+(\mathfrak{M})$ (respectively, $F^-(\mathfrak{M})$). The *span* of a fact $Ru_1, \dots, u_k$, whether positive or negative, is $\{u_1, \dots, u_k\}$ and the *size* of the span is $|\{u_1, \dots, u_k\}|$.

The *first-order model counting problem* asks, when given a positive integer n in *unary* and an FO-sentence φ, how many models φ has over the domain $n = \{0, \dots, n-1\}$; the vocabulary of the models is taken to be $voc(\varphi)$, and different but isomorphic models contribute separately to the output. The *weighted first-order model counting problem* adds two functions to the input, w and $\bar{w}$, that both map the set of all possible facts over n and $voc(\varphi)$ into a set of weights. In the *symmetric* weighted model counting problem studied in this paper, w and $\bar{w}$ are functions $w : voc(\varphi) \to \mathbb{Q}$ and $\bar{w} : voc(\varphi) \to \mathbb{Q}$. The output WFOMC$(\varphi, n, w, \bar{w})$ is then the sum of the *weights* $W(\mathfrak{M}, w, \bar{w})$ of all models $\mathfrak{M} \models \varphi$ with domain n and vocabulary $voc(\varphi)$,

$$W(\mathfrak{M}, w, \bar{w}) := \prod_{Ru_1\dots u_k \,\in\, F^+(\mathfrak{M})} w(R) \cdot \prod_{Ru_1\dots u_k \,\in\, F^-(\mathfrak{M})} \bar{w}(R). \quad (1)$$

This setting gives rise to several computational problems, depending on which inputs are fixed. In this article, we exclusively study *data complexity*, i.e., the problem of computing WFOMC$(\varphi, n, w, \bar{w})$ with the sole input $n \in \mathbb{Z}_+$ given in unary; φ, w and $\bar{w}$ are fixed and thus not part of the input. Algorithms for more general inputs can easily be extracted from our proofs, but we only study data complexity explicitly for the lack of space.

While weights are rational numbers, it will be easy to see that reals with a tame enough representation could also be included without sacrificing our results. We ignore this for the sake of simplicity and stick to rational weights. (See also [12].)

We now define, for technical purposes, some restricted versions of WFOMC and the operator W. First, if $\mathcal{M}$ is a class of models, we define WFOMC$(\varphi, n, w, \bar{w}) \upharpoonright \mathcal{M}$ to be the sum of the weights W$(\mathfrak{M}, w, \bar{w})$ of models $\mathfrak{M} \in \mathcal{M}$ with domain n and vocabulary $voc(\varphi)$ such that $\mathfrak{M} \models \varphi$. For $k \in \mathbb{Z}_+$, we let $F_k^+(\mathfrak{M})$ and $F_k^-(\mathfrak{M})$ denote the restrictions of $F^+(\mathfrak{M})$ and $F^-(\mathfrak{M})$ to facts with span of size k. We define $W_k(\mathfrak{M}, w, \bar{w})$ exactly as $W(\mathfrak{M}, w, \bar{w})$ but with $F^+(\mathfrak{M})$ and $F^-(\mathfrak{M})$ replaced by $F_k^+(\mathfrak{M})$ and $F_k^-(\mathfrak{M})$. When φ, n, w and $\bar{w}$ are clear from the context, the *weight of a class $\mathcal{M}$ of models* refers to WFOMC$(\varphi, n, w, \bar{w}) \upharpoonright \mathcal{M}$.

The quantifier-free part of a prenex normal form formula of FO is called a *matrix*. A prenex normal form sentence of type $\chi := \forall x_1 \dots \forall x_k \psi$, where ψ is the matrix, is an $\forall^*$*-sentence*, and the number k of quantifiers in χ is the *width* of χ. An $\exists^*$-sentence is defined analogously.

Below we will investigate standard two-variable logic FO^2 enhanced with a *functionality axiom*. Formulae in this language are conjunctions of type $\varphi \wedge \forall x \exists^{=1} y \psi(x, y)$, where φ and ψ are FO^2-formulae, ψ with the free variables x, y and φ a sentence. When studying this variant of FO, we exclusively use the variables x, y, with x denoting v_1 and y denoting v_2.

Let $Y = \{y_1, \dots, y_k\}$ be a set of distinct variables, and let R be an n-ary relation symbol for some $n \geq k$. An atom $Ry_{i_1} \dots y_{i_n}$ is a Y*-atom* if $\{y_{i_1}, \dots, y_{i_n}\} = Y$. For example, if x, y, z, v are distinct variable symbols, then $Txyzx$ and $Sxzy$ are $\{x, y, z\}$-atoms, while $Uxyzv$ and Vxy are not. Furthermore, Vxz is an $\{x, z\}$-atom while $x = z$ is not, as identity is not a relation symbol. A Y*-literal* is a Y-atom $Ry_{i_1} \dots y_{i_n}$ or a negated Y-atom $\neg Ry_{i_1} \dots y_{i_n}$. A Y-literal is *an m-ary literal* if $|Y| = m$, so for example Sxx and $\neg Px$ are unary literals; Sxx is even a unary atom while $\neg Px$ is not. A *higher arity literal* is a literal of arity at least two. We let $diff(x_1, \dots, x_k)$ denote the conjunction of inequalities $x_i \neq x_j$ for all distinct $i, j \in [k]$.

The set of formulae of the *uniform one-dimensional fragment* U_1 of FO is the smallest set $\mathcal{F}$ such that the following conditions hold.

1. Unary and nullary atoms are in $\mathcal{F}$.
2. All identity atoms $x = y$ are in $\mathcal{F}$.
3. If $\varphi, \psi \in \mathcal{F}$, then $\neg\varphi \in \mathcal{F}$ and $(\varphi \wedge \psi) \in \mathcal{F}$.

4. Let $X = \{x_0, \ldots, x_k\}$ and $Y \subseteq X$. Let φ be a Boolean combination of Y-atoms and formulae in $\mathcal{F}$ whose free variables (if any) are in X. Then
   a. $\exists x_1 \ldots \exists x_k\, \varphi \in \mathcal{F}$,
   b. $\exists x_0 \ldots \exists x_k\, \varphi \in \mathcal{F}$.

For example $\exists y \exists z((\neg Rxyz \vee Tzyxx) \wedge Qy)$ is a U_1-formula while $\exists x \exists y(Sxy \wedge Sxz)$ is not, as $\{x, y\} \neq \{x, z\}$. This latter formula is said to violate the uniformity condition of U_1. Also $\exists z \forall y \forall x(Txyz \wedge \exists uSxu)$ is a U_1-formula while $\exists x \exists y \exists z(Txyz \wedge \exists uTxyu)$ is not, as $\exists uTxyu$ leaves two variables free and thereby violates the one-dimensionality condition of U_1. The clause 4 above does not require that Y-atoms *must* be included, so also $\exists x \exists y \exists z\, diff(x, y, z)$ is a U_1-formula. We thus see that U_1 has some counting capacities. A matrix of a U_1-formula will below be called a U_1-*matrix*.

The article [14] contains a survey of U_1 with background about its expressive power and connections to extended modal logics. The article [11] provides an Ehrenfeucht-Fraïssé game characterization of U_1. It is worth noting that the so-called *fully uniform one-dimensional fragment* FU_1 has *exactly* the same expressive power as FO^2 when restricting to vocabularies with at most binary relations [14]. The logic FU_1 is obtained by dropping clause 2 from the above definition of U_1 and instead regarding the identity symbol as an ordinary binary relation in clause 4; see [14]. Thus U_1 is the extension of FU_1 with unrestricted use of identity.

The formula $\exists x \exists y \exists z\, diff(x, y, z)$ is an obvious example of a U_1-formula that is not expressible in FO^2. Another formula worth mentioning here that separates the expressive powers of U_1 and FO^2 is $\exists x \forall y \forall z(Ryz \rightarrow (x = y \vee x = z))$ which states that some node is part of every edge of R. The separation was shown in [14], and the proof is easy; simply consider the two-pebble game (defined in, e.g., [7]) on the complete graphs K_2 and K_3. The U_1-formula $\exists x \exists y \exists z \neg Sxyz$ is one of the simplest formulae separating U_1 from *both* FO^2 and the guarded negation fragment [2], as shown in [14].

For technical purposes, we also introduce the *strongly restricted* fragment of U_1, denoted SU_1, which was originally introduced and studied in [11]. The logic SU_1 imposes the additional condition on the above clause 4 that the set Y must contain exactly all of the variables $x_0, \ldots, x_k$. For example $\exists x \exists y \exists u(Rxyu \wedge x \neq u)$ is an SU_1-formula while $\exists x \exists y(Sxy \wedge x \neq z)$ is not, despite being a U_1-formula, as $z \notin \{x, y\}$. Despite the syntactic restriction imposed by SU_1 being simple, it has some significant consequences: it is shown in [11] that the satisfiability problem of SU_1 in the presence of a *single* built-in equivalence relation is only NEXPTIME-complete, while it is 2NEXPTIME-complete for U_1. We note that even the restriction SU_1 of U_1 contains FO^2 as a syntactic fragment.

A U_1-sentence φ is in *generalized Scott normal form*, if

$$\varphi = \bigwedge_{1 \leq i \leq m_\forall} \forall x_1 \ldots \forall x_{\ell_i}\, \varphi_i^\forall(x_1, \ldots, x_{\ell_i}) \wedge \bigwedge_{1 \leq i \leq m_\exists} \forall x \exists y_1 \ldots \exists y_{k_i} \varphi_i^\exists(x, y_1, \ldots, y_{k_i}),$$

where $\varphi_i^\exists$ and $\varphi_i^\forall$ are quantifier-free. A sentence of FO^2 is in (standard) Scott normal form if it is of type

$$\forall x \forall y\, \varphi(x, y) \wedge \quad \bigwedge_{1 \leq i \leq m_\exists} \forall x \exists y \psi_i(x, y)$$

with φ and each ψ_i quantifier-free. There exists a standard procedure (see, e.g., [7, 10]) that converts any given formula φ of FO^2 (respectively, U_1) in polynomial time into a formula $Sc(\varphi)$ in standard (respectively, generalized) Scott normal form such that φ is equivalent to $\exists P_1 \ldots \exists P_n Sc(\varphi)$, where $P_1, \ldots, P_n$ are fresh unary and nullary predicates. The procedure is well-known and used in most papers on FO^2 and U_1. Thus we only describe it briefly. For more details, see Appendix A.1 of the full version [15] of the current paper. The main idea is to replace, starting from the atomic level and working upwards from there, any subformula $\psi(x) = Qx_1 \ldots Qx_k \chi$, where $Q \in \{\forall, \exists\}$ and χ is quantifier-free, with an atomic formula $P_\psi(x)$, where P_ψ is a fresh relation symbol. This atom $P_\psi(x)$ is then separately axiomatized to be equivalent to $\psi(x)$.

