On Exact Sampling in the Two-Variable Fragment of First-Order Logic

Yuanhong Wang[1], Juhua Pu[1], Yuyi Wang[2], and Ondřej Kuželka[3]

[1]Beihang University, China
[2]CRRC Zhuzhou Institute, China & ETH Zurich, Switzerland
[3]Czech Technical University in Prague, Czech Republic

February 7, 2023

Abstract

In this paper, we study the sampling problem for first-order logic proposed recently by Wang et al.—how to efficiently sample a model of a given first-order sentence on a finite domain? We extend their result for the universally-quantified subfragment of two-variable logic $\mathbf{FO}^2$ ($\mathbf{UFO}^2$) to the entire fragment of $\mathbf{FO}^2$. Specifically, we prove the domain-liftability under sampling of $\mathbf{FO}^2$, meaning that there exists a sampling algorithm for $\mathbf{FO}^2$ that runs in time polynomial in the domain size. We then further show that this result continues to hold even in the presence of counting constraints, such as $\forall x \exists_{=k} y : \varphi(x, y)$ and $\exists_{=k} x \forall y : \varphi(x, y)$, for some quantifier-free formula $\varphi(x, y)$. Our proposed method is constructive, and the resulting sampling algorithms have potential applications in various areas, including the uniform generation of combinatorial structures and sampling in statistical-relational models such as Markov logic networks and probabilistic logic programs.

1 Introduction

Let Γ denote a function-free first-order sentence formed over a vocabulary $\mathcal{P}$, and let Δ be a finite domain. A model of Γ interprets each predicate in $\mathcal{P}$ over Δ such that the interpretation satisfies Γ. We use $\mathcal{M}_{\Gamma,\Delta}$ to denote the set of all models of Γ over Δ. The *uniform first-order sampling problem* on Γ over Δ is to uniformly generate a model μ of Γ according to the probability $\mathbb{P}[\mu] = 1/|\mathcal{M}_{\Gamma,\Delta}|$. The weighted variant of this problem adds nonnegative weights to atomic facts and their negations in the models; the total weight of a model is the product of its facts' weights. The problem is then to sample a model according to a probability strictly proportional to its weight.

We investigate the *symmetric weighted first-order sampling problem (WFOMS)* for the two-variable fragment $\mathbf{FO}^2$ of first-order logic. The term "symmetric" refers to the property that the weights are determined solely by the relation symbol. In this paper, we focus on studying the *data complexity* of WFOMS—the complexity of sampling a model when the Γ and w and $\bar{w}$ are fixed, and the domain is considered as an input. In particular, we are interested in designing a *domain-lifted* weighted model sampler for $\mathbf{FO}^2$, which runs in time polynomial in the size of the domain.

The WFOMS was first considered by Wang et al. [1] who showed that the data complexity of WFOMS is in polynomial time for formulas of the universally-quantified subfragment $\mathbf{UFO}^2$ of $\mathbf{FO}^2$. The subfragment $\mathbf{UFO}^2$, comprising of sentences of the form $\forall x \forall y : \psi(x, y)$ with some quantifier-free formula $\psi(x, y)$, is proved to admit a lifted weighted model sampler, and then identified to be *domain-liftable under sampling*.

Symmetric weighted model sampling problems have a wide range of practical applications. For example, many problems related to the generation of combinatorial structures can easily be formulated as WFOMS and solved using the techniques developed for this problem. There are also applications of WFOMS in the realm of *statistical-relational learning (SRL)* [2]. It is known that probabilistic inference in many SRL models is reducible to weighted first-order model counting (WFOMC) [3, 4], and the same reduction can also be applied to the corresponding sampling problems.

Among the various applications of WFOMS, the input first-order sentences are usually complex and go beyond the fragment of $\mathbf{UFO}^2$. For instance, even the very simple problem of uniformly generating graphs with no isolated vertices necessitates the utilization of the existentially-quantified formula $\forall x \exists y : E(x, y)$ to encode the constraint that every vertex must have at least one incident edge. However, directly extending the approach described in [1] to $\mathbf{FO}^2$ is infeasible. As the authors showed, their technique would at some point need to solve #P-hard problems: "*. . . applying our sampling algorithm on an* $\mathbf{FO}^2$ *sentence with existential quantifiers is intractable (not domain-lifted) unless* FP=#P*,. . .* ". We stress here that they did not show the intractability of WFOMS for the $\mathbf{FO}^2$ fragment, but rather the infeasibility of their specific method, indicating that a distinct sampling approach is required. Moreover, the standard Skolemization techniques used in automated reasoning [5] and WFOMC [4] to eliminate existential quantifiers beforehand are not applicable to WFOMS, as they introduce either functions or negative weights, which make the resulting sampling problem ill-defined. This further complicates the extension of the WFOMS approach to more complex formulas beyond $\mathbf{UFO}^2$.

1.1 Our Contribution

In this paper, we present a novel sampling algorithm for the full $\mathbf{FO}^2$. The algorithm employs a completely different approach than Skolemization, based on the *domain recursion* scheme. The basic idea is to consider one object from the domain at a time, and then sample the value of all related atomic facts, resulting in a new WFOMS over a smaller domain with the object removed. The new WFOMS has an identical form to the original one but possibly contains fewer existentially-quantified formulas. The algorithm then runs recursively on the reduced sampling problems until the domain becomes singleton or all existentially-quantified formulas are eliminated. We prove that the data complexity of our algorithm is in PTIME, meaning that the entire fragment of $\mathbf{FO}^2$ is domain-liftable under sampling.

We also show how to further extend the result to the cases, where we include *counting constraints*. Specifically, our generalized algorithm can be applied to the $\mathbf{FO}^2$ sentences with additional counting constraints of the form $\forall x \exists_{=k} y : \varphi(x, y)$ and $\exists_{=k} x \forall y : \varphi(x, y)$, where φ is a quantifier-free formula and k is a natural number. This extension, originally proposed by Kuusisto and Lutz [6] and Kuzelka [7] for first-order counting problems, is mainly motivated by the connection of WFOMS to the uniform generation of combinatorial structures. For example, our algorithm can be applied to efficiently solve the uniform sampling problem of k-regular graphs, a problem that has been widely studied in the combinatorics community [8, 9]. This problem can be formulated as a WFOMS on the following sentence:

$$\forall x \forall y : (E(x, y) \Rightarrow E(y, x)) \wedge \forall x : \neg E(x, x) \wedge \forall x \exists_{=k} y : E(x, y),$$

where $\forall x \exists_{=k} y : E(x, y)$ expresses that every vertex x has exactly k connected edges.

1.2 Related Work

The symmetric weighted first-order sampling problem was first proposed and studied in [1]. The approach, as well as the formal *liftability* notions considered in that study, were derived from the literature on *lifted inference* [10, 11, 3]. In lifted inference, the goal is to perform probabilistic inference in SRL models in a way that takes advantage of the *symmetries* in the high-level structure of the models. The symmetry also exists in WFOMS and is a vital property leveraged by this paper to prove the liftability under sampling of $\mathbf{FO}^2$. We note here that the importance of symmetry for lifted inference (and its reduced WFOMC) has also been extensively discussed by Beame et al. [12].

The domain recursion approach adopted in this paper is similar to the *domain recursion rule* used in weighted first-order model counting [13, 14, 15, 16]. The domain recursion rule for WFOMC is a technique that utilizes a gradual grounding process on the input first-order sentence, where only one element of the domain is grounded at a time. As each element is grounded, the partially grounded sentence is simplified until the element is entirely removed, resulting in a new WFOMC problem with a smaller domain. With the domain recursion rule, one can apply the principle of induction on the domain size, and compute WFOMC by dynamic programming. A closely related work to this paper is the approach presented by Kazemi et al. [14], where they used the domain recursion rule to compute WFOMC without Skolemization [4], which introduces negative weights. However, it is important to note that their approach can be only applied to some specific first-order formulas, whereas the domain recursion scheme presented in this paper, mainly designed for eliminating the existentially-quantified formulas, supports the entire $\mathbf{FO}^2$ fragment.

It is also worth mentioning that sampling from *propositional* logic formula (Boolean formula) is a relatively well-studied area [17, 18, 19]. However, many real-world problems can be represented more naturally and concisely in first-order logic, and suffer from a significant increase in formula size when grounded out to propositional logic. For example, a formula of the form $\forall x \exists y : \varphi$ is encoded as a Boolean formula of the form $\bigwedge_{i=1}^{n} \bigvee_{j=1}^{n} l_{i,j}$, whose length is quadratic in the domain size n. Since even finding a solution to a such large ground formula is challenging, most sampling approaches for propositional logic instead focus on designing approximate samplers. We also note that these approaches are not polynomial-time in the length of the input formula, and rely on access to an efficient SAT solver.

2 Preliminaries

In this section, we briefly review the main necessary technical concepts that we will use in the paper

2.1 Symmetric weighted first-order sampling

We consider the function-free fragment of first-order logic. An *atom* of arity k takes the form $P(x_1, \ldots, x_k)$ where P/k is from a vocabulary of predicates (also called relations), and $x_1, \ldots, x_k$

are logical variables from a vocabulary of variables. A *literal* is an atom or its negation. A *formula* is formed by connecting one or more literals together using conjunction or disjunction. A formula may optionally be 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*. The vocabulary of a formula α is taken to be $\mathcal{P}_\alpha$.

Given a vocabulary $\mathcal{P}$, a $\mathcal{P}$-structure $\mathcal{A}$ interprets each predicate in $\mathcal{P}$ over a given domain. We often interchangeably view a structure as a set of ground literals and their conjunction. Given a $\mathcal{P}$-structure $\mathcal{A}$ and $\mathcal{P}' \subseteq \mathcal{P}$, we write $\langle \mathcal{A} \rangle_{\mathcal{P}'}$ for the projection of $\mathcal{A}$ onto $\mathcal{P}'$. We follow the standard semantics of first-order logic for determining whether a structure is a model of a formula. We denote the set of all models of a sentence Γ over the domain Δ by $\mathcal{M}_{\Gamma,\Delta}$. The two-variable syntactic fragment of first-order logic ($\mathbf{FO}^2$) is obtained by restricting the variable vocabulary to $\{x, y\}$.

The *first-order model counting problem* [3] asks, when given a domain Δ and a sentence Γ, how many models Γ has over Δ. The *weighted first-order model counting problem (WFOMC)* adds a pair of weighting functions $(w, \bar{w})$ to the input, that both map the set of all predicates in Γ to a set of weights: $\mathcal{P}_\Gamma \to \mathbb{R}$. Given a set L of literals, the weight of L is defined as

$$\langle w, \bar{w} \rangle(L) := \prod_{l \in L_T} w(\mathsf{pred}(l)) \cdot \prod_{l \in L_F} \bar{w}(\mathsf{pred}(l))$$

where L_T (resp. L_F) denotes the set of true ground (resp. false) literals in L, and $\mathsf{pred}(l)$ maps a literal l to its corresponding predicate name. The value of $\mathsf{WFOMC}(\Gamma, \Delta, w, \bar{w})$ is then the sum of the weight $\langle w, \bar{w} \rangle(\mu)$ over all models of Γ.

Recently, the model counting problem was extended to the sampling regime by [1], and the *symmetric weighted first-order sampling problem (WFOMS)* defined therein is the main focus of this paper.

Definition 1 (**Symmetric weighted first-order sampling**). *Let $(w, \bar{w})$ be a pair of weighting function: $\mathcal{P}_\Gamma \to \mathbb{R}_{\geq 0}$. The symmetric weighted first-order sampling problem on Γ over a domain Δ under $(w, \bar{w})$ is to generate a model $G(\Gamma, \Delta, w, \bar{w})$ of Γ over Δ such that*

$$\mathbb{P}[G(\Gamma, \Delta, w, \bar{w}) = \mu] = \frac{\langle w, \bar{w} \rangle(\mu)}{\mathsf{WFOMC}(\Gamma, \Delta, w, \bar{w})} \tag{1}$$

for every $\mu \in \mathcal{M}_{\Gamma,\Delta}$.

Following the terminology in [1], we call a probabilistic algorithm that realizes a solution to the WFOMS a *weighted model sampler* (WMS). A WMS is *domain-lifted* (or simply lifted) if the model generation algorithm runs in time polynomial in the size of the domain Δ. A sentence, or class of sentences, is *domain-liftable* (or simply *liftable*) *under sampling* if it admits a domain-lifted WMS.

Example 1. *The WMS of the sentence*

$$(\forall x \forall y : (E(x, y) \Rightarrow E(y, x)) \wedge \neg E(x, x)) \wedge (\forall x \exists y E(x, y))$$

over a domain of size n under the weighting $w(E) = \bar{w}(E) = 1$ uniformly samples undirected graphs with no isolated vertices.

For technical purposes, when the domain is fixed, we allow the input sentence of the WFOMC (and WFOMS) to contain some ground literals, e.g., $\Gamma = (\forall x \forall y : fr(x, y) \land sm(x) \Rightarrow sm(y)) \land sm(e_1) \land \neg sm(e_3)$ over a fixed domain of $\{e_1, e_2, e_3\}$. The WFOMC problem on such sentences is also known as *conditional* WFOMC [20, 21]. We define the probability of a sentence Φ conditional on another sentence Γ over a domain Δ under $(w, \bar{w})$ as

$$\mathbb{P}[\Phi \mid \Gamma; \Delta, w, \bar{w}] := \frac{\mathsf{WFOMC}(\Phi \land \Gamma, \Delta, w, \bar{w})}{\mathsf{WFOMC}(\Gamma, \Delta, w, \bar{w})}.$$

With a slight abuse of notation, we also write the probability of a set L of ground literals conditional on a sentence Γ over a domain Δ under $(w, \bar{w})$ in the same form:

$$\mathbb{P}[L \mid \Gamma; \Delta, w, \bar{w}] := \mathbb{P}\left[\bigwedge_{l \in L} l \mid \Gamma; \Delta, w, \bar{w}\right].$$

Then, the required sampling probability of $G(\Gamma, \Delta, w, \bar{w})$ in the WFOMS can be written as $\mathbb{P}[G(\Gamma, \Delta, w, \bar{w}) = \mu] = \mathbb{P}[\mu \mid \Gamma; \Delta, w, \bar{w}]$. When the context is clear, we omit Δ and $(w, \bar{w})$ in the conditional probability.

We call a set L of ground literals *valid* in a WFOMS $(\Gamma, \Delta, w, \bar{w})$, if there exists a model $\mu \in \mathcal{M}_{\Gamma,\Delta}$ that includes L. A *skeleton* of a WFOMS $(\Gamma, \Delta, w, \bar{w})$ is a subset $\mathcal{P}$ of the vocabulary $\mathcal{P}_\Gamma$, such that the interpretation for $\mathcal{P}$ fully determines $\mathcal{P}_\Gamma \setminus \mathcal{P}$ in the models of Γ, and for any predicate $P \in \mathcal{P}_\Gamma \setminus \mathcal{P}$, $w(P) = \bar{w}(P) = 1$. Using the notion of skeleton, a WFOMS $(\Gamma, \Delta, w, \bar{w})$ can be reduced to randomly generating a valid $\mathcal{P}$-structure $G(\Gamma, \Delta, w, \bar{w})$ such that

$$\mathbb{P}[G(\Gamma, \Delta, w, \bar{w}) = \langle \mu \rangle_{\mathcal{P}}] = \mathbb{P}[\langle \mu \rangle_{\mathcal{P}} \mid \Gamma; \Delta, w, \bar{w}]$$

for every $\mu \in \mathcal{M}_{\Gamma,\Delta}$, where $\mathcal{P}$ is a skeleton of the problem.

In this paper, we will be performing reductions between WFOMS problems. We want such reductions to be *sound*.

Definition 2 (**Soundness**). *A reduction of a WFOMS of* $(\Gamma, \Delta, w, \bar{w})$ *to* $(\Gamma', \Delta', w', \bar{w}')$ *is sound iff there exists a polynomial-time deterministic function* f *mapping from* $\mathcal{M}_{\Gamma',\Delta'}$ *to* $\mathcal{M}_{\Gamma,\Delta}$*, and for every model* $\mu \in \mathcal{M}_{\Gamma,\Delta}$*,*

$$\mathbb{P}[\mu \mid \Gamma; \Delta, w, \bar{w}] = \sum_{\substack{\mu' \in \mathcal{M}_{\Gamma',\Delta'}: \\ f(\mu') = \mu}} \mathbb{P}[\mu' \mid \Gamma'; \Delta', w', \bar{w}']. \tag{2}$$

A general mapping function f used most in this paper is the projection $f(\mu') = \langle \mu' \rangle_{\mathcal{P}_\Gamma}$, where $\mathcal{P}_\Gamma$ is a skeleton of $(\Gamma', \Delta', w', \bar{w}')$. In this case, the mapping function is bijective and preserves the weight of the mapped models. Through a sound reduction, we can easily transform a WMS G' of $(\Gamma', w', \bar{w}', \Delta')$ to a WMS G of $(\Gamma, w, \bar{w}, \Delta)$ by $G(\Gamma, \Delta, w, \bar{w}) = f(G'(\Gamma', \Delta', w', \bar{w}'))$. Note that the soundness is transitive, i.e., if the reductions from a WFOMS $\mathfrak{S}_1$ to $\mathfrak{S}_2$ and from $\mathfrak{S}_2$ to $\mathfrak{S}_3$ are both sound, the reduction from $\mathfrak{S}_1$ to $\mathfrak{S}_3$ is also sound.

2.2 Types and Tables

We define a *1-literal* as an atomic predicate or its negation using only the variable x, and a *2-literal* as an atomic predicate or its negation using both variables x and y. An atom like

$R(x,x)$ or its negation is considered a 1-literal, even though R is a binary relation. A 2-literal is always of the form $R(x,y)$ and $R(y,x)$, or their respective negations.

Let $\mathcal{P}$ be a finite vocabulary. A *1-type* over $\mathcal{P}$ is a maximally consistent set of 1-literals formed by $\mathcal{P}$. Denote the set of all 1-types over $\mathcal{P}$ as $U_{\mathcal{P}}$. The size of $U_{\mathcal{P}}$ is finite and only depends on the size of $\mathcal{P}$. We often view a 1-type τ as a conjunction of its elements, whence $\tau(x)$ is simply a formula in the single variable x.

Let $\mathcal{A}$ be a structure over $\mathcal{P}$. A domain element $e \in \mathsf{dom}(\mathcal{A})$ *realizes* the 1-type τ if $\mathcal{A} \models \tau(e)$. Note that every element of $\mathcal{A}$ realizes exactly one 1-type over $\mathcal{P}$, which we call the *1-type of the element.* The *cardinality* of a 1-type is the number of elements realizing it.

A *2-table* over $\mathcal{P}$ is a maximally consistent set of 2-literals formed by $\mathcal{P}$. We often identify a 2-table π with a conjunction of its elements and write it as a formula $\pi(x,y)$. Denote $T_{\mathcal{P}}$ the set of all 2-tables over $\mathcal{P}$, whose size also only depends on the size of $\mathcal{P}$. Given a $\mathcal{P}$-structure $\mathcal{A}$ over a domain Δ, the *2-table of an element tuple* $(a,b) \in \Delta^2$ is the unique 2-table π that (a,b) satisfies in $\mathcal{A}$: $\mathcal{A} \models \pi(a,b)$. It is worth noting that the 1-types together with the 2-tables fully characterize a structure.

Example 2. *Consider the vocabulary $\mathcal{P} = \{F/2, G/1\}$ and the structure*

$$\{F(a,a), G(a), \neg F(b,b), G(b), F(a,b), \neg F(b,a)\}$$

over the domain $\{a,b\}$. The 1-type of the elements a and b are $F(x,x) \wedge G(x)$ and $\neg F(x,x) \wedge G(x)$ respectively. The cardinalities of these two 1-types are both 1, while that of the other 1-types $F(x,x) \wedge \neg G(x)$ and $\neg F(x,x) \wedge \neg G(x)$ are both 0. The 2-table of the element tuples (a,b) and (b,a) are $F(x,y) \wedge \neg F(y,x)$ and $\neg F(x,y) \wedge F(y,x)$ respectively.

2.3 Universally Quantified $\mathbf{FO}^2$ is Liftable under Sampling

As an elementary attempt to the symmetric weighted first-order sampling problem, Wang et al. [1] provided a positive result of the data complexity for the *universally quantified fragment of* $\mathbf{FO}^2$ ($\mathbf{UFO}^2$) of the form $\forall x \forall y : \psi(x,y)$, where $\psi(x,y)$ is a quantifier-free formula[1].

The proof of this result established a general framework for designing a WMS. Therefore, We summarize the main ideas of their argument here and refer the reader to their paper for the complete proof and technical details. We note that the approach presented here is slightly different from the original one in [1]. The main divergence is that, instead of using the notion of count distribution [7], we perform the sampling of 1-types by a random partition on the domain, which keeps in line with our sampling algorithm for $\mathbf{FO}^2$.

Theorem 1 (Proposition 1 in [1])**.** *The fragment* $\mathbf{UFO}^2$ *is domain-liftable under sampling.*

Proof sketch. Suppose that we wish to randomly sample models from some input $\mathbf{UFO}^2$ sentence $\Gamma = \forall x \forall y : \psi(x,y)$ over a domain $\Delta = \{e_1, e_2, \ldots, e_n\}$ under weights $(w, \bar{w})$. Given a $\mathcal{P}_\Gamma$-structure $\mathcal{A}$ over Δ, we denote τ_i the 1-type of the ith element and $\pi_{i,j}$ the 2-table of the tuple of the ith and jth elements. The structure $\mathcal{A}$ is fully characterized by the ground 1-types $\tau_i(e_i)$'s

[1] They went a bit beyond this fragment, e.g., $\mathbf{UFO}^2$ with cardinality constraints, which we also handle later in this paper.

and 2-tables $\pi_{i,j}(e_i,e_j)$'s. We can write the sampling probability of $\mathcal{A}$ as

$$\begin{aligned}\mathbb{P}[\mathcal{A} \mid \Gamma] &= \mathbb{P}\left[\bigwedge_{i\in[n]} \tau_i(e_i) \wedge \bigwedge_{i,j\in[n]:i<j} \pi_{i,j}(e_i,e_j) \mid \Gamma\right] \\ &= \underbrace{\mathbb{P}\left[\bigwedge_{i\in[n]} \tau_i(e_i) \mid \Gamma\right]}_{\mathfrak{P}_1} \cdot \underbrace{\mathbb{P}\left[\bigwedge_{\substack{i,j\in[n]:\\ i<j}} \pi_{i,j}(e_i,e_j) \mid \Gamma \wedge \bigwedge_{i\in[n]} \tau_i(e_i)\right]}_{\mathfrak{P}_2},\end{aligned}$$

where $[n]$ denotes the set of $\{1,2,\ldots,n\}$. This decomposition naturally gives rise to a two-phase sampling algorithm:

1. sample 1-types τ_i's according to the probability $\mathfrak{P}_1$, and
2. randomly assign 2-table $\pi_{i,j}$ to each element tuple according to $\mathfrak{P}_2$.

The sampling of 1-types can be achieved through a random partition of the domain Δ, resulting in $|U_{\mathcal{P}_\Gamma}|$ disjoint subsets of Δ; each subset contains the elements assigned to its corresponding 1-type. The symmetry property of the weighting function ensures that the satisfaction and weight of the models are not affected by permutations of the domain elements. Therefore, any partitions of the domain with the same partition size have the same probability to be sampled. This further decomposes the sampling problem of 1-types into two stages: the stochastic generation of partition size and the random partitioning of the domain according to the sampled size. Randomly partitioning the domain according to the sampled size is straightforward, and we will demonstrate that sampling a partition size can be done in time polynomial in the domain size.

Recall that the number $|U_{\mathcal{P}_\Gamma}|$ of all 1-types only depends on the input sentence, and thus enumerating all possible partition sizes is computationally tractable. For any partition size $(n_1,n_2,\ldots,n_{|U_{\mathcal{P}_\Gamma}|})$, there are totally $\binom{n}{n_1,n_2,\ldots,n_{|U_{\mathcal{P}_\Gamma}|}}$ partitions of the domain with the same sampling probability. It will turn out that the sampling probability, which is of the form $\mathfrak{P}_1$, can be computed in time polynomial in the domain size. The reason for this is: we expand $\mathfrak{P}_1$ into $\mathsf{WFOMC}(\Gamma \wedge \bigwedge_{i=1}^n \eta_i(e_i), \Delta, w, \bar{w})/\mathsf{WFOMC}(\Gamma, \Delta, w, \bar{w})$; the numerator WFOMC can be viewed as a WFOMC of Γ conditional on the unary facts in $\eta_i(e_i)$'s, whose complexity is polynomial in the domain size by [20]; and the denominator WFOMC can be also efficiently computed due to the liftability (in terms of WFOMC) of Γ by [3]. Finally, the sampling probability of the partition size $(n_1,n_2,\ldots,n_{|U_{\mathcal{P}_\Gamma}|})$ is given by $\mathfrak{P}_1 \cdot \binom{n}{n_1,n_2,\ldots,n_{|U_{\mathcal{P}_\Gamma}|}}$.