If φ is a sentence of U_1 (respectively SU_1, FO^2), then $Sc(\varphi)$ is likewise a sentence of U_1 (respectively SU_1, FO^2); see Appendix A.1 of [15]. Each novel predicate (P_ψ in the above example) is axiomatized to be equivalent to the subformula ($\psi(x)$ in the above example) whose quantifiers are to be eliminated, so the interpretation of the predicate is fully determined by the subformula in every model of the ultimate Scott normal form sentence. Thus, recalling that $\varphi \equiv \exists P_1 \ldots \exists P_k Sc(\varphi)$, where $P_1, \ldots P_k$ are the fresh predicates, we get the following (see Appendix A.1 of [15] and cf. [4]).

Lemma 2.1. $\mathrm{WFOMC}(\varphi, n, w, \bar{w}) = \mathrm{WFOMC}(Sc(\varphi), n, w', \bar{w}')$, *where* w' *and* $\bar{w}'$ *map the fresh symbols to* 1.

2.1 Types and tables

Let η be a finite relational vocabulary. A 1-type (over η) is a maximally consistent set of η-atoms and negated η-atoms in the single variable v_1; no nullary atoms are included. The number of 1-types over η is clearly finite. We often identify a 1-type α with the conjunction of its elements, whence $\alpha(v_1)$ is a formula in the single variable v_1. While the official variable with which α is defined is v_1, we frequently consider 1-types $\alpha(x), \alpha(y)$, etc., with v_1 replaced by other variables. To see some examples, consider the case where $\eta = \{R, P\}$ with R binary and P unary. Then the 1-types over η in the variable x are $Rxx \wedge Px$, $\neg Rxx \wedge Px$, $Rxx \wedge \neg Px$ and $\neg Rxx \wedge \neg Px$.

Let $\mathfrak{M}$ be an η-model and α a 1-type over η. An element $u \in dom(\mathfrak{M})$ *realizes* the 1-type α if $\mathfrak{M} \models \alpha(u)$. Note that every element of $\mathfrak{M}$ realizes exactly one 1-type over η.

Let $k \geq 2$ be an integer. A k-*table* over η is a maximally consistent set of $\{v_1, \ldots, v_k\}$-atoms and negated $\{v_1, \ldots, v_k\}$-atoms over η. We define that 2-tables do *not* contain identity atoms or negated identity atoms. For example, using x, y instead of v_1, v_2, the set $\{Rxxy, Rxyx, \neg Ryxx, Ryyx, \neg Ryxy, Rxyy, Sxy, \neg Syx\}$ is a 2-table over $\{R, S\}$, where R is a ternary and S a binary symbol. We often identify a k-table β with a conjunction of its elements. We also often consider formulae such as $\beta(x_1, \ldots, x_k)$, thereby writing k-tables in terms of variables other than $v_1, \ldots, v_k$.

For investigations on two-variable logic, we also need the notion of a 2-type. Recalling that we let x and y denote, respectively, v_1 and v_2 in two-variable contexts, we define that a 2-*type* over η is a conjunction $\beta(x, y) \wedge \alpha_1(x) \wedge \alpha_2(y) \wedge x \neq y$, where β is a 2-table while α_1 and α_2 are 1-types over η. Such a 2-type can be conveniently denoted by $\alpha_1 \beta \alpha_2$.

Let γ be either a 1-type or a k-table over η. Let L_+ and L_- be the sets of positive and negative literals in γ. Given weight functions $w : \eta \to \mathbb{Q}$ and $\bar{w} : \eta \to \mathbb{Q}$, the *weight of* γ, denoted by $\langle w, \bar{w} \rangle(\gamma)$, is the product $\prod_{R\overline{v} \in L_+} w(R) \cdot \prod_{\neg R\overline{v} \in L_-} \bar{w}(R)$, where $\overline{v}$ denotes the different possible tuples of variables in the literals of γ.

2.2 A Skolemization procedure

We now define a formula transformation procedure designed for the purposes of model counting. The procedure, which was originally introduced in [4], resembles Skolemization but does not in general produce an equisatisfiable formula. Here we present a slightly modified variant of the procedure from [4] suitable for our purposes.

If $Q \in \{\exists, \forall\}$ is a quantifier, we let Q' denote the *dual quantifier* of Q, i.e., $Q' \in \{\exists, \forall\} \setminus \{Q\}$. Let

$$\varphi := \forall x_1 \dots \forall x_k \exists y_1 \dots \exists y_m Q_1 z_1 \dots Q_n z_n\, \psi$$

be a first-order prenex normal form sentence where ψ is quantifier-free and $Q_i \in \{\exists, \forall\}$ for all i. We eliminate the block $\exists y_1 \dots \exists y_m$ of existential quantifiers of φ in two steps. First we replace φ by

$$\forall x_1 \dots \forall x_k (A x_1 \dots x_k \vee \neg \exists y_1 \dots \exists y_m Q_1 z_1 \dots Q_n z_n\, \psi),$$

where A is a fresh k-ary predicate. Then the negation is pushed inwards past the quantifier block $\exists y_1 \dots \exists y_m Q_1 z_1 \dots Q_n z_n$ and the resulting dual block $\forall y_1 \dots \forall y_m Q'_1 z_1 \dots Q'_n z_n$ is pulled out so that we end up with the prenex normal form sentence

$$\forall x_1 \dots \forall x_k \forall y_1 \dots \forall y_m Q'_1 z_1 \dots Q'_n z_n\, (A x_1 \dots x_k \vee \neg\psi).$$

Let $Sk_0(\varphi)$ denote the sentence obtained by changing the maximally long outermost block of existential quantifiers (the block $\exists y_1 \dots \exists y_m$ if $Q_1 = \forall$ above) to a block of universal quantifiers using the above two steps, and let $Sk(\varphi)$ be the $\forall^*$-sentence obtained by repeatedly applying Sk_0. For any conjunction $\chi := \psi_1 \wedge \dots \wedge \psi_n$ of prenex normal form sentences, we let $Sk(\chi) := Sk(\psi_1) \wedge \dots \wedge Sk(\psi_n)$.

The next lemma is proved similarly as the corresponding result in [4]. (Appendix A.2 of [15] also gives a proof.)

Lemma 2.2 (cf. [4])**.** *Let χ and φ be sentences, φ a conjunction of prenex normal form sentences. Let w and $\bar{w}$ be weight functions. Then*

$$\mathrm{WFOMC}(\varphi \wedge \chi, n, w, \bar{w}) = \mathrm{WFOMC}(Sk(\varphi) \wedge \chi, n, w', \bar{w}'),$$

where w' and $\bar{w}'$ are obtained from w and $\bar{w}$ by mapping the fresh symbols in $Sk(\varphi)$ to 1 in the case of w' and to -1 in the case of $\bar{w}'$. If φ is a sentence of FO^2, then so is $Sk(\varphi)$. If φ is a sentence of $\mathcal{L} \in \{\mathrm{U}_1, \mathrm{SU}_1\}$ in generalized Scott normal form, then $Sk(\varphi) \in \mathcal{L}$.

2.3 Further syntactic assumptions

Let φ be a sentence of U_1. Due to Lemmas 2.1 and 2.2, we have $\mathrm{WFOMC}(\varphi, n, w, \bar{w}) = \mathrm{WFOMC}(Sk(Sc(\varphi)), n, w', \bar{w}')$, where w' and $\bar{w}'$ treat the fresh symbols as discussed when defining Sc and Sk. Call $\chi := Sk(Sc(\varphi))$ and assume, w.l.o.g., that $\chi = \forall x_1 \chi_1 \wedge \dots \wedge \forall x_1 \dots \forall x_k \chi_k$ for some matrices χ_i. For technical convenience, when working with SU_1, we assume that there is at most one $\forall^*$-conjunct of any particular width; if not, sentences $\forall x_1 \dots \forall x_p \chi'$ and $\forall x_1 \dots \forall x_p \chi''$ can always be combined to $\forall x_1 \dots \forall x_p (\chi' \wedge \chi'')$.

Now, χ may contain nullary predicates. Let S be the set of nullary predicates of χ and let $f : S \to \{\top, \bot\}$ be a function. Let χ^f be the formula obtained from χ by replacing each nullary predicate P by $f(P)$. It is easy to compute $\mathrm{WFOMC}(\chi, n, v, \bar{v})$ from the values $\mathrm{WFOMC}(\chi^f, n, v, \bar{v})$ for all functions $f : S \to \{\top, \bot\}$. Thus, when studying WFOCM for U_1 and SU_1, we begin with a formula $\forall x_1 \chi_1 \wedge \dots \wedge \forall x_1 \dots \forall x_k \chi_k$ assumed to be free of nullary predicates. We also assume, w.l.o.g., that the greatest width k is at least 2 and equal to the greatest arity of relation symbols occurring in the formula. (We can always add dummy $\forall^*$-conjuncts of higher width, and we can add a dummy k-ary symbol R to a conjunct $\forall x_1 \dots \forall x_k \chi_k$ by replacing χ_k by $R x_1 \dots x_k \wedge \chi_k$ and setting $w(R) = \bar{w}(R) = 1$.)