For sampling $\pi_{i,j}$'s according to $\mathfrak{P}_2$, we first ground out Γ over the domain Δ:

$$\bigwedge_{i,j\in[n]:i<j} \psi(e_i,e_j) \wedge \psi(e_j,e_i).$$

Let $\psi'_{i,j}(x,y)$ be the simplified formula of $\psi(x,y) \wedge \psi(y,x)$ obtained by replacing the unary ground literals with their truth value given by the 1-types τ_i and τ_j. Then the probability $\mathfrak{P}_2$ can be written as

$$\mathbb{P}[\bigwedge_{i,j\in[n]:i<j} \pi_{i,j}(e_i,e_j) \mid \bigwedge_{i,j\in[n]:i<j} \psi'_{i,j}(e_i,e_j)].$$

Note that in this probability, each ground 2-tables $\pi_{i,j}(e_i, e_j)$ are independent in the sense that they do not share any ground literals. The independence also holds for the ground formulas $\psi'_{i,j}(e_i, e_j)$'s. It follows that the probability $\mathfrak{P}_2$ can be factorized into

$$\prod_{i,j\in[n]:i<j} \mathbb{P}[\pi_{i,j}(e_i, e_j) \mid \psi'_{i,j}(e_i, e_j)].$$

Hence, the sampling of each $\pi_{i,j}$ can be solved separately by randomly choosing a model of its respective ground formula $\psi'(e_i, e_j)$ according to the probability $\mathbb{P}[\pi_{i,j}(e_i, e_j) \mid \psi'_{i,j}(e_i, e_j)]$. The overall computational complexity is clearly polynomial in the domain size.

The procedure for both sampling τ_i's and $\pi_{i,j}$'s is polynomial in the domain size, which completes the liftability under sampling of $\mathbf{UFO}^2$. □

Extending the approach above to the case of $\mathbf{FO}^2$ would requires a novel and more sophisticated strategy, especially for the sampling of 2-tables, as decoupling the grounding of $\forall x \exists y : \varphi(x, y)$ to the form of $\bigwedge_{i,j\in[n]:i<j} \psi_{i,j}(e_i, e_j)$ is impossible even conditioning on the sampled 1-types.

2.4 Notations

We will use $[n]$ to denote the set of $\{1, 2, \ldots, n\}$. The notation $\{x_i\}_{i\in[n]}$ represents the set of terms $\{x_1, x_2, \ldots, x_n\}$, and $(x_i)_{i\in[n]}$ the vector of $(x_1, x_2, \ldots, x_n)$. We also use the bold symbol $\mathbf{x}$ to denote a vector $(x_i)_{i\in[n]}$, and denote by $\mathbf{x}^{\mathbf{y}}$ the product over element-wise power of two vectors $\mathbf{x}^{\mathbf{y}} = \prod_{i\in[n]} x_i^{y_i}$. The notation $\oplus$ is used to denote the concatenation of two vectors. Using the vector notation, we often write the multinomial coefficient $\binom{N}{x_1, x_2, \ldots, x_n}$ as $\binom{N}{\mathbf{x}}$.

3 Sampling Algorithm for $\mathbf{FO}^2$

We now show the domain-liftability under sampling of the $\mathbf{FO}^2$ fragment by providing a lifted WMS for it. It is common for logical algorithms to operate on normal form representations instead of arbitrary sentences. The normal form of $\mathbf{FO}^2$ used in our sampling algorithm is the *Scott normal form (SNF)* [22]; an $\mathbf{FO}^2$ sentence is in SNF, if it is written as:

$$\Gamma = \forall x \forall y : \psi(x, y) \wedge \bigwedge_{k\in[m]} \forall x \exists y : \varphi_k(x, y), \tag{3}$$

where ψ and φ_k's are quantifier-free formulas. It is well-known that one can convert any $\mathbf{FO}^2$ sentence Γ in polynomial time into a formula Γ_S in SNF such that Γ and Γ_S are equisatisfiable [23]. The principal idea is to substitute, starting from the atomic level and working upwards, any subformula $\varphi(x) = Qy : \phi(x, y)$, where $Q \in \{\forall, \exists\}$ and ϕ is quantifier-free, with an atomic formula A_φ, where A_φ is a fresh predicate symbol. This novel atom $A_\varphi(x)$ is then separately "axiomatized" to be equivalent to $\varphi(x)$. The weight of A_φ is set to be $w(A_\varphi) = \bar{w}(A_\varphi) = 1$. It follows from reasoning similar to one by Kuusisto and Lutz [6] that such reduction is not only equisatisfiable but also sound (according to Definition 2).

Lemma 1. *For any WFOMS of $\mathfrak{S} = (\Gamma, \Delta, w, \bar{w})$ where Γ is an $\mathbf{FO}^2$ sentence, there exists a WFOMS $\mathfrak{S}' = (\Gamma', \Delta, w', \bar{w}')$, where Γ' is in SNF and independent of Δ, such that the reduction from $\mathfrak{S}$ to $\mathfrak{S}'$ is sound.*

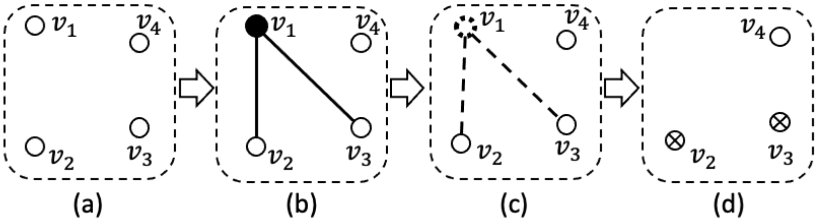


Figure 1: A sampling step for an undirected graph with no isolated vertices: (a) begin with an initial graph that has no edges, and in the more general sampling problem, $V_\forall = V_\exists = \{v_1, v_2, v_3, v_4\}$; (b) sample edges for the vertex v_1; (c) remove the vertex v_1 with its sampled edges; (d) and obtain a graph with some vertices already nonisolated (v_2 and v_3), resulting in a new sampling problem with $V'_\forall = \{v_2, v_3, v_4\}$ and $V'_\exists = \{v_4\}$.

The proof is clear, as every novel predicate (e.g., P_φ) introduced in the SNF transformation is axiomatized to be equivalent to the subformula ($\varphi(x)$), whose quantifiers are to be eliminated, and thus the interpretation of the predicate is fully determined by the subformula in every model of the resulting SNF sentence (see the details in Appendix A.1).

3.1 An Intuitive Example

We start with an intuitive example of how to generate an undirected graph of size n without any isolated vertex uniformly at random, to illustrate the basic idea of our sampling algorithm. This graph structure can be expressed by an $\mathbf{FO}^2$ sentence in SNF,

$$\Gamma_G := (\forall x \forall y : (E(x,y) \Rightarrow E(y,x)) \land \neg E(x,x)) \land (\forall x \exists y : E(x,y)),$$

and the sampling problem corresponds to a WFOMS on Γ_G under $w(E) = \bar{w}(E) = 1$ over a domain of vertices $V = \{v_i\}_{i\in[n]}$. In this sentence, the only realizable 1-type is $\neg E(x,x)$, and the realizable 2-tables are $\pi^1(x,y) = E(x,y) \land E(y,x)$ and $\pi^2(x,y) = \neg E(x,y) \land \neg E(y,x)$ representing the connectedness of two vertices.

We first apply the following transformation on Γ_G resulting in Γ_{GT}:

1. introduce an auxiliary *Tseitin predicate* $Z/1$ that indicates the non-isolation of vertices, and append $\forall x : Z(x) \Leftrightarrow \exists y : E(x,y)$ to Γ_G, and

2. remove $\forall x \exists y : E(x,y)$,

and set the weight of the predicate Z to $w(Z) = \bar{w}(Z) = 1$. Then we consider a bit more general WFOMS of $(\Gamma_{GT} \land \bigwedge_{v\in V_\exists} Z(v), V_\forall, w, \bar{w})$, where $V_\exists \subseteq V_\forall \subseteq V$ and $V_\exists$ represents the set of nonisolated vertices on the graph induced by the vertex set $V_\forall$. The original WFOMS on Γ_G can be clearly reduced to the more general problem by setting $V_\exists = V_\forall = V$, and the reduction is sound with the mapping function $f(\mu') = \langle \mu' \rangle_{\{E\}}$.

Let $\widehat{\Gamma}_G$ denote the sentence $\Gamma_{GT} \land \bigwedge_{v\in V_\exists} Z(v)$. For the more general WFOMS, since $\{E\}$ is its skeleton, the problem is equivalent to sampling an $\{E\}$-structure $\mathcal{A}$ over $V_\forall$ according to the probability $\mathbb{P}[\mathcal{A} \mid \widehat{\Gamma}_G]$. Given an $\{E\}$-structure $\mathcal{A}$ over $V_\forall$, denote by $\mathcal{A}_i$ the *substructure* of $\mathcal{A}$ concerning the vertex $v_i \in V_\forall$, which consists of the 2-tables of all vertex tuples containing v_i and other vertices in $V_\forall$:

$$\mathcal{A}_i := \bigcup_{v_j \in V_\forall : j \neq i} \pi_{i,j}(v_i, v_j),$$

where $\pi_{i,j}$ is the 2-table of (v_i, v_j). Following the domain recursion scheme, we choose a vertex v_t from $V_\forall$ and decompose the sampling probability of $\mathcal{A}$ into

$$\mathbb{P}[\mathcal{A} \mid \widehat{\Gamma}_G] = \mathbb{P}\left[\mathcal{A} \mid \widehat{\Gamma}_G \wedge \mathcal{A}_t\right] \cdot \mathbb{P}[\mathcal{A}_t \mid \widehat{\Gamma}_G].$$

The decomposition leads to two successive subproblems of the general WFOMS: the first one is to sample a valid substructure $\mathcal{A}_t$ from $\widehat{\Gamma}_G$; the other can be viewed as a new WFOMS on $\widehat{\Gamma}_G$ given the sampled $\mathcal{A}_t$.

We first show that the new WFOMS can be also reduced to the general WFOMS but with the smaller domain $V'_\forall = V_\forall \setminus \{v_t\}$ and

$$V'_\exists = \{v_i \mid v_i \in V_\exists : v_i \neq v_t \wedge \pi_{t,i} = \pi^2\}.$$

The reduction is obviously sound because every model of the WFOMS $(\Gamma_{GT} \wedge \bigwedge_{v \in V'_\exists} Z(v), V'_\forall, w, \bar{w})$ can be mapped to a unique model of the WFOMS $(\widehat{\Gamma}_G \wedge \mathcal{A}_t, V_\forall, w, \bar{w})$, and vice versa, without affecting the weight of the models. Thus, the decomposition can be performed recursively on any WFOMS on $\widehat{\Gamma}_G$ over $V_\forall$. Specifically, the algorithm takes $V_\forall$ and $V_\exists$ as input,

1. selects a vertex v_t from $V_\forall$,
2. samples its substructure $\mathcal{A}_t$ according to the probability $\mathbb{P}[\mathcal{A}_t \mid \widehat{\Gamma}_G]$, and
3. obtains a new problem with updated $V'_\forall$ and $V'_\exists$ for recursion.

The recursion procedure terminates when all substructures $\mathcal{A}_i$'s are sampled ($V_\forall$ contains a single vertex), or the problem degenerates to a WFOMS on $\mathbf{UFO}^2$ sentence ($V_\exists$ is empty). The number of recursions is less than $|V|$, the total number of vertices. An example of a recursion step is shown in Figure 1.

The remaining problem is to sample the substructure $\mathcal{A}_t$ according to $\mathbb{P}[\mathcal{A}_t \mid \widehat{\Gamma}_G]$. Recall that $\mathcal{A}_t$ determines the edges between v_t and vertices in $V'_\forall$. Let $V_1 = V'_\forall \setminus V_\exists$ and $V_2 = V'_\forall \setminus V_1$. The sampling of $\mathcal{A}_t$ can be realized by performing two random binary partitions on V_1 and V_2 respectively, resulting in $\{V_{11}, V_{12}\}$ and $\{V_{21}, V_{22}\}$, where the vertices in V_{11} and V_{21} will be connected to v_t, while the vertices in V_{12} and V_{22} will not be connected to it. It can be demonstrated that the sampling probability of a substructure $\mathcal{A}_t$ only depends on the size $(|V_{11}|, |V_{12}|, |V_{21}|, |V_{22}|)$. The proof of this claim can be found in Section 3.4, where the more general case of $\mathbf{FO}^2$ sampling is addressed. As a result, the sampling of $\mathcal{A}_t$ can be accomplished by a random generation of the partition size, followed by two random partitions of the sampled size on V_1 and V_2 respectively. We use the enumerative sampling method to generate a partition size. The number of all possible partition sizes is clearly polynomial in $|V_\forall|$, and it will be shown in Section 3.4 that the sampling probability of each partition size can be computed in time polynomial in $|V'_\forall|$. Therefore, the complexity of the sampling algorithm is polynomial in the number of vertices. This, together with the complexity of the recursion procedure, which is also polynomial in the number of vertices, implies that the whole sampling algorithm is lifted.

3.2 A More General Sampling Problem

W.l.o.g.[2] , we suppose that each formula $\varphi_k(x, y)$ in the SNF sentence (3) is an atomic formula $R_k(x, y)$, where R_k is a binary predicate in $\mathcal{P}_{\psi(x,y)}$, and its weights $w(R_k) = \bar{w}(R_k) = 1$. We

[2] Any SNF sentence can be transformed into such form by introducing an auxiliary predicate R_k with weights $w(R_k) = \bar{w}(R_k) = 1$ for each $\varphi_k(x, y)$, append $\forall x \forall y : R_k(x, y) \Leftrightarrow \varphi_k(x, y)$ to the sentence, and replacing $\varphi_k(x, y)$ with $R_k(x, y)$. The transformation is obviously sound when viewing it as a reduction in WFOMS.

first construct the following sentence from the SNF one:

$$\Gamma_T := \forall x \forall y : \psi(x,y) \wedge \bigwedge_{k\in[m]} \forall x : Z_k(x) \Leftrightarrow (\exists y : R_k(x,y)), \tag{4}$$

where each $Z_k/1$ is a Tseitin predicate with the weight $w(Z_k) = \bar{w}(Z_k) = 1$.