We then turn to two-variable logic with a functionality axiom. Consider a sentence $\varphi' := \varphi \wedge \forall x \exists^{=1} y \psi(x, y)$, where φ and $\psi(x, y)$ are FO^2-formulae. By applying the Scott normal form procedure for eliminating quantified subformulae and using the Skolemization operator Sk, it is easy to obtain (see Appendix A.3 of [15]) a sentence $\varphi'' := \forall x \forall y \chi \wedge \forall x \exists^{=1} y \chi'(x, y)$ with χ and $\chi'(x, y)$ quantifier-free so that $\mathrm{WFOMC}(\varphi', n, w, \bar{w}) = \mathrm{WFOMC}(\varphi'', n, w', \bar{w}')$, where w' and $\bar{w}'$ extend w and $\bar{w}$. If φ'' has nullary predicates, we eliminate them in the way discussed above. Thus, when studying WFOMC for FO^2 with a functionality axiom below, we begin with a sentence of the form $\forall x \forall y \varphi_1 \wedge \forall x \exists^{=1} y \varphi_2(x, y)$ where φ_1 and φ_2 are quantifier-free. We also assume, w.l.o.g., that the sentence contains at least one binary relation symbol and no symbols of arity greater than two. (These assumptions are justified in Appendix A.4 of [15].)

3 Counting for FO^2 with functionality

We now show that the symmetric weighted model counting problem for FO^2-sentences with a functionality axiom is in PTIME. As discussed in the preliminaries, it suffices to consider a formula

$$\Phi_0 := \forall x \forall y\, \varphi_0^{\forall}(x, y) \;\wedge\; \forall x \exists^{=1} y\, \varphi_0^{\exists}(x, y)$$

where $\varphi_0^{\forall}(x, y)$ and $\varphi_0^{\exists}(x, y)$ are quantifier-free and do not contain nullary relation symbols. Further assumptions justified in the preliminaries are that Φ_0 contains at least one binary relation symbol and no relation symbols of arity greater than two. From now on, we thus consider a fixed formula Φ_0 of the above form as well as fixed weight functions w and $\bar{w}$.

To simplify the constructions below, it would help if the subformula $\varphi_0^{\exists}(x, y)$ of Φ_0 was of the form $x \neq y \;\wedge\; \psi$ so that a witness for the existential quantifier would always be different from the point it is a witness to. However, there seems to be no obvious way to convert Φ_0 into the desired form while preserving weighted model counts. We thus use a conversion that does not preserve these counts and then show how to rectify this. Let

$$\begin{aligned} \Phi \;:=\; & \forall x \forall y \left(\varphi_0^{\forall}(x, y) \;\wedge\; \neg(x \neq y \wedge \varphi_0^{\exists}(x, x) \wedge \varphi_0^{\exists}(x, y))\right) \\ & \wedge\; \forall x \exists^{=1} y \Big(x \neq y \quad \wedge \Big(\quad (\varphi_0^{\exists}(x, x) \wedge Sy) \\ & \qquad\qquad \vee\; (\varphi_0^{\exists}(x, x) \wedge Sx \wedge Ty) \\ & \qquad\qquad \vee\; (\neg\varphi_0^{\exists}(x, x) \wedge \varphi_0^{\exists}(x, y)) \Big)\Big) \end{aligned}$$

where S and T are fresh unary predicates. Let $\mathcal{M}$ be the class of models (over $voc(\Phi)$) where S and T are interpreted to be distinct singletons. Slightly abusing notation, assume further that both w and $\bar{w}$ assign to both S and T the value 1.

The remainder of this section is devoted to showing how to compute $\mathrm{WFOMC}(\Phi, n, w, \bar{w}) \upharpoonright \mathcal{M}$. We note that the class

$$\mathcal{M}_1 := \{\, \mathfrak{M} \in \mathcal{M} \mid dom(\mathfrak{M}) = n \,\}$$

of models relevant to $\mathrm{WFOMC}(\Phi, n, w, \bar{w}) \upharpoonright \mathcal{M}$ can be obtained from the class $\mathcal{M}_0$ of models relevant to $\mathrm{WFOMC}(\Phi_0, n, w, \bar{w})$ by interpreting S and T as distinct singletons in all possible ways, so every model in $\mathcal{M}_0$ gives rise to $n(n-1)$ models in $\mathcal{M}_1$. It is thus easy to see that we get $\mathrm{WFOMC}(\Phi_0, n, w, \bar{w})$ from $\mathrm{WFOMC}(\Phi, n, w, \bar{w}) \upharpoonright \mathcal{M}$ by dividing by $n(n-1)$. (The case $n = 1$ is computed separately.)

We note that there seems to be no obvious way to modify Φ to additionally enforce S and T to be distinct singletons. While this property is expressible by a sentence of FO^2, adding such a sentence would destroy the intended syntactic structure of Φ. Note here that Lemma 2.2 does not in general produce an equivalent formula, so using it for modifying the required FO^2-sentence would not help.

3.1 Partitioning models

For simplicity, let $\Phi = \forall x \forall y\, \varphi^{\forall}(x, y) \wedge \forall x \exists^{=1} y\, \varphi^{\exists}(x, y)$, so $\varphi^{\forall}(x, y)$ and $\varphi^{\exists}(x, y)$ denote, respectively, the quantifier-free parts of the

$\forall\forall$-conjunct and $\forall\exists^{=1}$-conjunct of Φ. In the rest of Section 3, types and tables mean types and tables with respect to $voc(\Phi)$.

Now, recall from the preliminaries that a 2-type $\tau(x,y)$ is a conjunction $\alpha(x) \wedge \beta(x,y) \wedge \alpha'(y) \wedge x \neq y$ where β is a 2-table and α, α' are 1-types. We denote such a 2-type by $\alpha\beta\alpha'$. We call α the *first 1-type* and α' the *second 1-type* of $\tau(x,y)$ and denote these 1-types by $\tau(1)$ and $\tau(2)$. The 2-type $\tau(x,y)$ is *coherent* if

$$\tau(x,y) \models \varphi^{\forall}(x,y) \wedge \varphi^{\forall}(y,x) \wedge \varphi^{\forall}(x,x) \wedge \varphi^{\forall}(y,y).$$

A 1-type $\alpha(x)$ is *coherent* if $\alpha(x) \models \varphi^{\forall}(x,x)$. The *inverse* of a 2-type $\tau(x,y)$ is the 2-type $\tau'(x,y) \equiv \tau(y,x)$. A 2-type is *symmetric* if it is equal to its inverse.

The *witness* of an element u in a model $\mathfrak{M}$ of Φ is the unique element v such that $\mathfrak{M} \models \varphi^{\exists}(u,v)$. A 2-type $\tau(x,y)$ is *witnessing* if $\tau(x,y)$ is coherent and we have $\tau(x,y) \models \varphi^{\exists}(x,y)$. The 2-type $\tau(x,y)$ is *both ways witnessing* if both it and its inverse are witnessing; note that a both ways witnessing 2-type can be symmetric but does not have to. The set of all witnessing 2-types is denoted by Λ.

We next define the notions of a *block* and a *cell*. These are an essential part of the subsequent constructions. One central idea of our model counting strategy is to partition the domain of a model $\mathfrak{M}$ of Φ into blocks which are further partitioned into cells. A *block type* is simply a witnessing 2-type. The *block type of an element u* of $\mathfrak{M} \models \Phi$ is the unique witnessing 2-type $\tau(x,y)$ such that $\mathfrak{M} \models \tau(u,v)$, where v is the witness of u. The domain M of $\mathfrak{M}$ is partitioned by the family $(B^{\mathfrak{M}}_{\tau})_{\tau}$ where each set $B^{\mathfrak{M}}_{\tau} \subseteq M$ contains precisely the elements of $\mathfrak{M}$ with block type τ. Some of the sets $B^{\mathfrak{M}}_{\tau}$ can of course be empty. We call the sets $B^{\mathfrak{M}}_{\tau}$ the *blocks* of $\mathfrak{M}$ and refer to $B^{\mathfrak{M}}_{\tau}$ as the *block of type* τ. We fix a linear order $<$ over all block types and denote its reflexive variant by $\leq$.

Each block further partitions into *cells*. A *cell type* is a pair (σ,τ) of witnessing 2-types. For brevity, we denote cell types by $\sigma\tau$ instead of (σ,τ). The *cell type of an element u* in a model $\mathfrak{M} \models \Phi$ is the unique pair $\sigma\tau$ such that $u \in B^{\mathfrak{M}}_{\sigma}$ and $v \in B^{\mathfrak{M}}_{\tau}$, v the witness of u. Each block $B^{\mathfrak{M}}_{\sigma}$ is partitioned by the family $(C^{\mathfrak{M}}_{\sigma\tau})_{\tau}$ where each set $C^{\mathfrak{M}}_{\sigma\tau} \subseteq B^{\mathfrak{M}}_{\sigma}$ contains precisely the elements of $\mathfrak{M}$ that are of cell type $\sigma\tau$. Again, some of the sets $C^{\mathfrak{M}}_{\sigma\tau}$ can be empty. We call the sets $C^{\mathfrak{M}}_{\sigma\tau}$ the *cells* of $B^{\mathfrak{M}}_{\sigma}$ and refer to $C^{\mathfrak{M}}_{\sigma\tau}$ as the *cell of type* $\sigma\tau$.

3.2 The counting strategy

We now describe our strategy for computing WFOMC$(\Phi_0, n, w, \bar{w})$ informally. A formal treatment will be given later on. We first explain how to compute WFOMC$(\Phi, n, w, \bar{w})$ and then discuss how to get WFOMC$(\Phi, n, w, \bar{w}) \upharpoonright \mathcal{M}$ and WFOMC$(\Phi_0, n, w, \bar{w})$.

The strategy for computing WFOMC$(\Phi, n, w, \bar{w})$ is based on blocks and cells. We are interested in models of a given size n and with domain $n = \{0, \ldots, n-1\}$, so we let $\mathcal{M}^{\Phi}_{n}$ denote the set of all $voc(\Phi)$-models $\mathfrak{M}$ with domain n that satisfy Φ.

A *cell configuration* is a partition $(C_{\sigma\tau})_{\sigma\tau}$ of the set n where some sets can be empty. The *cell configuration of a model* $\mathfrak{M} \in \mathcal{M}^{\Phi}_{n}$ is the family $(C^{\mathfrak{M}}_{\sigma\tau})_{\sigma\tau}$ as defined in Section 3.1. For a cell configuration Γ, we use $\mathcal{M}^{\Phi}_{n,\Gamma}$ to denote the class of all models in $\mathcal{M}^{\Phi}_{n}$ that have cell configuration Γ. It is clear that the family $(\mathcal{M}^{\Phi}_{n,\Gamma})_{\Gamma}$, where Γ ranges over all cell configurations, partitions $\mathcal{M}^{\Phi}_{n}$ (though some sets $\mathcal{M}^{\Phi}_{n,\Gamma}$ can be empty). It would be convenient to iterate over cell configurations Γ and independently compute the weight of all models in each $\mathcal{M}^{\Phi}_{n,\Gamma}$, eventually summing up the computed weights. However, this option is ruled out since the number of cell configurations is exponential in n. Fortunately, it suffices to only know the *sizes* of cells rather than their concrete extensions.

Let $\sigma_1, \ldots, \sigma_k$ enumerate all block types. Then the sequence $\sigma_1\sigma_1, \sigma_1\sigma_2, \ldots, \sigma_k\sigma_k$ enumerates all cell types. A *multiplicity configuration* is a vector $(n_{\sigma_1\sigma_1}, n_{\sigma_1\sigma_2}, \ldots, n_{\sigma_k\sigma_k})$ where each $n_{\sigma_i\sigma_j}$ is a number in $\{0, \ldots, n\}$ and $n_{\sigma_1\sigma_1} + \cdots + n_{\sigma_k\sigma_k} = n$. The *multiplicity configuration of a model* $\mathfrak{M} \in \mathcal{M}^{\Phi}_{n}$ is obtained by letting each $n_{\sigma\tau}$ be the size of $C^{\mathfrak{M}}_{\sigma\tau}$. For a multiplicity configuration Δ, we use $\mathcal{M}^{\Phi}_{n,\Delta}$ to denote the class of all models from $\mathcal{M}^{\Phi}_{n}$ that have multiplicity configuration Δ. Clearly, the number of multiplicity configurations is polynomial in n, so we can iterate over them and—as we shall see—independently compute the weight of all models in each $\mathcal{M}^{\Phi}_{n,\Delta}$ in polynomial time.

Each cell configuration gives rise to a unique multiplicity configuration. Conversely, for every multiplicity configuration $\Delta = (n_{\sigma_1\sigma_1}, n_{\sigma_1\sigma_2}, \ldots, n_{\sigma_k\sigma_k})$, there are $\ell = \binom{n}{n_{\sigma_1\sigma_1}, n_{\sigma_1\sigma_2}, \ldots, n_{\sigma_k\sigma_k}}$ cell configurations giving rise to Δ. For any two such cell configurations Γ, Γ', the weight of $\mathcal{M}^{\Phi}_{n,\Gamma}$ (i.e., the sum of the weights of the models in $\mathcal{M}^{\Phi}_{n,\Gamma}$) is identical to the weight of $\mathcal{M}^{\Phi}_{n,\Gamma'}$. To obtain the weight of $\mathcal{M}^{\Phi}_{n,\Delta}$, it thus suffices to consider a single cell configuration Γ giving rise to Δ, compute the weight of $\mathcal{M}^{\Phi}_{n,\Gamma}$ and multiply by ℓ.

We now briefly describe how to compute the number of models in $\mathcal{M}^{\Phi}_{n,\Gamma}$, ignoring weights. With easy modifications, the approach will ultimately also give the weight of $\mathcal{M}^{\Phi}_{n,\Gamma}$. Although our algorithm is not going to explicitly construct the models in $\mathcal{M}^{\Phi}_{n,\Gamma}$, to describe how the number of those models is counted, we simultaneously consider how we could construct all of them.

Let $(B_\sigma)_\sigma$ be the *block configuration* that corresponds to the cell configuration $\Gamma = (C_{\sigma\tau})_{\sigma\tau}$, that is, $B_\sigma = \bigcup_\tau C_{\sigma\tau}$ for each block type σ. As the domain is fixed to be n, we consider all possible ways to assign 1-types to the elements of n and 2-tables to pairs of distinct elements such that we realize the cell configuration Γ. There is no freedom for the 1-types: if $u \in B_\sigma$, then we must assign the 1-type $\sigma(1)$ to u. To assign 2-tables, we consider each pair of blocks (B_σ, B_τ) with $\sigma \leq \tau$ independently, identifying each possible way to simultaneously assign 2-tables to pairs in $B_\sigma \times B_\tau$. (When $\sigma = \tau$, we must be careful to (1) consider only pairs (u,v) of *distinct* elements and (2) to assign a 2-table to only one of $(u,v), (v,u)$.) It is important to understand that in B_σ, there is exactly one cell, namely $C_{\sigma\tau}$, whose elements require a witness from B_τ. Similarly, in B_τ, it is precisely the elements of $C_{\tau\sigma}$ that require a witness in B_σ. Since witnesses are unique, we start by identifying the ways to simultaneously define functions $f : C_{\sigma\tau} \to B_\tau$ and $g : C_{\tau\sigma} \to B_\sigma$ that determine the witnesses. It then remains to count the number of ways to assign 2-types to the remaining edges that are witnessing in neither direction. This is easy—as long as we know the number N of these remaining edges—since each edge realizes the 1-type $\sigma(1)$ at the one end and $\tau(2)$ at the other. We use a look-up table to find the number of 2-tables that are 'compatible' with this. The number N depends on how many pairs in $B_\sigma \times B_\tau$ and $B_\tau \times B_\sigma$ belong to the functions that determine the witnesses, but N will nevertheless be easy to determine, as we shall see.

The precise arithmetic formulae for counting the number of ways to assign 2-tables to all elements from $B_\sigma \times B_\tau$ are given in Section 3.3. There are several cases that need to be distinguished. We now briefly look at the most important cases informally.

We start with the case $\sigma = \tau$, that is, the two blocks B_σ, B_τ are in fact the same single block, and we aim to assign 2-tables within that block. Then exactly the elements from the cell $C_{\sigma\sigma}$ require a witness in B_σ itself. If σ is not both ways witnessing, then $C_{\sigma\sigma}$ will be the domain of an anti-involutive function $C_{\sigma\sigma} \to B_\sigma$ that determines a witness in B_σ for each element in $C_{\sigma\sigma}$. If σ is both ways witnessing and its own inverse, this function is involutive. The case where σ is both ways witnessing but not its own inverse is pathological in the sense that there are then no valid ways to assign 2-tables unless $C_{\sigma\sigma}$ is empty. To sum up, in each case, the core task in designing the desired arithmetic formula is thus to count the number of suitable anti-involutive or involutive functions.

Now consider the case where $\sigma \neq \tau$ and thus B_σ and B_τ are different blocks. Here again several subcases arise based on whether σ and τ are both ways witnessing. The most interesting case is where neither σ nor τ is both ways witnessing. We then need to count the ways of finding two functions $f : C_{\sigma\tau} \to B_\tau$ and $g : C_{\tau\sigma} \to B_\sigma$ that are nowhere inverses of each other. In the case where σ and τ are both ways witnessing and inverses of each other, we need to count the number of perfect matchings between the sets $C_{\sigma\tau}$ and $C_{\tau\sigma}$. The case where at least one of the witness types, say σ, is both ways witnessing, but σ and τ are not inverses of each other, is again pathological.

Implementing the above ideas, we will show how to obtain, for any pair of blocks B_σ, B_τ, where we have $\sigma \leq \tau$, a function $M_{\sigma\tau}(n_\sigma, n_{\sigma\tau}, n_\tau, n_{\tau\sigma})$ that counts the 'weighted number of ways' to connect the blocks B_σ and B_τ with 2-tables, when given the sizes n_σ and n_τ of the blocks as well as the sizes $n_{\sigma\tau}$ and $n_{\tau\sigma}$ of the cells $C_{\sigma\tau} \subseteq B_\sigma$ and $C_{\tau\sigma} \subseteq B_\tau$; we note that while this fixes the intuitive interpretation of $M_{\sigma\tau}(n_\sigma, n_{\sigma\tau}, n_\tau, n_{\tau\sigma})$, the function $M_{\sigma\tau}$ will become formally defined in terms of arithmetic operations in Section 3.4. (Furthermore, for extra clarity, we provide in Appendix B.1 of [15] a more detailed description of what the weighted number of ways to connect B_σ and B_τ with 2-tables means.)

Recall that Λ is the set of all block types and note that $n_\sigma = \sum_{\sigma' \in \Lambda} n_{\sigma\sigma'}$ and likewise for n_τ, so n_σ and n_τ are determined by the sizes of all cells in the blocks B_σ and B_τ. With the aim of achieving notational uniformity, we can thus replace $M_{\sigma\tau}$ by a function

$$N_{\sigma\tau}(n_{\sigma_1\sigma_1}, n_{\sigma_1\sigma_2}, \ldots, n_{\sigma_k\sigma_k}) \tag{2}$$

that outputs $M_{\sigma\tau}(n_\sigma, n_{\sigma\tau}, n_\tau, n_{\tau\sigma})$ but has a full multiplicity type as an input. Noting that the weight functions w and $\bar{w}$ give rise to the weight $w_\alpha := \langle w, \bar{w}\rangle(\alpha)$ of each 1-type α, we now observe that we can compute WFOMC$(\Phi, n, w, \bar{w})$ by the function

$$\mathcal{U}(n) := \sum_{n_{\sigma_1\sigma_1}+n_{\sigma_1\sigma_2}+\cdots+n_{\sigma_k\sigma_k}=n} \left(\binom{n}{n_{\sigma_1\sigma_1}, n_{\sigma_1\sigma_2}, \ldots, n_{\sigma_k\sigma_k}} \cdot \Big(\prod_{\sigma \in \Lambda} (w_{\sigma(1)})^{n_\sigma} \Big) \prod_{\sigma,\, \tau\, \in \Lambda} N_{\sigma\tau}(n_{\sigma_1\sigma_1}, n_{\sigma_1\sigma_2}, \ldots, n_{\sigma_k\sigma_k}) \right). \tag{3}$$