We then consider a more general WFOMS problem on the following sentence

$$\widehat{\Gamma} := \Gamma_T \wedge \bigwedge_{i\in[n]} \beta_i(e_i) \tag{5}$$

over a domain of $\{e_i\}_{i\in[n]}$, where each $\beta_i(x)$ is a quantifier-free conjunction over a subset of the atoms $\{Z_k(x)\}_{k\in[m]}$. We call $\beta_i(x)$ the *existential constraint* on the element e_i and allow $\beta_i(x) = \top$, which means e_i is not existentially quantified. The original WFOMS on Γ can be reduced to a more general problem by setting all existential constraints to be $\bigwedge_{k\in[m]} Z_k(x)$. On the other hand, the WFOMS on the $\mathbf{UFO}^2$ sentence $\forall x \forall y : \psi(x,y)$ is also reducible to the problem with $\beta_i(x) = \top$ for all $i \in [n]$. It is easy to check that these two reductions are both sound. The main idea of our sampling algorithm is to use the domain recursion scheme to *gradually remove the existential constraints until we eventually end up with a WFOMS problem on a UFO2 sentence.*

3.3 Partitioning the Domain

We also need the notions of *blocks* and *cells*. Unless stated otherwise, in the rest of this section, 1-types and 2-tables are defined over $\mathcal{P}_\Gamma$, where Γ is the sentence in SNF. Please bear in mind that the Tseitin predicates Z_k's are not in these 1-types.

Consider an $\mathbf{FO}^2$ sentence $\widehat{\Gamma}$ of the form (5). A *block type* β is a subset of the Tseitin atoms $\{Z_k(x)\}_{k\in[m]}$. The number of the block types is 2^m, where m is the number of existentially-quantified formulas. We often represent a block type as $\beta(x)$ and view it as a conjunctive formula over the atoms within the block. The *block type of an element* e is the existential constraint $\beta(x)$ imposed on e. We fix the order of all block types and denote by β^i the ith block type. The domain Δ is then partitioned by the blocks $\{B_{\beta^i}\}_{i\in[2^m]}$, where each subset $B_{\beta^i} \subseteq \Delta$ contains precisely the domain elements with the block type β^i. It is important to note that the block types only indicate which Tseitin atoms should hold for a given element, and the Tseitin atoms not covered by the block types are left unspecified. In contrast, the 1-types explicitly determine the truth values of all unary and reflexive atoms, excluding the Tseitin atoms.

The blocks are further partitioned into *cells*. A *cell type* $\eta = (\beta, \tau)$ is a pair of a block type β and a 1-type τ. We also write a cell type as a conjunctive formula of $\eta(x) = \beta(x) \wedge \tau(x)$. Given a $\mathcal{P}_\Gamma$-structure $\mathcal{A}$, the *cell type of an element* e is the combination of its block type (which is given by the sentence $\widehat{\Gamma}$) and its realizing 1-type in $\mathcal{A}$. Each block B_β is partitioned by the *cells* $\{C_\eta^{\mathcal{A}} \mid \eta = (\beta, \tau), \tau \in U_{\mathcal{P}_\Gamma}\}$, where each cell $C_\eta^{\mathcal{A}} \subseteq B_\beta$ contains precisely the domain elements that are of cell type η.

For brevity, we denote by $N_u = |U_{\mathcal{P}_\Gamma}|$, the number of all 1-types and $N_c = 2^m \times N_u$, the number of all cell types. We fix a linear order of 1-types as well as cell types, and let τ^i and η^j be the ith 1-type and jth cell type respectively. Given a $\mathcal{P}_{\widehat{\Gamma}}$-structure $\mathcal{A}$ with the cell partition $\{C_{\eta^i}^{\mathcal{A}}\}_{i\in[N_c]}$, we call the partition size of $\left(|C_{\eta^i}^{\mathcal{A}}|\right)_{i\in[N_c]}$ the *cell configuration* of $\mathcal{A}$, and the vector

$\left(|C^{\mathcal{A}}_{(\beta,\tau^i)}|\right)_{i\in[N_u]}$ the *cell configuration of $\mathcal{A}$ in the block β.* A $\mathcal{P}_{\widehat{\Gamma}}$-structure will have a unique cell configuration.

We will often care about the set of all cell configurations over a set of elements, which is defined as the *configuration space.*

Definition 3 (**Configuration Space**)**.** *Given a nonnegative integer M and a positive integer m, we define the configuration space $\mathcal{T}_{M,m}$ as*

$$\mathcal{T}_{M,m} = \left\{ (n_i)_{i\in[m]} \mid \sum_{i\in[m]} n_i = M, n_1, n_2, \ldots, n_m \in \mathbb{N} \right\}.$$

The size of $\mathcal{T}_{M,m}$ is equal to $\binom{M+m-1}{m-1}$, whose value is clearly polynomial in M (while exponential in m).

3.4 The Sampling Algorithm

We now describe our algorithm for the WFOMS of $(\widehat{\Gamma}, \Delta, w, \bar{w})$ where $\widehat{\Gamma}$ is of the form (5) and $\Delta = \{e_i\}_{i\in[n]}$, and prove that the algorithm is domain-lifted (i.e. runs in time polynomial in the domain size). It can be easily verified that $\mathcal{P}_\Gamma$ is a skeleton of the WFOMS problem. Hence, this WFOMS problem is equivalent to sampling a valid $\mathcal{P}_\Gamma$-structure $\mathcal{A}$ according to the probability $\mathbb{P}[\mathcal{A} \mid \widehat{\Gamma}]$.

Given a $\mathcal{P}_\Gamma$-structure $\mathcal{A}$ over Δ, let τ_i be the 1-type of the element e_i, and denote by $\eta_i = (\beta_i, \tau_i)$ the cell type of e_i. Using the notation of conditional probability, we decompose the sampling probability as

$$\mathbb{P}[\mathcal{A} \mid \widehat{\Gamma}] = \mathbb{P}\left[\bigwedge_{i\in[n]} \tau_i(e_i) \mid \widehat{\Gamma}\right] \cdot \mathbb{P}\left[\mathcal{A} \mid \widehat{\Gamma} \wedge \bigwedge_{i\in[n]} \tau_i(e_i)\right]$$

Following a similar idea to the one used for sampling models from $\mathbf{UFO}^2$ sentences, our proposed algorithm is divided into two phases—we first sample the 1-type for each element, and then handle the sampling of the structure conditional on the sampled 1-types.

3.4.1 Sampling 1-Types

Let us first consider the sampling of 1-types according to $\mathbb{P}[\bigwedge_{i\in[n]} \tau_i(e_i) \mid \widehat{\Gamma}]$. Note that the block type of each element has been determined by the sentence $\widehat{\Gamma}$, and thus the problem is equivalent to sampling the cell type of each element [3]. Due to the symmetry of the WFOMS problem, the sampling probability of a cell partition is completely determined by its corresponding cell configuration. Therefore, the problem of randomly partitioning cells is further reduced to a problem of sampling cell configurations.

The sampling algorithm for cell configurations is outlined in Algorithm 1. The algorithm begins by sampling a random cell configuration in Line 1-14, which is then used to randomly partition each block into cells in Line 15-21. While the partitioning process is relatively straightforward, in the following discussion, we will focus specifically on how to sample a cell configuration.

[3] In the special case where there is a single block, e.g., $\widehat{\Gamma}$ is of the standard SNF or in $\mathbf{UFO}^2$, the sampling problem can be simplified.

Algorithm 1 $\mathsf{OneTypeSampler}(\widehat{\Gamma}, \Delta, w, \bar{w})$

```
$W \leftarrow \mathsf{WFOMC}(\widehat{\Gamma}, \Delta, w, \bar{w})$
Obtain the blocks $B_{\beta^1}, B_{\beta^2}, \ldots, B_{\beta^{2^m}}$ from $\widehat{\Gamma}$
for $\left(\mathbf{n}_{\beta^i}\right)_{i\in[2^m]} \in \mathsf{Prod}\left(\mathcal{T}_{|B_{\beta^1}|,N_u}, \ldots, \mathcal{T}_{|B_{\beta^{2^m}}|,N_u}\right)$ do
    $\mathbf{n} \leftarrow \bigoplus_{i\in[2^m]} \mathbf{n}_{\beta^i}$
    Compute $\mathcal{W}_{\mathbf{n}}$ by (6)
    $W' \leftarrow \mathcal{W}_{\mathbf{n}} \cdot \prod_{t=1}^{2^m} \binom{|B_{\beta^t}|}{\mathbf{n}_{\beta^t}}$
    // $\mathsf{Uniform}(0,1)$ produces a uniformly random number over $[0,1]$
    if $\mathsf{Uniform}(0,1) < \frac{W'}{W}$ then
        $\mathbf{n}^* \leftarrow \mathbf{n}$
        break
    else
        $W \leftarrow W - W'$
    end if
end for
for $i \in [2^m]$ do
    Fetch the cell configuration $\mathbf{n}^*_{\beta^i}$ in $\beta^i$ from $\mathbf{n}^*$
    Randomly partition $B_{\beta^i}$ into $\{C_{\beta^i,\tau^j}\}_{j\in[N_u]}$ according to $\mathbf{n}^*_{\beta^i}$
    for $j \in [N_u]$ do
        Assign the 1-type $\tau^j$ to all elements in $C_{\beta^i,\tau^j}$
    end for
end for
```

The basic idea is again based on enumerative sampling. Let $B_{\beta^1}, B_{\beta^2}, \ldots, B_{\beta^{2^m}}$ be the blocks defined by $\widehat{\Gamma}$. Any cell configuration $\mathbf{n}$ can be viewed as a concatenation of 2^m cell configurations $\mathbf{n}_{\beta^1}, \mathbf{n}_{\beta^2}, \ldots, \mathbf{n}_{\beta^{2^m}}$ in the blocks, and each $\mathbf{n}_{\beta^i}$ is from the configuration space $\mathcal{T}_{|B_{\beta^i}|,N_u}$. Hence, in the algorithm, the enumeration of all cell configurations is performed by applying the Cartesian product function Prod on the configuration spaces $\mathcal{T}_{|B_{\beta^1}|,N_u}, \mathcal{T}_{|B_{\beta^2}|,N_u}, \ldots, \mathcal{T}_{|B_{\beta^{2^m}}|,N_u}$.

For the computation of the sampling probability, we first observe that any cell partitions that produce the same configuration $\mathbf{n}$ will have the same sampling probability. We use $\mathcal{W}_{\mathbf{n}}$ to denote the sampling weight (the numerator WFOMC of the probability) of any such cell partition. Then the sampling probability of a cell configuration $\mathbf{n}$ can be derived from $\mathcal{W}_{\mathbf{n}} \cdot \prod_{i\in[2^m]} \binom{|B_{\beta^i}|}{\mathbf{n}_{\beta^i}}$, where the latter product equates to the total number of the partitions giving rise to $\mathbf{n}$.

The value of $\mathcal{W}_{\mathbf{n}}$ will play a crucial role in the remaining sampling algorithm. In this context, we provide its formal definition. Given a nonnegative integer vector $\mathbf{n}$ of size N_c, let $\widetilde{n} = \sum_{i\in[N_c]} n_i$, and $\mathcal{W}_{\mathbf{n}}$ be defined as

$$\mathcal{W}_{\mathbf{n}} := \mathsf{WFOMC}(\Gamma_T \wedge \bigwedge_{i\in[\widetilde{n}]} \widetilde{\eta}_i(\widetilde{e}_i), \widetilde{\Delta}, w, \bar{w}), \tag{6}$$

where $\{\eta_i\}_{i\in[\widetilde{n}]}$ is a set of cell types that gives rise to the configuration $\mathbf{n}$, $\widetilde{\Delta}$ is a domain of size $\widetilde{n}$, and $\widetilde{e}_i$ is the ith element in $\widetilde{\Delta}$. Recall that a cell type can be represented by a conjunction of unary atoms, and thus each $\widetilde{\eta}_i(\widetilde{e}_i)$ in $\mathcal{W}_{\mathbf{n}}$ can be interpreted as a set of unary facts. The value of $\mathcal{W}_{\mathbf{n}}$ is then equal to the WFOMC of the $\mathbf{FO}^2$ sentence Γ_T conditional on a set of unary facts.

Such conditional counting problems have been thoroughly studied by Van den Broeck and Davis [20], and the computational complexity has been shown to be polynomial in the domain size $\widetilde{n}$ and the number of facts. Since the number of facts is clearly polynomial in $\widetilde{n}$, computing $\mathcal{W}_{\mathbf{n}}$ can be done in time polynomial in $\widetilde{n}$.