Recall, however, that we aim to compute WFOMC$(\Phi, n, w, \bar{w}) \upharpoonright \mathcal{M}$ rather than WFOMC$(\Phi, n, w, \bar{w})$. And eventually we want to compute WFOMC$(\Phi_0, n, w, \bar{w})$, which can be obtained simply by dividing WFOMC$(\Phi, n, w, \bar{w}) \upharpoonright \mathcal{M}$ by $n(n-1)$. In order to get from WFOMC$(\Phi, n, w, \bar{w})$ to WFOMC$(\Phi, n, w, \bar{w}) \upharpoonright \mathcal{M}$, we need to discard weights contributed by models where S and T are not interpreted as non-overlapping singletons. This is easy: we only need to discard multiplicity configurations $(n_{\sigma_1\sigma_1}, n_{\sigma_1\sigma_2}, \ldots, n_{\sigma_k\sigma_k})$ that do not make S and T distinct singletons. Let $\langle n \rangle$ be the set of multiplicity configurations with the undesired ones excluded. Summing up, WFOMC$(\Phi_0, n, w, \bar{w})$ can thus be computed by the function

$$\mathcal{W}(n) = \frac{1}{n(n-1)} \cdot \sum_{(n_{\sigma_1\sigma_1}, n_{\sigma_1\sigma_2}, \ldots, n_{\sigma_k\sigma_k}) \in \langle n \rangle} \left(\binom{n}{n_{\sigma_1\sigma_1}, n_{\sigma_1\sigma_2}, \ldots, n_{\sigma_k\sigma_k}} \cdot \Big(\prod_{\sigma \in \Lambda} (w_{\sigma(1)})^{n_\sigma} \Big) \prod_{\sigma,\, \tau\, \in \Lambda} N_{\sigma\tau}(n_{\sigma_1\sigma_1}, n_{\sigma_1\sigma_2}, \ldots, n_{\sigma_k\sigma_k}) \right). \tag{4}$$

In the next Section 3.3 we deal with the combinatorics for defining the functions $N_{\sigma\tau}$. The actual functions $N_{\sigma\tau}$ are then specified in Section 3.4 where we conclude our argument.

3.3 The relevant combinatorics

Let $k \in \mathbb{N}$. The following equation is well known.

$$\sum_{i=0}^{i=k} (-1)^i \binom{k}{i} = \begin{cases} 0 \text{ if } k \neq 0 \\ 1 \text{ if } k = 0. \end{cases} \tag{5}$$

On the intuitive level, the *alternating sum* on the left hand side of the equation relates directly to the *inclusion-exclusion* principle. We shall make frequent use of this equation in the constructions below.

The first result of this section, Proposition 3.1 below, will ultimately help us in counting the number of ways to connect a block *to itself* with 2-tables. However, the result is interesting in its own right and thus we formulate it abstractly, like most results in this section, without reference to 2-types or other logic-related notions.

Recall that a unary function is *anti-involutive* if $f(f(x)) \neq x$ for all $x \in dom(f)$. Note that this implies $f(x) \neq x$ for all $x \in dom(f)$, i.e., f is *fixed point free.*

Proposition 3.1. *Let n and $m \leq n$ be nonnegative integers. The number of anti-involutive functions $m \to n$ is*

$$I(m,n) := \sum_{i=0}^{i=\lfloor m/2 \rfloor} (-1)^i (n-1)^{m-2i} \binom{m}{2i} \frac{(2i)!}{2^i (i!)}. \tag{6}$$

Proof. We first note that for a nonnegative integer i, there are $\binom{2i}{2,\ldots,2}\frac{1}{i!}$ ways to partition $2i$ elements into doubletons, where 2 is written i times in the bottom row. Writing the multinomial coefficient $\binom{2i}{2,\ldots,2}$ open, we see that $\binom{2i}{2,\ldots,2}\frac{1}{i!} = \frac{(2i)!}{2^i(i!)}$.

Now, for a fixed point free function f, if $f(f(x)) = x$ for some x, then we call the doubleton $\{x, f(x)\}$ a *symmetric pair* of f. A *fixed point free function $f : m \to n$ with i labelled symmetric pairs* is a pair (f, L) where $f : m \to n$ is a fixed point free function and L is a set of exactly i symmetric pairs of f. Note that f may have other symmetric pairs outside L, so L only distinguishes i specially labelled symmetric pairs.

It is easy to see that the number of fixed point free functions $m \to n$ with i labelled symmetric pairs is given by

$$(n-1)^{m-2i} \binom{m}{2i} \frac{(2i)!}{2^i(i!)}. \tag{7}$$

Therefore Equation 6 has the following intuitive interpretation. The equation first counts—when i is zero—all fixed point free functions $m \to n$ without any *labelled* symmetric pairs; unlabelled symmetric pairs are allowed. Then, when $i = 1$, the equation subtracts the number of fixed point free functions $m \to n$ with one *labelled* symmetric pair. Then, with $i = 2$ the equation adds the the number of fixed point free functions $m \to n$ with two *labelled* symmetric pairs, and so on, all the way to $i = \lfloor m/2 \rfloor$.

Now, fix a single fixed point free function $f : m \to n$ with *exactly* j symmetric pairs. Labelling $k \le j$ of the j symmetric pairs can be done in $\binom{j}{k}$ ways. Thus f gets counted in Equation 6 precisely $S(j) := \binom{j}{0} - \binom{j}{1} + \binom{j}{2} - \cdots * \binom{j}{j}$ times, where $*$ is $+$ if j is even and $-$ if j is odd. By Equation 5, $S(j)$ is 0 when $j \neq 0$ and 1 when $j = 0$. Thus f gets counted zero times if $j \neq 0$ and once if $j = 0$. □

Proposition 3.1 will be used for counting functions that find a witness for each element of a cell C of size m from a block $B \supseteq C$ of size n. However, we also need to count the ways of assigning non-witnessing 2-tables to the remaining edges inside B. The next two results, Lemma 3.2 and Proposition 3.3, will help in this.

Let G be an undirected graph with the set V of vertices and E of edges. A *labelling* of G with k *symmetric colours* and ℓ *directed colours* is a pair of functions (s, d) such that

1. s maps some set $U \subseteq E$ into $[k]$, not necessarily surjectively,
2. d maps the complement $E \setminus U$ of U into $[\ell] \times V$ such that each edge $e \in E \setminus U$ gets mapped to a pair (i, u) where $u \in e$. Intuitively, d picks a colour in $[\ell]$ and an *orientation* for e. It is *not* required that each $i \in [\ell]$ gets assigned to some edge.

The colour $j \in [\ell]$ is said to *define a function* if the relation $\{ (u, v) \mid \{u, v\} \in E \setminus U,\ d(\{u, v\}) = (j, v) \}$ is a function.

Rather than counting labellings of graphs, we need to count *weighted labellings*: a weighted labelling of a graph G with k symmetric and ℓ directed colours is a triple

$$W = ((s, d), (w_1, \dots, w_k), (x_1, \dots x_\ell))$$

such that (s, d) is a labelling of G and $w_1, \dots, w_k$ are weights of the symmetric colours $1, \dots, k$ and $x_1, \dots, x_\ell$ weights of the directed colours $1, \dots, \ell$. (Here for example 1 is called both a directed and symmetric colour. This will pose no problem.) The *total weight* t_W of the weighted labelling W is the product of the weights assigned to the edges of G. The *weighted number of labellings* of G with k symmetric and ℓ directed colours with weights $w_1, \dots, w_k$ and $x_1, \dots, x_\ell$ is the sum of the total weights t_W of all weighted labellings $W = ((s, d), (w_1, \dots, w_k), (x_1, \dots x_\ell))$ of G.

The following is easy to prove (see Appendix B.2 of [15]).

Lemma 3.2. *The function*

$$L_{k,\ell}(N, w_1, \dots, w_k, x_1, \dots, x_\ell) := \sum_{i_1+\cdots+i_k+j_1+\cdots+j_\ell = N} \Bigg(\binom{N}{i_1, \dots, i_k, j_1, \dots j_\ell} \cdot 2^{j_1+\cdots+j_\ell} \Big(\prod_{p \in [k]} (w_p)^{i_p} \Big) \Big(\prod_{q \in [\ell]} (x_q)^{j_q} \Big) \Bigg) \quad (8)$$

gives the weighted number of labellings of an arbitrary N-edge graph with k symmetric and ℓ directed colours with weights $w_1, \dots, w_k$ and $x_1, \dots, x_\ell$. At least one of k, ℓ is assumed nonzero here. The first (resp. second) product on the bottom row outputs 1 if $k = 0$ (resp. $\ell = 0$).

We also define $L_{0,0}(N) := 0$ for $N > 0$ and $L_{0,0}(0) := 1$, and also $L_{k,\ell}(m, w_1, \dots, w_k, x_1, \dots, x_\ell) := 0$ for all negative integers m. The following is easy to prove (see Appendix B.3 of [15]).

Proposition 3.3. *Let n and $m \le n$ be nonnegative integers, and let $w_1, \dots, w_k$ and $x_1, \dots, x_\ell, y$ be weights for k symmetric and $\ell + 1$ directed colours. The function*

$$J_{k,\ell+1}(m, n, w_1, \dots, w_k, x_1, \dots, x_l, y) := I(m, n) \cdot y^m \cdot L_{k,\ell}\Big(\binom{n}{2} - m,\ w_1, \dots, w_k, x_1, \dots, x_\ell\Big) \quad (9)$$

gives the weighted number of labellings of the complete n-element graph with k symmetric and $\ell + 1$ directed colours with the above weights such that the edges of colour $\ell + 1$ define an anti-involutive function $m \to n$.

The following result will ultimately help us in counting the ways of connecting two *different* blocks to each other with 2-tables.

Proposition 3.4. *Let $A \neq \emptyset$ and $B \neq \emptyset$ be disjoint finite sets, $|A| = M$ and $|B| = N$. Let $A_m \subseteq A$ and $B_n \subseteq B$ be sets of sizes m and n, respectively. There exist*

$$K(m, M, n, N) := \sum_{i=0}^{i=\min(m,n)} (-1)^i \binom{m}{i}\binom{n}{i}\Big(i! \cdot M^{(n-i)} \cdot N^{(m-i)}\Big) \quad (10)$$

ways to define two functions $f : A_m \to B$ and $g : B_n \to A$ that are nowhere inverses of each other.

Proof. Fix some $i \le min(m, n)$, and fix two sets $A_i \subseteq A_m$ and $B_i \subseteq B_n$, both of size i. There exist $\big(i! \cdot M^{(n-i)} \cdot N^{(m-i)}\big)$ ways to define a pair of functions $f : A_m \to B$ and $g : B_n \to A$ such that $f \restriction A_i$ and $g \restriction B_i$ are bijections and inverses of each other; here $i!$ is the number of ways the two functions can be defined in restriction to A_i and B_i so that they become inverses of each other over A_i and B_i. (Note that f and g can be inverses elsewhere too.) Thus

$$\binom{m}{i}\binom{n}{i}\Big(i! \cdot M^{(n-i)} \cdot N^{(m-i)}\Big)$$

gives the number of tuples (f, g, A', B') such that $f : A_m \to B$ and $g : B_n \to A$ are functions and $A' \subseteq A_m$ and $B' \subseteq B_n$ sets of size i such that $f \restriction A'$ and $g \restriction B'$ are inverses of each other.

Now, fix two sets $A_j \subseteq A_m$ and $B_j \subseteq B_n$ of size j both. Fix two functions $f : A_m \to B$ and $g : B_n \to A$ that are inverses of each other on A_j and B_j and *nowhere else*. Thus the pair f, g is counted in the alternating sum of Equation 10 exactly $S(j) := \binom{j}{0} - \binom{j}{1} + \binom{j}{2} - \cdots * \binom{j}{j}$ times, where $*$ is $+$ if j is even and $-$ otherwise. By Equation 5, $S(j)$ is zero when $j \neq 0$ and one when $j = 0$. Thus the pair f, g gets counted zero times if $j \neq 0$ and otherwise once. □

We also define $K(m, M, n, N) := 0$ for any $m \le M$ and $n \le N$ with $M = 0 \neq n$ or $N = 0 \neq m$. Furthermore, we define $K(0, 0, 0, N) = K(0, M, 0, 0) = 1$ for all $M, N \in \mathbb{N}$.

The next result, Proposition 3.5, extends Proposition 3.4 so that also the non-witnessing edges will be taken into account. To formulate the result, we define that for disjoint finite sets A and B, the *complete bipartite graph on $A \times B$* is the undirected bipartite graph with the set $\{ \{a, b\} \mid a \in A,\ b \in B \}$ of edges.

Proposition 3.5. *Let A and B be finite disjoint sets, $|A| = M$ and $|B| = N$. Let $A_m \subseteq A$ and $B_n \subseteq B$ be sets of sizes m and n, respectively. Let $w_1, \dots, w_k$ and $x_1, \dots, x_\ell, y, z$ be weights. The function*

$$P_{k,\ell+2}(m, M, n, N, w_1, \dots, w_k, x_1, \dots, x_\ell, y, z) := K(m, M, n, N) \cdot y^m z^n \cdot L_{k,\ell}(MN - m - n,\ w_1, \dots, w_k, x_1, \dots, x_\ell) \quad (11)$$

gives the weighted number of labellings of the complete bipartite graph on $A \times B$ with k symmetric and $\ell + 2$ directed colours with weights $w_1, \ldots, w_k$ and $x_1, \ldots, x_\ell, y, z$ such that the directed colours $\ell + 1$ and $\ell + 2$ define, respectively, functions $f : A_m \to B$ and $g : B_n \to A$ that are nowhere inverses of each other.

Proof. The relatively easy proof is given in Appendix B.4 of [15]. □

The results so far in this section provide us with ways of counting in cases where witnesses are found via 2-types that are not both ways witnessing. We now deal with the remaining cases.

Recall that $n!!$ denotes the standard *double factorial* operation defined such that for example $7!! = 7 \cdot 5 \cdot 3 \cdot 1$ and $8!! = 8 \cdot 6 \cdot 4 \cdot 2$. We define the function $F : \mathbb{N} \to \mathbb{N}$ such that $F(0) = 1$ and for all $m \in \mathbb{Z}_+$, we have $F(m) = (m-1)!!$ if m is even and $F(m) = 0$ otherwise. It is well known and easy to show that $F(m)$ is the number of *perfect matchings* of the complete graph G with the set m of vertices, i.e., the number of 1-*factors* of a complete graph of order m (and with the set m of vertices). A *perfect matching of the set* m means a perfect matching of the complete graph with vertex set m. The following is easy to prove (see Appendix B.5 of [15]).

Proposition 3.6. *Let n and $m \leq n$ be nonnegative integers, and let $w_1, \ldots, w_k, y$ and $x_1, \ldots, x_\ell$ be weights. The function*

$$S_{k+1,\ell}(m, n, w_1, \ldots, w_k, y, x_1, \ldots, x_\ell) :=$$
$$F(m) \cdot y^{m/2} \cdot L_{k,\ell}\Big(\binom{n}{2} - \lfloor m/2 \rfloor,\ w_1, \ldots, w_k, x_1, \ldots, x_\ell\Big) \quad (12)$$

gives the weighted number of labellings of the complete graph with the set n of vertices with $k + 1$ symmetric and ℓ directed colours with weights $w_1, \ldots, w_k, y$ and $x_1, \ldots, x_\ell$ such that the edges of the symmetric colour $k + 1$ define a perfect matching of the set $m \subseteq n$.

Let $F' : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be the function such that $F'(n, m) = n!$ if $n = m$ and $F'(n, m) = 0$ otherwise. A perfect matching between two disjoint sets S and T means a perfect matching of the complete bipartite graph on $S \times T$. The following is immediate.

Proposition 3.7. *Let A and B be finite disjoint finite sets, $|A| = M$ and $|B| = N$. Let $A_m \subseteq A$ and $B_n \subseteq B$ be sets of sizes m and n, respectively. The function*

$$T_{k+1,\ell}(m, M, n, N, w_1, \ldots, w_k, y, x_1, \ldots, x_\ell) :=$$
$$F'(n, m) \cdot y^n \cdot L_{k,\ell}(MN - n,\ w_1, \ldots, w_k, x_1, \ldots, x_\ell) \quad (13)$$

gives the weighted number of labellings of the complete bipartite graph on $A \times B$ with $k + 1$ symmetric and ℓ directed colours with weights $w_1, \ldots, w_k, y$ and $x_1, \ldots, x_\ell$ such that the symmetric colour $k + 1$ defines a perfect matching between A_m and B_n.

3.4 Defining the functions $N_{\sigma\tau}$

We now discuss how the functions $N_{\sigma\tau}$ are defined for all pairs $\sigma\tau$ of block types, thereby completing the definition of Equation 4.

Fix a pair $\sigma\tau$ of block types. Let y and z, respectively, be the weights of the 2-tables of the 2-types σ and τ. Let $w_1, \ldots, w_k$ (respectively, $x_1, \ldots, x_\ell$) enumerate the weights of the symmetric (resp., unsymmetric) 2-tables β that can connect the block B_σ to the block B_τ so that neither the resulting 2-type $\sigma(1)\beta\tau(1)$ nor its inverse is witnessing, and furthermore, $\sigma(1)\beta\tau(1)$ is coherent. If $\sigma = \tau$, these are the weights of the coherent 2-tables that can connect a point in block B_σ to another point in the same block so that the resulting 2-type is not witnessing in either direction.

We next consider different cases depending on how σ and τ relate to each other. We let $\overline{n}$ denote the input tuple to $N_{\sigma\tau}$, so $\overline{n}$ contains the multiplicities $n_{\sigma'\sigma''}$ of all cell types $\sigma'\sigma''$. For a witness 2-type σ', we let $n_{\sigma'}$ abbreviate the sum $\sum_{\sigma'' \in \Lambda} n_{\sigma'\sigma''}$ (recall Λ is the set of all block types). The witness 2-type σ' is *compatible* with a witness 2-type σ'' if $\sigma'(2) = \sigma''(1)$.

Case 1. We assume that **1.a)** $\sigma \neq \tau$; **1.b)** σ and τ are *compatible* with each other; **1.c)** neither σ nor τ is a both ways witnessing 2-type. By Proposition 3.5, the weight contributed by all the edges from B_σ to B_τ is thus given by

$$N_{\sigma\tau}(\overline{n}) := P_{k,\ell+2}(n_{\sigma\tau}, n_\sigma, n_{\tau\sigma}, n_\tau, w_1, \ldots, w_k, x_1, \ldots, x_\ell, y, z).$$

which defines $N_{\sigma\tau}$ under these particular assumptions.

The remaining cases are similar but use different functions defined in the previous section. For example, when $\sigma = \tau$ and σ is not two-ways witnessing, we use the function $J_{\ell,k+1}$ from Equation 9 in Proposition 3.3; see the Appendix B.6 of [15] (Case 4). All the remaining cases are also discussed in that appendix. By inspecting the operations of Equation 4, we conclude the following.

Theorem 3.8. *The weighted model counting problem of each two-variable logic sentence with a functionality axiom is in* PTIME.

4 Weighted model counting for U_1

In this section we prove that WFOMC is in PTIME for each sentence of U_1. To that end, we first establish the same result for SU_1, stated as Lemma 4.5 below. We follow a proof strategy that makes explicit how the syntactic restrictions of SU_1 naturally lead to polynomial time model counting. We then provide a reduction from U_1 to SU_1.

4.1 Weighted model counting for SU_1

Let $\psi(x_1, \ldots, x_k)$ be a quantifier-free first-order formula, and let $\ell \leq k$ be a positive integer. Let F denote the set of all surjections $[k] \to [\ell]$. The conjunction $\bigwedge\{\, \psi(x_{f(1)}, \ldots, x_{f(k)}) \mid f \in F \,\}$ is called the ℓ-*surjective image* of ψ.

Definition 4.1. Let φ be a conjunction of $\forall^*$-sentences of FO (These need not be sentences of U_1 or SU_1.) We now define the *surjective completion* $sur(\varphi)$ of φ by modifying φ as follows.

1.) Let k be the maximum width of the $\forall^*$-conjuncts of φ. We modify φ so that for all $i \in [k]$, there exists a conjunct of width i. This can be ensured by adding dummy conjuncts, if necessary. We let φ' denote the resulting sentence.