Lemma 2. *The complexity of* $\mathsf{OneTypeSampler}(\cdot,\cdot,\cdot,\cdot)$ *in Algorithm 1 is polynomial in the size of the input domain.*

Proof. For each block B_{β^i}, there are totally $|\mathcal{T}_{|B_{\beta^i}|,N_u}|$ possible cell configurations in B_{β^i}. Traversing over all blocks in the loop at line 3 in Algorithm 1 will enumerate a total of $\prod_{i=1}^{2^m}|\mathcal{T}_{|B_{\beta^i}|,N_u}|$ possible cell configurations. Even though this number may appear daunting, it is polynomial in the domain size by the definition of configuration space. The remaining complexity of the algorithm is derived from the computation of $\mathcal{W}_{\mathbf{n}}$, which has been shown to be polynomial-time in $\sum_{i\in[N_c]} n_i$. Finally, the value of $\sum_{i\in[N_c]} n_i$ is equal to the domain size, completing the proof. □

3.4.2 Domain Recursive Sampling

Now, let us consider the sampling problem of $\mathcal{P}_\Gamma$-structures conditional on the sampled 1-types η_i's. We rewrite the sampling probability $\mathbb{P}[\mathcal{A} \mid \widehat{\Gamma} \wedge \bigwedge_{i\in[n]} \tau_i(e_i)]$ as $\mathbb{P}[\mathcal{A} \mid \Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i)]$ for better presentation.

Let $\pi_{i,j}$ be the 2-table of the element tuple (e_i, e_j), and denote by $\mathcal{A}_i$ the set of ground 2-tables over all element tuples involved in the element e_i:

$$\mathcal{A}_i := \bigcup_{j\in[n]:j\neq i} \pi_{i,j}(e_i, e_j).$$

Following the idea of domain recursion, we select an element e_t from Δ and decompose the sampling probability as

$$\mathbb{P}\left[\mathcal{A} \mid \Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i)\right] = \mathbb{P}\left[\mathcal{A} \mid \Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i) \wedge \mathcal{A}_t\right] \cdot \mathbb{P}\left[\mathcal{A}_t \mid \Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i)\right].$$

We first demonstrate that for any valid substructure $\mathcal{A}_t$ of the sampling problem, the WFOMS specified by the probability $\mathbb{P}[\mathcal{A} \mid \Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i) \wedge \mathcal{A}_t]$ can be reduced to a WFOMS of the same form as the original problem on $\Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i)$, but over a smaller domain $\Delta' = \Delta \setminus e_t$.

Given a 2-table π and a block type β, let $\beta_{|\pi}$ be a new block type:

$$\beta_{|\pi} = \beta \setminus \{Z_k(x) \mid k \in [m] : R_k(y,x) \in \pi(x,y)\}$$

We call $\beta_{|\pi}$ the *relaxed* block type of β under π, as it removes a part of the existential constraint that is already satisfied by the relations in π. We can also apply the relaxation under π on a cell type $\eta = (\beta, \tau)$, resulting in $\eta_{|\pi} = (\beta_{|\pi}, \tau)$. Let

$$\widetilde{\Gamma} = \Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i) \wedge \mathcal{A}_t, \tag{7}$$

and

$$\widetilde{\Gamma}' = \Gamma_T \wedge \bigwedge_{i\in[n]\setminus\{t\}} \eta_{i|\pi_{t,i}}(e_i). \tag{8}$$

We have the following lemma.

Lemma 3. *If $\mathcal{A}_t$ is valid w.r.t. the WFOMS of $(\Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i), \Delta, w, \bar{w})$, i.e., $\widetilde{\Gamma}$ is satisfiable, the reduction from the WFOMS of $(\widetilde{\Gamma}, \Delta, w, \bar{w})$ to $(\widetilde{\Gamma}', \Delta', w, \bar{w})$ is sound.*

Proof. Let $\mathfrak{S}_1$ and $\mathfrak{S}_2$ be the WFOMS of $(\widetilde{\Gamma}, \Delta, w, \bar{w})$ and $(\widetilde{\Gamma}', w, \Delta', \bar{w})$ respectively. Both of $\mathfrak{S}_1$ and $\mathfrak{S}_2$ have $\mathcal{P}_\Gamma$ as their skeleton. Thus, the problem of $\mathfrak{S}_1$ (resp. $\mathfrak{S}_2$) is equivalent to sampling a $\mathcal{P}_\Gamma$-structure over the domain Δ (resp. Δ'). The mapping function in the reduction can be defined on $\mathcal{P}_\Gamma$-structures over Δ'. We argue that the mapping function is $f(\mathcal{A}') = \mathcal{A}' \cup \mathcal{A}_t \cup \tau_t(e_t)$. The function f is clearly deterministic and polynomial-time.

To simplify the rest arguments of the proof, we will first show that f is bijective, i.e., for any valid $\mathcal{P}_\Gamma$-structure $\mathcal{A}$ of $\mathfrak{S}_1$, there exists a unique valid structure $\mathcal{A}'$ of $\mathfrak{S}_2$ such that $f(\mathcal{A}') = \mathcal{A}$. Let the respective structure to be $\mathcal{A}' = \mathcal{A} \setminus \mathcal{A}_t \setminus \{\tau_t(e_t)\}$, and the uniqueness is clear. Next, we demonstrate that $\mathcal{A}'$ is valid w.r.t. $\mathfrak{S}_2$. Ground out $\widetilde{\Gamma}$ into Δ:

$$\mathcal{A}_t \wedge \bigwedge_{i\in[n]} \eta_i(e_i) \wedge \bigwedge_{i,j\in[n]} \psi(e_i, e_j) \wedge \Lambda,$$

where $\Lambda = \bigwedge_{k\in[m]} \bigwedge_{i\in[n]} \left(Z_k(e_i) \Leftrightarrow \bigvee_{j\in[n]} R_k(e_i, e_j) \right)$. By replacing the ground Tseitin atoms in cell types $\eta_i(e_i)$ in Λ with their corresponding truth assignments and then discarding Λ, we obtain a ground formula without Tseitin atoms:

$$\mathcal{A}_t \wedge \bigwedge_{i\in[n]} \tau_i(e_i) \wedge \bigwedge_{i,j\in[n]} \psi(e_i, e_j) \wedge \bigwedge_{i\in[n]} \bigwedge_{\substack{k\in[m]:\\ Z_k(x)\in\beta_i}} \bigvee_{j\in[n]} R_k(e_i, e_j). \tag{9}$$

It can be easily shown that $\mathcal{A}$ is valid w.r.t. $\mathfrak{S}_1$, iff $\mathcal{A}$ satisfies the formula (9). It follows that the structure $\mathcal{A}'$ satisfies the following formula

$$\bigwedge_{i\in[n]\setminus\{t\}} \tau_i(e_i) \wedge \bigwedge_{i,j\in[n]\setminus\{t\}} \psi(e_i, e_j) \wedge \bigwedge_{i\in[n]\setminus\{t\}} \bigwedge_{\substack{k\in[m]:\\ Z_k(x)\in\beta_{i|\pi_{t,i}}}} \bigvee_{j\in[n]\setminus\{t\}} R_k(e_i, e_j), \tag{10}$$

which is obtained from (9) by substituting the ground atoms in $\mathcal{A}_t$ and $\tau_t(e_t)$. The formula (10) is nothing else but the grounding of $\widetilde{\Gamma}'$ over Δ' followed by the same replacement of ground Tseitin atoms in the relaxed cell types $\eta_{i|\pi_{t,i}}$'s. So we can conclude that $\mathcal{A}'$ is also valid w.r.t. $\mathfrak{S}_2$.

Now, we are prepared to demonstrate the consistency of sampling probability through the mapping function. Since f is bijective, it remains to be shown that

$$\mathbb{P}[f(\mathcal{A}') \mid \widetilde{\Gamma}; \Delta, w, \bar{w}] = \mathbb{P}[\mathcal{A}' \mid \widetilde{\Gamma}'; \Delta', w, \bar{w}]$$

for any valid structure $\mathcal{A}'$ of $\mathfrak{S}_2$. By the definition of the mapping function f, we have

$$\langle w, \bar{w}\rangle(f(\mathcal{A}')) = \langle w, \bar{w}\rangle(\mathcal{A}') \cdot \langle w, \bar{w}\rangle(\mathcal{A}_t) \cdot \langle w, \bar{w}\rangle(\tau_t).$$

Moreover, due to the bijection of f and the fact that $\mathcal{P}_\Gamma$ is a skeleton of $\mathfrak{S}_1$ and $\mathfrak{S}_2$, we have

$$\begin{aligned}\mathsf{WFOMC}(\widetilde{\Gamma}, \Delta, w, \bar{w}) &= \sum_{\mu \in \mathcal{M}_{\widetilde{\Gamma},\Delta}} \langle w, \bar{w}\rangle(\langle \mu \rangle_{\mathcal{P}_\Gamma}) \\ &= \sum_{\mu' \in \mathcal{M}_{\widetilde{\Gamma}',\Delta'}} \langle w, \bar{w}\rangle(f(\langle \mu' \rangle_{\mathcal{P}_\Gamma})) \\ &= \langle w, \bar{w}\rangle(\mathcal{A}_t) \cdot \langle w, \bar{w}\rangle(\tau_t) \cdot \sum_{\mu' \in \mathcal{M}_{\widetilde{\Gamma}',\Delta'}} \langle w, \bar{w}\rangle(\langle \mu' \rangle_{\mathcal{P}_\Gamma}) \\ &= \langle w, \bar{w}\rangle(\mathcal{A}_t) \cdot \langle w, \bar{w}\rangle(\tau_t) \cdot \mathsf{WFOMC}(\widetilde{\Gamma}', \Delta', w, \bar{w}).\end{aligned} \tag{11}$$

Finally, by the definition of conditional probability, we can write

$$\begin{aligned}\mathbb{P}[f(\mathcal{A}') \mid \widetilde{\Gamma}; \Delta, w, \bar{w}] &= \frac{\langle w, \bar{w}\rangle(f(\mathcal{A}'))}{\mathsf{WFOMC}(\widetilde{\Gamma}, \Delta, w, \bar{w})} \\ &= \frac{\langle w, \bar{w}\rangle(\mathcal{A}') \cdot \langle w, \bar{w}\rangle(\mathcal{A}_t) \cdot \langle w, \bar{w}\rangle(\tau_t)}{\langle w, \bar{w}\rangle(\mathcal{A}') \cdot \langle w, \bar{w}\rangle(\mathcal{A}_t) \cdot \mathsf{WFOMC}(\widetilde{\Gamma}', \Delta', w, \bar{w})} \\ &= \frac{\langle w, \bar{w}\rangle(\mathcal{A}')}{\mathsf{WFOMC}(\widetilde{\Gamma}', \Delta', w, \bar{w})} \\ &= \mathbb{P}[\mathcal{A}' \mid \widetilde{\Gamma}'; \Delta', w, \bar{w}],\end{aligned} \tag{12}$$

and thus complete the proof. □

With the sound reduction presented above, what remains to the algorithm is the sampling of $\mathcal{A}_t$ given the probability $\mathbb{P}\left[\mathcal{A}_t \mid \Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i)\right]$.

Recall that $\mathcal{A}_t$ consists of the ground 2-tables of all tuples comprising e_t and the elements in Δ'. We follow a similar approach as in the cell type sampling and accomplish the sampling of $\mathcal{A}_t$ through random partitions on cells. Let $C_{\eta^1}, C_{\eta^2}, \ldots, C_{\eta^{N_c}}$ be the cell partition of Δ' corresponding to the sampled cell types $\eta_1, \eta_2, \ldots, \eta_{n-1}$. Let N_b be the number of all 2-tables, and fix the linear order of 2-tables $\pi^1, \pi^2, \ldots, \pi^{N_b}$. Any substructure $\mathcal{A}_t$ can be viewed as partitions on each cell into N_b disjoint subsets; each subset corresponds to a 2-table π^j and precisely contains the elements that realize π^j in combination with e_t.

Given a substructure $\mathcal{A}_t$, we use $\left\{G^{\mathcal{A}_t}_{\eta^i,\pi^j}\right\}_{j\in[N_b]}$ to denote the refined partition on C_{η^i}, and $\mathbf{g}^{\mathcal{A}_t}_{\eta^i} = \left(|G^{\mathcal{A}_t}_{\eta^i,\pi^j}|\right)_{j\in[N_b]}$ its corresponding cardinality vector. Let $\mathbf{g}^{\mathcal{A}_t} = \bigoplus_{i\in[N_c]} \mathbf{g}^{\mathcal{A}_t}_{\eta^i}$ be the concatenation of cardinality vectors over all cells, which is called the *2-table configuration* of $\mathcal{A}_t$.