2.) We merge the conjuncts of φ' with the same width, so that for example $\forall x \forall y \psi(x, y) \wedge \forall x \forall y \chi(x, y)$ would become $\forall x \forall y(\psi(x, y) \wedge \chi(x, y))$. Thus the resulting formula φ'' is a conjunction of $\forall^*$-sentences so that no two conjuncts have the same width.

3.) Define $\varphi''_k := \varphi''$ where k is the maximum width of the $\forall^*$-sentences of φ''. Inductively, let $1 \leq \ell < k$ and assume we have defined a sentence $\varphi''_{\ell+1} = \chi_1 \wedge \cdots \wedge \chi_k$ where each χ_i is an $\forall^*$-sentence of width i. Let $\psi_{\ell+1}$ and ψ_ℓ be the matrices of $\chi_{\ell+1}$ and χ_ℓ, so we have

$$\chi_{\ell+1} = \forall x_1 \ldots \forall x_{\ell+1}\, \psi_{\ell+1}(x_1, \ldots, x_{\ell+1}),$$
$$\chi_\ell \quad = \forall x_1 \ldots \forall x_\ell\, \psi_\ell(x_1, \ldots, x_\ell).$$

Let ψ'_ℓ denote the ℓ-surjective image of $\psi_{\ell+1}$. Replace the conjunct χ_ℓ of $\varphi''_{\ell+1}$ by $\forall x_1 \ldots \forall x_\ell(\psi_\ell \wedge \psi'_\ell)$. Define φ''_ℓ to be the resulting modification of $\varphi''_{\ell+1}$. Define $sur(\varphi)$ to be the formula φ''_1.

Let $\varphi := \forall x_1 \ldots \forall x_k \psi$ be an $\forall^*$-sentence. We let $\mathit{diff}(\varphi)$ denote the sentence $\forall x_1 \ldots \forall x_k(\mathit{diff}(x_1, \ldots, x_k) \rightarrow \psi)$, letting $\mathit{diff}(x_1) := \top$. For a conjunction $\varphi' := \varphi_1 \wedge \cdots \wedge \varphi_k$ of $\forall^*$-sentences, we define $\mathit{diff}(\varphi') := \mathit{diff}(\varphi_1) \wedge \cdots \wedge \mathit{diff}(\varphi_k)$.

Lemma 4.2. *We have $\varphi \equiv \mathit{diff}(\mathit{sur}(\varphi))$ for any conjunction φ of first-order $\forall^*$-sentences.*

Proof. Clearly $\varphi \equiv \mathit{sur}(\varphi)$. Also $\mathit{sur}(\varphi) \equiv \mathit{diff}(\mathit{sur}(\varphi))$, as sur is based on steps where the surjective image of a matrix is pushed to be part of the matrix of a formula with one variable less. □

As discussed in the preliminaries, to prove that the weighted model counting problem of SU_1-sentences is in PTIME, it suffices to show this for conjunctions of $\forall^*$-sentences of SU_1 of the type $\varphi' = \forall x_1 \psi_1' \wedge \cdots \wedge \forall x_1 \ldots \forall x_p \psi_p'$ where each ψ_i' is quantifier-free. Other assumptions justified in the preliminaries are that φ' contains no nullary atoms; p is equal to the greatest arity of the symbols in $\mathit{voc}(\varphi')$; and $p \geq 2$. By Lemma 4.2, φ' is equivalent to $\varphi'' := \mathit{diff}(\mathit{sur}(\varphi'))$. We remove the conjunct of width 1 from φ'' and integrate it to the conjunct of width 2, so if

$$\varphi'' = \forall x_1\, \chi_1(x_1) \wedge \forall x_1 \forall x_2 \Big(\mathit{diff}(x_1, x_2) \rightarrow \chi_2(x_1, x_2)\Big) \wedge \Phi,$$

we replace φ'' by

$$\varphi := \forall x_1 \forall x_2 \Big(\mathit{diff}(x_1, x_2) \rightarrow (\chi_1(x_1) \wedge \chi_2(x_1, x_2))\Big) \wedge \Phi.$$

(We ignore the case with a one-element domain as we can simply store and return the answer in that case.) For the remainder of Section 4.1, we fix the obtained sentence φ and weight functions w and $\bar{w}$ that assign weights to each symbol R in the vocabulary η of φ; our aim is to compute $\mathrm{WFOMC}(\varphi, n, w, \bar{w})$. We let

$$\varphi \;=\; \forall x_1 \forall x_2\, \psi_2 \;\wedge\; \cdots \;\wedge\; \forall x_1 \ldots \forall x_p\, \psi_p, \tag{14}$$

so the individual matrices are denoted by ψ_i. We denote each conjunct $\forall x_1 \ldots \forall x_k\, \psi_k$ by φ_k. The next two lemmas are crucial for computing $\mathrm{WFOMC}(\varphi, n, w, \bar{w})$ in polynomial time.

Lemma 4.3. *$\mathfrak{M} \models \varphi$ iff for all $k \in \{2, \ldots, p\}$, we have $\mathfrak{M}_k \models \varphi_k$ for every k-element submodel $\mathfrak{M}_k$ of $\mathfrak{M}$.*

Proof. The first implication is immediate since universal sentences are preserved under taking submodels. For the converse implication, assume that for all $k \in \{2, \ldots, p\}$, $\mathfrak{M}_k \models \varphi_k$ for all submodels $\mathfrak{M}_k$ of $\mathfrak{M}$ of size k. Assume that $\mathfrak{M} \not\models \varphi$. Thus $\mathfrak{M} \not\models \varphi_k$ for some k. The matrix ψ_k of φ_k is of the type $\mathit{diff}(x_1, \ldots x_k) \rightarrow \psi$, so there exists some k-element submodel $\mathfrak{M}_k$ of $\mathfrak{M}$ with domain $\{u_1, \ldots, u_k\}$ such that $\mathfrak{M}_k \not\models \psi_k(u_1, \ldots, u_k)$. This is a contradiction, so $\mathfrak{M} \models \varphi$. □

Let $\mathfrak{M}$ and $\mathfrak{M}'$ be η-models such that $\mathfrak{M}'$ is obtained by changing exactly one fact of span size k from positive to negative or vice versa. Let S be the k-element set spanned by that fact. Then $\mathfrak{M}$ and $\mathfrak{M}'$ are *S-variants* of each other.

Lemma 4.4. *Let $\mathfrak{M}$ and $\mathfrak{M}'$ be S-variants of each other, $|S| > 1$. Let $U \neq S$ be a set of elements of $\mathfrak{M}$ such that $|U| = m > 1$. Let $\mathfrak{M}_U$ and $\mathfrak{M}_U'$ be the submodels of $\mathfrak{M}$ and $\mathfrak{M}'$ induced by U. Then $\mathfrak{M}_U \models \varphi_m$ iff $\mathfrak{M}_U' \models \varphi_m$.*

Proof. Firstly, if the formula $\varphi_m = \forall x_1 \ldots \forall x_m\, \psi_m$ contains atoms of arity two or more, then, by the syntactic restrictions of SU_1, each of those atoms mentions exactly all of the variables $x_1, \ldots x_m$. Secondly, ψ_m is of the form $\mathit{diff}(x_1, \ldots, x_m) \rightarrow \psi$. □

Lemma 4.5. *The weighted model counting problem for each* SU_1*-sentence is in* PTIME.

Proof. As discussed above, we prove the claim for the sentence φ we have fixed. Let T be the set of 1-types over the vocabulary η of φ. Fix an ordering of T and let $\alpha_1, \ldots, \alpha_\ell$ enumerate T in that order. For a positive integer $k = \{0, \ldots, k-1\}$, a function $f : k \rightarrow T$ is a *type assignment* over k. Two type assignments $f : k \rightarrow T$ and $g : k \rightarrow T$ are said to have the *same multiplicity*, if for each $\alpha \in T$, the functions f and g map the same number of elements in k to α.

For a type assignment $f : k \rightarrow T$, let $\mathcal{M}_{f,k}$ be the set of all η-models $\mathfrak{M}$ such that the following conditions hold.

1. The domain of $\mathfrak{M}$ is $k = \{0, \ldots, k-1\}$, and the size of the span of each positive fact $Ru_1 \ldots u_m$ of $\mathfrak{M}$ is either 1 or k, i.e., each positive fact either spans a single domain element or all of the domain elements of $\mathfrak{M}$.
2. For each $m \in \{0, \ldots, k-1\}$, we have $\mathfrak{M} \models \alpha_{f(m)}(m)$.
3. $\mathfrak{M} \models \varphi_k$.

Recalling the relativised weight function W_k from the preliminaries, we define the *local weight* $lw(\varphi_k, f)$ of φ_k with respect to a type assignment $f : k \rightarrow T$ so that

$$lw(\varphi_k, f) \;:=\; \sum_{\mathfrak{M} \,\in\, \mathcal{M}_{f,k}} W_k(\mathfrak{M}, w, \bar{w}).$$

Thus $lw(\varphi_k, f)$ could be characterized as giving the weighted number of models of φ_k with domain k and with 1-types distributed according to f so that only those positive and negative facts are counted that have span k. Clearly $lw(\varphi, f) = lw(\varphi, g)$ for any $g : k \rightarrow T$ that has the same multiplicity as f, so only the number of realizations of the 1-types matters rather than the concrete realizations. Therefore we define, for any nonnegative integers $k_1, \ldots, k_\ell$ such that $k_1 + \cdots + k_\ell = k$, that $lw(\varphi_k, (k_1, \ldots, k_\ell)) := lw(\varphi_k, h)$, where $h : k \rightarrow T$ is a type assignment that maps, for each $i \in [\ell]$, precisely k_i elements of k to α_i. Note that there exist only finitely many numbers $lw(\varphi_k, (k_1, \ldots, k_\ell))$ such that $k \in \{2, \ldots, p\}$ and $k_1 + \cdots + k_\ell = k$. We can thus compile a look-up table of these finitely many local weights.