We first assume that $\mathcal{A}_t$ is valid in the sampling problem $(\Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i), \Delta, w, \bar{w})$. It will turn out that the sampling probability of $\mathcal{A}_t$ is completely determined by its corresponding 2-table configuration $\mathbf{g}^{\mathcal{A}_t}$. To begin with, as stated by (11), we can write the sampling weight $\mathsf{WFOMC}(\Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i) \wedge \mathcal{A}_t, \Delta, w, \bar{w})$ as

$$\mathsf{WFOMC}(\widetilde{\Gamma}', \Delta', w, \bar{w}) \cdot \langle w, \bar{w}\rangle(\tau_t) \cdot \langle w, \bar{w}\rangle(\mathcal{A}_t) \tag{13}$$

where $\widetilde{\Gamma}'$, defined as (8), is the reduced sentence by the 2-tables in $\mathcal{A}_t$. Let $\mathbf{n}^{\mathcal{A}_t}$ be the cell configuration corresponding to the ground cells in $\widetilde{\Gamma}'$. The value of $\mathsf{WFOMC}(\widetilde{\Gamma}', \Delta', w, \bar{w})$ is exactly $\mathcal{W}_{\mathbf{n}^{\mathcal{A}_t}}$, which is formally defined in Section 3.4.1. Denote by $\mathbf{w} = \big(\langle w, \bar{w}\rangle(\pi^i)\big)_{i \in N_b}$, the weight vector of 2-tables. We can then write (13) as

$$\mathcal{W}_{\mathbf{n}^{\mathcal{A}_t}} \cdot \langle w, \bar{w}\rangle(\tau_t) \cdot \prod_{i \in [N_c]} \mathbf{w}^{\mathbf{g}_{\eta^i}^{\mathcal{A}_t}}. \tag{14}$$

In the equation above, τ_t has already been decided in the cell type η_t (by $\mathsf{OneTypeSampler}$), and the last term only depends on the 2-table configurations $\mathbf{g}^{\mathcal{A}_t}$. It is easy to check that the cell configuration of $\mathbf{n}^{\mathcal{A}_t}$ is also fully determined by $\mathbf{g}^{\mathcal{A}_t}$. To illustrate this, let $n_\eta^{\mathcal{A}_t}$ be the cardinality of cell type η in $\mathbf{n}^{\mathcal{A}_t}$, and $g_{\eta,\pi}^{\mathcal{A}_t}$ the cardinality of π in $\mathbf{g}_\eta^{\mathcal{A}_t}$, i.e., $|G_{\eta,\pi}^{\mathcal{A}_t}|$. For any cell type η, the value of $n_\eta^{\mathcal{A}_t}$ be can computed by

$$n_\eta^{\mathcal{A}_t} = \sum_{i \in [N_c], j \in [N_b] : {\eta^i}_{|\pi^j} = \eta} g_{\eta^i, \pi^j}^{\mathcal{A}_t}. \tag{15}$$

By the argument above, the sampling probability (14) of a valid substructure $\mathcal{A}_t$ is completely determined by $\mathbf{g}^{\mathcal{A}_t}$. Thus, we can sample $\mathcal{A}_t$, in the same spirit of sampling 1-types in Section 3.4.1, by first sampling a 2-table configuration $\mathbf{g}^{\mathcal{A}_t}$, and then partitioning the cells accordingly. We can then simply apply the enumerative sampling method, as the number of possible 2-table configurations is clearly polynomial in the domain size. For any 2-table configuration $\mathbf{g}^{\mathcal{A}_t}$, its sampling weight can be computed by multiplying (14) by $\prod_{i \in [N_c]} \binom{n_{\eta^i}}{\mathbf{g}_{\eta^i}^{\mathcal{A}_t}}$, where n_{η^i} is the size of C_{η^i}.

So far in our discussion, we have been always assuming that the substructure $\mathcal{A}_t$ is valid in the WFOMS $(\Gamma_T \wedge \bigwedge_{i \in [n]} \eta_i(e_i), \Delta, w, \bar{w})$, or it should not be sampled. We guarantee this assumption by imposing some constraints on the 2-table configuration $\mathbf{g}^{\mathcal{A}_t}$. We call a 2-table π *coherent* with a 1-types tuple (τ, τ') if, for some domain elements a and b, the interpretation of $\tau(a) \cup \pi(a,b) \cup \tau'(b)$ satisfies the formula $\psi(a,b) \wedge \psi(b,a)$. Then, the first constraint is that any 2-table $\pi_{t,i}$ in $\mathcal{A}_t$ must be coherent with τ_t and τ_i. This translates to a requirement on 2-table configuration that, when partitioning a cell η^i, the cardinality of 2-tables that are not coherent with τ_t and τ_i is restricted to be 0. The second constraint is that, for any index $k \in \{i \mid Z_i \in \beta_t\}$, the substructure $\mathcal{A}_t$ must contain at least one ground atom of the form $R_k(e_t, a)$, where a is a domain element from Δ, to make $\mathcal{A}_t$ satisfy the existential formula $\exists y : R_k(e_t, y)$. This means that there must be at least one nonzero cardinality in the 2-table configuration such that its corresponding 2-table π satisfies $R_k(x,y) \in \pi$.

By combining all the ingredients discussed above, we now present our sampling algorithm for the sentence Γ_T conditionally on the cell types η_i's, as shown in Algorithm 2. The overall structure of the algorithm follows a recursive approach, where a recursive call with a smaller domain and relaxed cell types is invoked at Line 33. The algorithm terminates when the input domain contains a single element (at Line 1) or there are no existential constraints on the elements (at Line 4). In Lines 10-23, all possible 2-table configurations are enumerated. For each configuration, we compute its corresponding weight in Lines 13-15 and decide whether it should be sampled in Lines 16-21. When the 2-table configuration has been sampled, we randomly partition the cells in Lines 25-32, and then update the sampled structure and the cell type of each element respectively at Line 29 and 30. The function $\mathsf{ExSat}(\mathbf{g}, \eta)$ at Line 12 is used to check whether the 2-table configuration $\mathbf{g}$ guarantees the validity of the sampled substructures, as discussed above. The pseudo-code for this function is presented in Appendix A.5.2.

Algorithm 2 $\mathsf{DRSampler}(\Gamma_T, \Delta, w, \bar{w}, (\eta_i)_{i\in[n]})$

```
if $n = 1$ then
    return $\emptyset$
end if
if only the block of type $\beta = \top$ is nonempty then
    return a model $\mu$ sampled by a $\mathbf{UFO}^2$ WMS from $\forall x \forall y : \psi(x,y) \wedge \bigwedge_{i\in[n]} \tau_i(e_i)$ over $\Delta$ under $(w, \bar{w})$
end if
Choose $t \in [n]$; $\mu \leftarrow \tau_t(e_t)$; $\Delta' \leftarrow \Delta \setminus \{e_t\}$
Get the cell configuration $\mathbf{n} = \left(n_{\eta^i}\right)_{i\in[N_c]}$ of $(\eta_i)_{i\in[n]\setminus\{t\}}$
$W \leftarrow \mathcal{W}_{\mathbf{n}}$
for $\left(\mathbf{g}_{\eta^i}\right)_{i\in[N_c]} \leftarrow \mathsf{Prod}(\mathcal{T}_{n_{\eta^1},N_b}, \ldots, \mathcal{T}_{n_{\eta^{N_c}},N_b})$ do
    $\mathbf{g} \leftarrow \bigoplus_{i\in[N_c]} \mathbf{g}_{\eta^i}$
    if $\mathsf{ExSat}\left(\mathbf{g}, \eta_t\right)$ then
        Get the new cell configuration $\mathbf{n'}$ w.r.t $\mathbf{g}$ by (15)
        Compute $\mathcal{W}_{\mathbf{n'}}$ by (6)
        $W' \leftarrow \mathcal{W}_{\mathbf{n'}} \cdot \langle w, \bar{w}\rangle(\tau_t) \cdot \prod_{i\in[N_c]} \binom{n_{\eta^i}}{\mathbf{g}_{\eta^i}} \mathbf{w}^{\mathbf{g}_{\eta^i}}$
        if $\mathsf{Uniform}(0,1) < \frac{W'}{W}$ then
            $\mathbf{g}^* \leftarrow \mathbf{g}$
            break
        else
            $W \leftarrow W - W'$
        end if
    end if
end for
Obtain the cell partition $\{C_{\eta^i}\}_{i\in[N_c]}$ from $(\eta_i)_{i\in[n]\setminus\{t\}}$
for $i \in [N_c]$ do
    Fetch the cardinality vector $\mathbf{g}^*_{\eta^i}$ of $\eta^i$ from $\mathbf{g}^*$
    Randomly partition the cell $C_{\eta^i}$ into $\left\{G_{\eta^i,\pi^j}\right\}_{j\in[N_b]}$ according to $\mathbf{g}^*_{\eta^i}$
    for $j \in [N_b]$ do
        $\mu \leftarrow \mu \cup \left\{\pi^j(e_t, e)\right\}_{e\in G_{\eta^i,\pi^j}}$
        $\forall e_s \in G_{\eta^i,\pi^j}, \eta'_s \leftarrow \eta_{s|\pi^j}$
    end for
end for
$\mu \leftarrow \mu \cup \mathsf{DRSampler}(\Gamma_T, \Delta', w, \bar{w}, (\eta'_i)_{i\in[n-1]})$
return $\mu$
```

Lemma 4. *The complexity of* $\mathsf{DRSampler}(\cdot,\cdot,\cdot,\cdot)$ *in Algorithm 2 is polynomial in the size of the input domain.*

Proof. The algorithm $\mathsf{DRSampler}$ is called at most n times, where n is the size of the domain. The main computation of each recursive call is for the loop, where we need to iterate over all $\prod_{i\in[N_c]} |\mathcal{T}n_{\eta^i}, N_b|$ possible configurations. The size of a configuration space $\mathcal{T}_{M,m}$ is polynomial in M, and thus the complexity of this loop is also polynomial in the domain size. The other complexity of computing $\mathcal{W}_{\mathbf{n'}}$ has been shown to be polynomial in the summation over the vector

$\mathbf{n}'$, which is clearly smaller than the domain size. □

3.4.3 A Lifted WMS for $\mathbf{FO}^2$

We present our WMS for $\mathbf{FO}^2$ in Algorithm 3. Given a $\mathbf{FO}^2$ sentence Γ in SNF, the algorithm first obtains the sentence Γ_T of the general form (5). Then the algorithms OneTypeSampler and DRSampler then applied successively to sample a skeleton structure of $\widehat{\Gamma}$. It is easy to verify that the skeleton structure is also a model of Γ, as it can be regarded as the output of the mapping function in the sound reduction from the WFOMS problem on Γ to $\widehat{\Gamma}$. Since both OneTypeSampler and DRSampler have been proved to be polynomial-time in the domain size by Lemma 2 and 4, the WMS in Algorithm 3 is clearly lifted.

Algorithm 3 $\mathsf{WMS}(\Gamma, \Delta, w, \bar{w})$

INPUT: An $\mathbf{FO}^2$ sentence Γ of the form (3), a domain $\Delta = \{e_i\}_{i\in[n]}$ of size n, a weighting $(w, \bar{w})$
OUTPUT: A model μ of Γ over Δ

```
Construct $\Gamma_T$ from $\Gamma$ by (4)
$\forall i \in [n], \beta_i(x) \leftarrow \bigwedge_{k\in[m]} Z_k(x)$
$\widehat{\Gamma} \leftarrow \Gamma_T \wedge \bigwedge_{i\in[n]} \beta_i(e_i)$
$(\tau_i)_{i\in[n]} \leftarrow \mathsf{OneTypeSampler}(\widehat{\Gamma}, \Delta, w, \bar{w})$
$\forall i \in [n], \eta_i \leftarrow (\beta_i, \tau_i)$
$\mu \leftarrow \mathsf{DRSampler}(\Gamma_T, \Delta, w, \bar{w}, (\eta_i)_{i\in[n]})$
return $\mu$
```

Theorem 2. *The fragment* $\mathbf{FO}^2$ *is domain-liftable under sampling.*

Proof of Theorem 2. The proof is directly following from the above and from Lemma 1. □

Remark 1. *We note that there are several optimizations to our WMS, e.g., heuristically selecting the domain element in DRSampler so that the algorithm can quickly reach the terminal condition. However, the current algorithm is clear and efficient enough to prove our main result, so that we leave the discussion on some of the optimizations to Appendix A.5.1.*

4 A Genralization to $\mathbf{FO}^2$ with Cardinality Constraints

In this section, we extend our results to $\mathbf{FO}^2$ with *cardinality constraints*. A single cardinality constraint is a statement of the form $|P| \bowtie q$, where $\bowtie$ is a comparison operator (e.g., $=$, $\leq$, $\geq$, $<$, $>$) and q is a natural number. These constraints are imposed on the number of distinct positive ground literals in a structure $\mathcal{A}$ formed by the predicate P. For example, a structure $\mathcal{A}$ satisfies the constraint $|P| \leq q$ if there are at most q literals for P that are true in $\mathcal{A}$. For illustration, we allow cardinality constraints as atomic formulas in the FO formulas, e.g., $(|E| = 2) \wedge (\forall x \forall y : E(x,y) \Rightarrow E(y,x))$ is a valid formula (its models can be interpreted as undirected graphs with exactly one edge) and the satisfaction relation $\models$ is extended naturally.