For each tuple $(n_1, \ldots, n_\ell)$ of nonnegative integers such that $n_1 + \cdots + n_\ell = n$, fix a unique type assignment $h : n \rightarrow T$ that maps exactly n_i elements of n to α_i for each $i \in [\ell]$. Then, using h, define $\mathcal{M}_{(n_1, \ldots, n_\ell)}$ to be the class of η-models with domain n where exactly the elements i such that $h(i) = \alpha_i$, realize α_i. Clearly $\mathrm{WFOMC}(\varphi, n, w, \bar{w})$ is now given by

$$\sum_{n_1 + \cdots + n_\ell = n} \binom{n}{n_1, \ldots, n_\ell} \mathrm{WFOMC}(\varphi, n, w, \bar{w}) \restriction \mathcal{M}_{(n_1, \ldots, n_\ell)}. \tag{15}$$

Therefore, to conclude the proof, we need to find a suitable formula for $\mathrm{WFOMC}(\varphi, n, w, \bar{w}) \restriction \mathcal{M}_{(n_1, \ldots, n_\ell)}$. We shall do that next.

For each $\alpha_i \in T$, let w_{α_i} be the weight of the type α_i. Let $k_1, \ldots, k_\ell$ be nonnegative integers that sum to $k \leq n$. A k-element set with k_i realizations of α_i for each $i \in [\ell]$ can be chosen in $\binom{n_1}{k_1} \cdot \ldots \cdot \binom{n_\ell}{k_\ell}$ ways from the set n with n_i realizations of α_i fixed for each $i \in [\ell]$. By Lemmas 4.3 and 4.4, we thus see that

$$\mathrm{WFOMC}(\varphi, n, w, \bar{w}) \restriction \mathcal{M}_{(n_1, \ldots, n_\ell)} = \Big(\prod_{i \leq \ell} (w_{\alpha_i})^{n_i}\Big)$$
$$\cdot \prod_{2 \leq k \leq p} \; \prod_{k_1 + \cdots + k_\ell = k} lw(\varphi_k, (k_1, \ldots, k_\ell))^{\binom{n_1}{k_1} \cdot \ldots \cdot \binom{n_\ell}{k_\ell}}. \tag{16}$$

Therefore the function in Line (15) can clearly be computed in PTIME in n (which is given in unary). □

4.2 Weighted model counting for U_1

As discussed in the preliminaries, the weighted model counting problem of U_1-sentences can be reduced to the corresponding problem for conjunctions of $\forall^*$-sentences of U_1. A natural next step would be to follow the strategy of Section 4.1. However, that approach would fail due to Lemma 4.4 which depends crucially on the exact syntactic properties of SU_1. Thus we need a different approach. We now show how to reduce the weighted model counting problem for U_1 to the corresponding problem for SU_1.

We begin with the Lemma 4.6 below. Restricting attention to $\forall^*$-sentences in the lemma is crucial, since SU_1 is in general strictly less expressive than U_1, as shown in [11].

Lemma 4.6. *Every $\forall^*$-sentence of U_1 translates to an equivalent Boolean combination $\forall^*$-sentences of SU_1.*

Proof. We sketch the proof; see Appendix B.7 of [15] for further details. It is easy to show that every $\exists^*$-sentence of U_1 is equivalent to a disjunction of $\exists^*$-sentences of the form

$$\exists x_1 \ldots \exists x_\ell \Big(\alpha_1(x_1) \land \cdots \land \alpha_\ell(x_\ell) \land \beta(x_1,\ldots,x_k) \land \mathit{diff}(x_1,\ldots,x_\ell)\Big),$$

where α_i are 1-types and β is a k-table. For this to be an SU_1-sentence, k would need to be equal to ℓ. However, this sentence can be seen equivalent to the following conjunction of SU_1-sentences:

$$\exists x_1 \ldots \exists x_k \Big(\alpha_1(x_1) \land \cdots \land \alpha_k(x_k) \land \beta(x_1,\ldots,x_k) \land \mathit{diff}(x_1,\ldots,x_k) \Big) \land \exists x_1 \ldots \exists x_\ell \Big(\alpha_1(x_1) \land \cdots \land \alpha_\ell(x_\ell) \land \mathit{diff}(x_1,\ldots,x_\ell)\Big).$$

□

Theorem 4.7. *The weighted model counting problem is in* PTIME *for each sentence of* U_1.

Proof. As discussed in the preliminaries, it suffices to prove the theorem for a conjunction χ of $\forall^*$-sentences of U_1. We apply Lemma 4.6 to χ, obtaining a sentence $\psi \equiv \chi$ which is a Boolean combination of $\forall^*$-sentences of SU_1. By Lemmas 2.1 and 2.2, we have $\text{WFOMC}(\psi, n, w, \bar{w}) = \text{WFOMC}(Sk(Sc(\psi)), n, w', \bar{w}')$, where w' and $\bar{w}'$ are obtained from w and $\bar{w}$ by mapping the new symbols as specified in the lemmas. $Sk(Sc(\psi))$ is an $\forall^*$-sentence of SU_1. □

5 Counting and prefix classes

First-order prefix classes admit the following neat classification:

Proposition 5.1. *Consider a prefix class C_w of first-order logic defined by a quantifier-prefix $w \in \{\exists, \forall\}^*$.*

1. *If $|w| \geq 3$, then C_w contains a formula with a $\#P_1$-complete symmetric weighted model counting problem.*
2. *If $|w| < 3$, then the symmetric weighted model counting problem of each formula in C_w is in* PTIME.

We note that the proof of the proposition makes use of the results and techniques of [3, 4] in various ways, and thus much of the credit goes there. We only sketch the proof here; see Appendix C of [15] for more details.

Firstly, [3] shows that there is an FO^3-sentence φ with a $\#P_1$-complete model counting problem. We turn φ into a conjunction of prenex form sentences by eliminating quantified subformulae in a way resembling the Scott normal form procedure. We then apply the Skolemization operator Sk (see Section 2.2). Combining the obtained $\forall^*$-conjuncts, we get a sentence $\chi := \forall x \forall y \forall z \psi$ with the same model counting problem as φ; here ψ is quantifier-free.

We then start modifying the $\forall\forall\forall$-sentence χ in order to obtain, for each prefix class C with three quantifiers, a sentence in C with the same model counting problem as χ. The required modifications can be easily done by using operations that slightly generalize the Skolemization operation from Section 2.2. These operations are defined as follows. Let $\chi' := \forall x_1 \ldots \forall x_k Q_1 x_{k+1} \ldots Q_m x_m \, \chi''$ be a prenex form sentence with χ'' quantifier-free and with $Q_i \in \{\exists, \forall\}$. We turn χ' into $\forall x_1 \ldots \forall x_k Q'_1 x_{k+1} \ldots Q'_m x_m (Ax_1 \ldots x_k \lor \neg \chi'')$, where A is a fresh k-ary predicate and each Q'_i is the dual of Q_i. The difference with the Skolemization operation of Section 2.2 is simply that Q_1 is not required to be $\exists$. This new sentence has the same model counting problem as χ' when the fresh symbol A is given weights exactly as in Lemma 2.2. The proof of this claim is similar to the proof of Lemma 2.2.

The second claim of Proposition 5.1 holds by the result for FO^2.

6 Conclusions

It can be shown that WFOMC for formulae of two-variable logic with counting C^2 can be reduced to WFOMC for FO^2 with an *unbounded number* of functionality axioms. Proving tractability in that setting remains an interesting open problem. One difficulty here is that the interaction patterns of different functional relations cause effects that could intuitively be described as 'non-local' and seem to require significantly more general combinatorial arguments than those in Section 3. The tools of [13] could prove useful here.

Acknowledgments

The authors were supported by the ERC grant 647289 'CODA.' We thank the reviewers for their valuable feedback.

References

[1] Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. 2017. *An Introduction to Description Logic.* Cambridge University Press.
[2] Vince Bárány, Balder ten Cate, and Luc Segoufin. 2015. Guarded Negation. *J. ACM* 62, 3 (2015), 22:1–22:26.
[3] Paul Beame, Guy Van den Broeck, Eric Gribkoff, and Dan Suciu. 2015. Symmetric Weighted First-Order Model Counting. In *34th ACM Symposium on Principles of Database Systems (PODS).* 313–328.
[4] Guy Van den Broeck, Wannes Meert, and Adnan Darwiche. 2014. Skolemization for Weighted First-Order Model Counting. In *Principles of Knowledge Representation and Reasoning (KR).*
[5] Guy Van den Broeck and Dan Suciu. 2017. Query Processing on Probabilistic Data: A Survey. *Foundations and Trends in Databases* 7, 3-4 (2017), 197–341.
[6] Pedro M. Domingos and Daniel Lowd. 2009. *Markov Logic: An Interface Layer for Artificial Intelligence.* Morgan & Claypool Publishers.
[7] Heinz-Dieter Ebbinghaus and Jörg Flum. 1995. *Finite model theory.* Springer.
[8] Lauri Hella and Antti Kuusisto. 2014. One-Dimensional Fragment of First-order Logic. In *Advances in Modal Logic 10.* 274–293.
[9] Seyed Mehran Kazemi, Angelika Kimmig, Guy Van den Broeck, and David Poole. 2016. New Liftable Classes for First-Order Probabilistic Inference. In *Annual Conference on Neural Information Processing Systems (NIPS).* 3117–3125.
[10] Emanuel Kieronski and Antti Kuusisto. 2014. Complexity and Expressivity of Uniform One-Dimensional Fragment with Equality. In *Mathematical Foundations of Computer Science (MFCS) Part I.* 365–376.
[11] Emanuel Kieronski and Antti Kuusisto. 2015. Uniform One-Dimensional Fragments with One Equivalence Relation. In *Annual Conference on Computer Science Logic (CSL).* 597–615.
[12] Angelika Kimmig, Guy Van den Broeck, and Luc De Raedt. 2017. Algebraic Model Counting. *J. Applied Logic* 22 (2017), 46–62.
[13] Eryk Kopczynski and Tony Tan. 2015. Regular Graphs and the Spectra of Two-Variable Logic with Counting. *SIAM J. Comput.* 44, 3 (2015), 786–818.
[14] Antti Kuusisto. 2016. On the Uniform One-Dimensional Fragment. In *International Workshop on Description Logics (DL).*
[15] Antti Kuusisto and Carsten Lutz. 2018. Weighted Model Counting Beyond Two-Variable Logic. *CoRR* abs/1804.10185 (2018). http://arxiv.org/abs/1804.10185