The cardinality constraints are not necessarily expressible in FO logic without grounding out the constraint over the domain, and have a strong connection to the fragment of $\mathbf{C}^2$, which will be introduced in the next section. In [1], the authors also extended their WMS (which was originally developed for $\mathbf{UFO}^2$) to handle cardinality constraints. However, their method was relatively straightforward whereas the extension to $\mathbf{FO}^2$ is more complicated.

Let Γ be an $\mathbf{FO}^2$ sentence and

$$\Upsilon := \varphi(|P_1| \bowtie q_1, \dots, |P_M| \bowtie q_M), \tag{16}$$

where φ is a Boolean formula, $\{P_i\}_{i\in[M]} \subseteq \mathcal{P}_\Gamma$, and $\forall i \in [M], q_i \in \mathbb{N}$. Consider the WFOMS problem on $\Gamma \wedge \Upsilon$ over the domain Δ under $(w, \bar{w})$. The overall structure of the sampling algorithm for $\Gamma \wedge \Upsilon$ remains unchanged from Algorithm 3. The algorithm still begins with obtaining the general sentence $\widehat{\Gamma}$ from Γ. Then the formula $\widehat{\Gamma} \wedge \Upsilon$ is fed into $\mathsf{OneTypeSampler}$ and $\mathsf{DRSampler}$ successively to sample the $\mathcal{P}_\Gamma$-structure. Please refer to Appendix A.5.3 for the detailed algorithm. We only describe its modifications to the original one below.

The algorithm of $\mathsf{OneTypeSampler}$ for $\widehat{\Gamma} \wedge \Upsilon$ is similar to Algorithm 1, with the main difference being that the computation of WFOMC problems, specifically $\mathsf{WFOMC}(\widehat{\Gamma}, \Delta, w, \bar{w})$ and $\mathcal{W}_{\mathbf{n}}$, now include cardinality constraints Υ in their input sentences. To account for this change, we slightly modify the definition of $\mathcal{W}_{\mathbf{n}}$ in (6) by taking $\Gamma_T \wedge \bigwedge_{i\in[\widetilde{n}]} \widetilde{\eta}_i \wedge \Upsilon$ as input, and denote the new term by $\mathcal{W}_{\mathbf{n},\Upsilon}$. According to Proposition 5 in [7], the addition of cardinality constraints to a liftable sentence does not affect the liftability of the resulting formula (in terms of WFOMC problems). Therefore, the computation of $\mathcal{W}_{\mathbf{n},\Upsilon}$ remains polynomial-time in the domain size, as the original sentence in $\mathcal{W}_{\mathbf{n}}$ was already proven to be liftable.

For the sampling problem conditional on the sampled cell types η_i's, the domain recursive property still holds as we will show in turn. Given a set L of ground literals and a predicate P, let $N(P, L)$ denote the number of positive ground literals for P in L. Given a valid substructure $\mathcal{A}_t$ of the element e_t, denote the 1-type of e_t by τ_t as usual, let $q'_i = q_i - N(P_i, \mathcal{A}_t) - N(P_i, \tau_t(e_t))$ for every $i \in [M]$, and define

$$\Upsilon' = \varphi(|P_1| \bowtie q'_1, \dots, |P_M| \bowtie q'_M). \tag{17}$$

Let $\widetilde{\Gamma}_C = \widetilde{\Gamma} \wedge \Upsilon$ and $\widetilde{\Gamma}'_C = \widetilde{\Gamma}' \wedge \Upsilon'$, where $\widetilde{\Gamma}$ and $\widetilde{\Gamma}'$ are defined as (7) and (8) respectively. Then the reduction from the WFOMS problem on $\widetilde{\Gamma}$ to $\widetilde{\Gamma}'$ is sound.

Lemma 5. *If $\mathcal{A}_t$ is valid w.r.t. the WFOMS of $(\bigwedge_{i\in[n]} \eta_i(e_i) \wedge \Gamma_T \wedge \Upsilon, \Delta, w, \bar{w})$, i.e., $\widetilde{\Gamma}_C$ is satisfiable, the reduction from the WFOMS of $(\widetilde{\Gamma}_C, \Delta, w, \bar{w})$ to $(\widetilde{\Gamma}'_C, \Delta', w, \bar{w}')$ is sound.*

Proof of Lemma 5. The proof follows the same argument for Lemma 3. The only statement that needs to be argued again is the bijection of the mapping function $f(\mathcal{A}') = \mathcal{A}' \cup \mathcal{A}_t \cup \tau_t(e_t)$. Let $\mathfrak{S}_1^C$ and $\mathfrak{S}_2^C$ be the WFOMS of $(\widetilde{\Gamma}_C, \Delta, w, \bar{w})$ and $(\widetilde{\Gamma}'_C, \Delta', w, \bar{w})$. For any valid $\mathcal{A}$ of $\mathfrak{S}_1^C$, $\mathcal{A}$ must satisfy both $\widetilde{\Gamma}$ and Υ. It follows that $\mathcal{A}' = \mathcal{A} \setminus \mathcal{A}_t \setminus \{\tau_t(e_t)\}$ satisfies $\widetilde{\Gamma}'$ and Υ', meaning that $\mathcal{A}'$ is also valid w.r.t. $\mathfrak{S}_2^C$. This establishes the bijection of f. The remainder of the proof, including the consistency of sampling probability, proceeds exactly the same as Lemma 3, specifically follows (11) and (12). □

The core structure of $\mathsf{DRSampler}$ remains the same as the sound reduction still holds. However, the recursive call is now made with the reduced sentence $\Gamma_T \wedge \Upsilon'$. The other slight modifications include:

- $\mathcal{W}_{\mathbf{n}^{\mathcal{A}_t}}$ in (14) has been replaced with $\mathcal{W}_{\mathbf{n}^{\mathcal{A}_t},\Upsilon'}$, and

- the validity check for the sampled 2-table configuration in ExSat now includes an additional check for the well-definedness of the reduced cardinality constraints Υ', returning False if any $q_i' \notin \mathbb{N}$ for $i \in [M]$.

As discussed above, the extension of our sampling algorithm to handle cardinality constraints in $\mathbf{FO}^2$ only slightly increases the complexity of the procedure. Furthermore, the computation of $\mathcal{W}_{\mathbf{n},\Upsilon}$ remains polynomial-time in the domain size, meaning that the generalized algorithm is still lifted, and thus proving the liftability under sampling of $\mathbf{FO}^2$ with cardinality constraints [4].

Theorem 3. *Let Γ be an $\mathbf{FO}^2$ sentence and Υ of the form* (16)*. Then $\Gamma \wedge \Upsilon$ is domain-liftable under sampling.*

Proof. The proof follows from the discussion above. □

5 A Further Generalization to $\mathbf{SC}^2$

With the lifted WMS for $\mathbf{FO}^2$ with cardinality constraints, we can further extend our result to the case involving the counting quantifiers $\exists_{=k}$ [24]. Here, we study the sentences of the form

$$\begin{aligned}\Gamma \wedge (\forall x \exists_{=k_1} y : \phi_1(x, y)) \wedge \cdots \wedge (\forall x \exists_{=k_{M'}} : \phi_{M'}(x, y))\\ \wedge (\exists_{=k_1'} x \forall y : \phi_1'(x, y)) \wedge \cdots \wedge (\exists_{=k_{M''}'} x \forall y : \phi_{M''}'(x, y)),\end{aligned}$$

where Γ is an $\mathbf{FO}^2$ sentence and $\exists_{=k}$ is the counting quantifier that specifies the exact number of elements in the domain that satisfy a given formula. For instance, a structure $\mathcal{A}$ over a domain Δ satisfies the sentence $\exists_{=k} x : \psi(x)$, if there are exactly k distinct elements $t_1, \ldots, t_k \in \Delta$ such that $\mathcal{A} \models \psi(t_i)$ for all $i \in [k]$. We call this fragment *two-variable logic with counting in SNF* $\mathbf{SC}^2$, as its extended conjunction to $\mathbf{FO}^2$ sentences resembles SNF. The presence of counting quantifiers significantly enhances the expressiveness of $\mathbf{SC}^2$, e.g., k-regular graphs can be encoded in $\mathbf{SC}^2$, as demonstrated in the introduction.

Recently, Kuzelka [7] showed that the liftability of $\mathbf{FO}^2$ can be generalized to *the fragment of two-variable logic with counting* $\mathbf{C}^2$, a superset of $\mathbf{SC}^2$, by reducing the WFOMC problem on $\mathbf{C}^2$ sentences to $\mathbf{FO}^2$ sentences with cardinality constraints. We demonstrate that this reduction can be also applied to the sampling problem and it is sound, when the fragment is restricted to be $\mathbf{SC}^2$.

Lemma 6. *For any WFOMS $\mathfrak{S} = (\Phi, \Delta, w, \bar{w})$ where Φ is a $\mathbf{SC}^2$ sentence, there exists a WFOMS $\mathfrak{S}' = (\Gamma' \wedge \Upsilon, \Delta, w', \bar{w}')$, where Γ' is an $\mathbf{FO}^2$ sentence, Υ denotes cardinality constraints of the form* (16) *and both Γ' and Υ are independent of Δ, such that the reduction from $\mathfrak{S}$ to $\mathfrak{S}'$ is sound.*

The proof follows a similar technique used in [7], and the details are deferred to Appendix A.2. We note here that further generalizing this result to the general $\mathbf{C}^2$ sentences is infeasible, since the original reduction used in [7] for $\mathbf{C}^2$ sentences introduced some negative weights on predicates. However, it can be established that the domain-liftability under sampling of $\mathbf{C}^2$ can

[4] It is worth noting that the computational complexity of $\mathcal{W}_{\mathbf{n},\Upsilon}$ is independent of the values $q_1, \ldots, q_M$ in the cardinality constraints Υ, so that the reduction on these constraints does not affect the liftability of the reduced $\mathcal{W}_{\mathbf{n}^{\mathcal{A}_t},\Upsilon'}$.

be demonstrated by directly applying our domain recursion sampling method without resorting to the reduction to cardinality constraints. For a more detailed discussion, please refer to Appendix A.3.

Since $\mathbf{FO}^2$ with cardinality constraints has been proved to be liftable under sampling, it is easy to prove the liftability under sampling of the $\mathbf{SC}^2$ fragment.

Theorem 4. *The fragment of* $\mathbf{SC}^2$ *is domain-liftable under sampling.*

Proof. The proof follows from Lemma 6 and Theorem 3. □

Moreover, one can further introduce additional cardinality constraints into $\mathbf{SC}^2$ without degrading its liftability under sampling.

Corollary 1. *Let* Φ *be a* $\mathbf{SC}^2$ *sentence and* Υ *of the form* (16)*. Then* $\Phi \wedge \Upsilon$ *is domain-liftable under sampling.*

6 Experimental Results

We conducted several experiments to evaluate the performance and correctness of our sampling algorithms. All algorithms were implemented in Python and the experiments were performed on a computer with an 8-core Intel i7 3.60GHz processor and 32 GB of RAM.

Many sampling problems can be expressed as WFOMS problems. Here we consider two typical ones.

- **Sampling combinatorial structures**: the uniform generation of some combinatorial structures can be directly reduced to a WFOMS, e.g., the uniform generation of *graphs with no isolated vertices* and *k-regular graphs* in Examples 1 and the introduction. We added four more combinatorial sampling problems to these two for evaluation: *functions*, *functions w/o fix-points* (i.e., the functions f satisfying $f(x) \neq x$), *permutations* and *permutations w/o fix-points*. The details of these problems are described in Appendix A.4.1.

- **Sampling from MLNs**: our algorithms can be also applied to sample possible worlds from MLNs. An MLN defines a distribution over structures (i.e., possible worlds in SRL literature), and its respective sampling problem is to randomly generate possible worlds according to this distribution. There is a standard reduction from the sampling problem of an MLN to a WFOMS problem (see Append A.4.1 and also [1]). We used two MLNs in our experiments: 1) A variant of the classic *friends-smokers* MLN with the constraint that every person has at least one friend:

$$\{(\infty, \neg fr(x,x)), (\infty, fr(x,y) \Rightarrow fr(y,x)), (0, sm(x)),$$
$$(0.2, fr(x,y) \wedge sm(x) \Rightarrow sm(y)), (\infty, \exists y : fr(x,y))\}.$$

  2) The *employment* MLN used in [4]:

$$\{(1.3, \exists y : workfor(x,y) \vee boss(x))\},$$

  which states that with high probability, every person either is employed by a boss or is a boss. The details about the reduction from sampling from MLNs to WFOMS and the corresponding WFOMS problems of these two MLNs can be found in Appendix A.4.1.

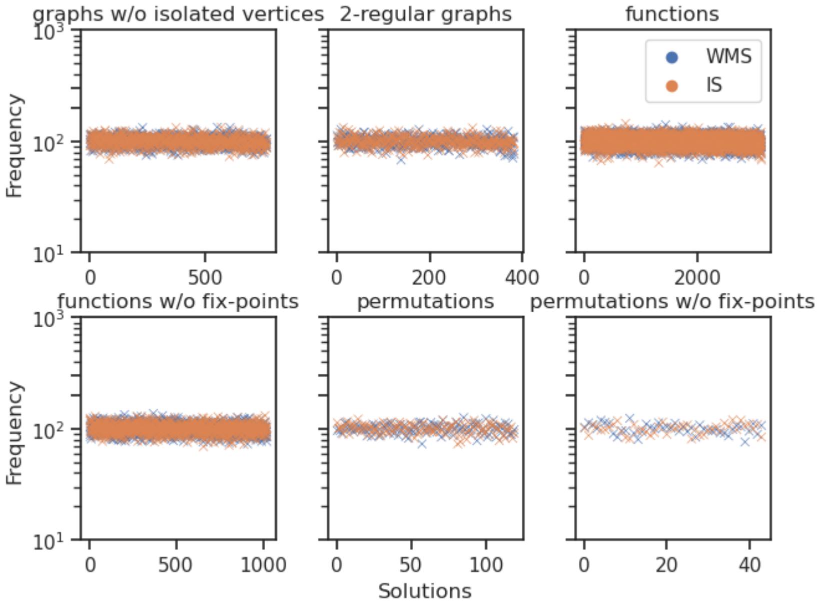
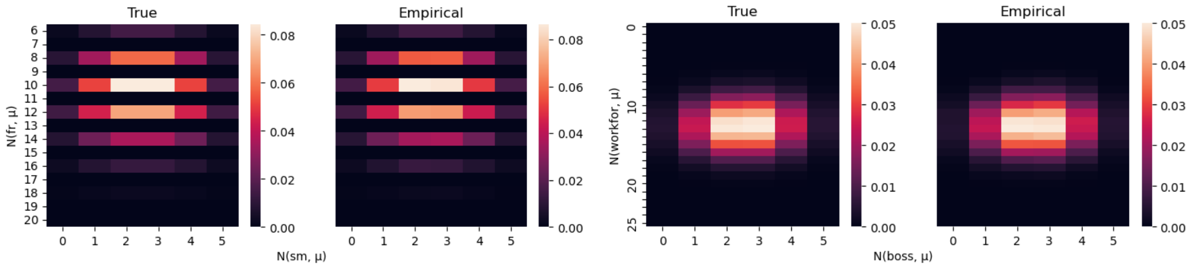


Figure 2: Uniformity comparison between an ideal sampler (IS) and our WMS.



(a) friends-smokers

(b) employment

Figure 3: Conformity testing for the count distribution of MLNs.

6.1 Correctness

We first examine the correctness of our implementation on the uniform generation of combinatorial structures over small domains, where exact sampling is feasible via enumeration-based techniques; we choose the domain size of 5 for evaluation. To serve as a benchmark, we implemented a simple ideal uniform sampler, denoted by IS, by enumerating all the models and then drawing samples uniformly from these models. For each combinatorial structure encoded into an $\mathbf{FO}^2$ sentence Γ, a total of $100 \times |\mathcal{M}_{\Gamma,\Delta}|$ models were generated from both IS and our WMS. Figure 2 depicts the model distribution produced by these two algorithms—the horizontal axis represents models numbered lexicographically, while the vertical axis represents the generated frequencies of models. The figure suggests that the distribution generated by our WMS is indistinguishable from that of IS. Furthermore, a statistical test on the distributions produced by WMS was performed, and no statistically significant difference from the uniform distribution was found. The details of this test can be found in Appendix A.4.2.

For sampling problems from MLNs, enumerating all the models is infeasible even for a domain

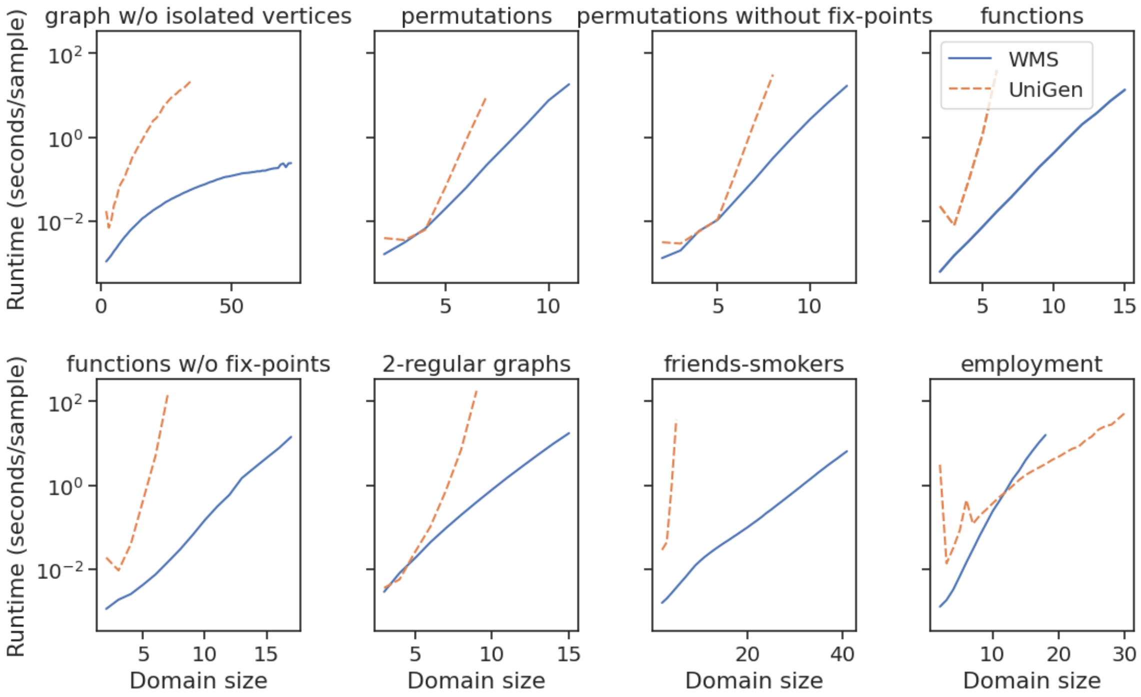


Figure 4: Performance of WMS versus UniGen.

of size 5, e.g., there are $2^{2^5+5} = 2^{37}$ models in the employment MLN. That is why we test the *count distribution* of vocabulary for these two MLNs. Instead of specifying the probability of each model, the count distribution only tells us how probably a certain number of predicates are interpreted to be true in the models. An advantage of testing count distributions is that they can be efficiently computed for our MLNs. Please refer to [7] for more details about count distributions. We also note that the conformity of count distribution is a necessary condition for the correctness of algorithms. We keep the domain size to be 5 and sampled 10^5 models from friends-smokers and employment MLNs respectively. The empirical distributions of count-statistics, along with the true count distributions, are shown in Figure 3. It is easy to check the conformity of the empirical distribution to the true one from the figure. The statistical test was also performed on the count distribution, and the results confirm the conclusion drawn from the figure (also see Appendix A.4.2).

6.2 Performance

To evaluate the performance, we compared our weighted model samplers with Unigen [18, 25], the state-of-the-art approximate sampler for Boolean formulas. A WFOMS problem can be reduced to a sampling problem of the Boolean formula by grounding the input sentence over the given domain. Since Unigen only works for uniform sampling, we employed the technique in [26] to encode the weighting function in the WFOMS problem into a Boolean formula.

For each sampling problem, we randomly generated 1000 models by our WMS and Unigen respectively and computed the average sampling time of one model. The performance comparison is shown in Figure 4. In most cases, our approach is much faster than UniGen. The exception in the employment MLN, where UniGen performed better than WMS, is likely due to the simplicity of this specific instance for its underlying SAT solver. This coincides with the theoretical result that our WMS is polynomial-time in the domain size, while UniGen usually needs amounts of expensive SAT calls on the grounding formulas.

7 Conclusion and Future Work

In this paper, we prove the domain-liftability under sampling of $\mathbf{FO}^2$ by presenting a novel and efficient approach to its symmetric weighted first-order sampling problems. The result is further extended to the fragment of $\mathbf{SC}^2$ with the presence of counting constraints. The widespread applicability of WFOMS renders the proposed approach a promising candidate to serve as a universal paradigm for a plethora of sampling problems.

A potential avenue for further research is to expand the methodology presented in this paper to encompass more expressive first-order languages. Specifically, the utilization of the domain recursion scheme employed in this study could be extended beyond the confines of the $\mathbf{FO}^2$ and $\mathbf{SC}^2$, as its analogous counterpart in WFOMC has been demonstrated to be effective in proving the domain-liftability of the fragments $\mathbf{S}^2\mathbf{FO}^2$ and $\mathbf{S}^2\mathbf{RU}$ [15].

In addition to extending the input logic, other potential directions for future research include incorporating elementary axioms, such as tree axiom [27] and linear order axiom [16], as well as more general weighting functions that involve negative weights. However, it is important to note that these extensions would likely require a more advanced and nuanced approach than the one proposed in this paper, and may present significant challenges.

Finally, the lower complexity bound of WFOMS is also an interesting open problem. A direct implication from the infeasibility of WFOMC in [28] suggests that there is unlikely for an (even approximate) lifted WMS to exist for full first-order logic. However, the establishment of a tighter lower bound for fragments of FO, such as $\mathbf{FO}^3$, remains an unexplored and challenging area that merits further investigation.

A Appendix

A.1 Scott Normal Forms

We briefly describe the transformation of $\mathbf{FO}^2$ formulas to SNF and prove the soundness of its corresponding reduction on the WFOMS problems. The process is well-known, so we only sketch the related details.

Let Γ be a sentence of $\mathbf{FO}^2$. To put it into SNF, consider a subformula $\varphi(x) = Qy : \phi(x, y)$, where $Q \in \{\forall, \exists\}$ and ϕ is quantifier-free. Let A_φ be a fresh unary predicate[5] and consider the sentence

$$\forall x : (A_\varphi(x) \Leftrightarrow (Qy : \phi(x, y)))$$

which states that $\varphi(x)$ is equivalent to $A_\varphi(x)$. Let Q' denote the dual of Q, i.e., $Q' = \{\forall, \exists\}\backslash\{Q\}$, this sentence can be seen equivalent to

$$\begin{aligned}\Gamma' :=& \forall x Q y : (A_\varphi(x) \Rightarrow \phi(x, y)) \\ &\wedge \forall x Q' y : (\phi(x, y) \Rightarrow A_\varphi(x)).\end{aligned}$$

Let

$$\Gamma'' = \Gamma' \wedge \Gamma[\varphi(x)/A_\varphi(x)],$$

where $\Gamma[\varphi(x)/A_\varphi(x)]$ is obtained from Γ by replacing $\varphi(x)$ with $A_\varphi(x)$. For any domain Δ, every model of Γ'' over Δ can be mapped to a unique model of Γ over Δ. The bijective mapping function is simply the projection $\langle\cdot\rangle_{\mathcal{P}_\Gamma}$. Let both the positive and negative weights of A_φ be 1 and denote the new weighting functions as w' and $\bar{w}'$. It is clear that the reduction from $(\Gamma, \Delta, w, \bar{w})$ to $(\Gamma'', \Delta, w', \bar{w}')$ is sound. Repeat this process from the atomic level and work upwards until the sentence is in SNF. The whole reduction remains sound due to the transitivity of soundness.

A.2 A Sound Reduction from $\mathbf{SC}^2$ to $\mathbf{FO}^2$ with Cardinality Constraints

In this section, we show the sound reduction from a WFOMS problem on $\mathbf{SC}^2$ sentence to a WFOMS problem on $\mathbf{FO}^2$ sentence with cardinality constraints.

We first need the following two lemmas.

Lemma 7. *Let Γ be a first-order logic sentence, and let Δ be a domain. Let Φ be a first-order sentence with cardinality constraints, defined as follows:*

$$\begin{aligned}\Pi :=& (|P| = k \cdot |\Delta|) \\ &\wedge (\forall x \forall y : P(x, y) \Leftrightarrow (R_1^P(x, y) \vee \cdots \vee R_k^P(x, y))) \\ &\wedge \bigwedge_{i \in [k]} (\forall x \exists y : R_i^P(x, y)) \\ &\wedge \bigwedge_{i, j \in [k] : i \neq j} (\forall x \forall y : \neg R_i^P(x, y) \vee \neg R_j^P(x, y)),\end{aligned}$$

where R_i^P's are auxiliary predicates not in $\mathcal{P}_\Gamma$ with weight $w(R_i^P) = \bar{w}(R_i^P) = 1$. Then the reduction from the WFOMS $(\Gamma \wedge \forall x \exists_{=k} y : P(x, y), \Delta, w, \bar{w})$ to $(\Gamma \wedge \Pi, \Delta, w, \bar{w})$ is sound.

[5] If $\varphi(x)$ has no free variables, e.g., $\exists x : \phi(x)$, the predicate A_φ is nullary.

Proof. Let $f(\cdot) = \langle \cdot \rangle_{\mathcal{P}_\Gamma \cup \{P\}}$ be a mapping function. We first show that f is from $\mathcal{M}_{\Gamma\wedge\Pi,\Delta}$ to $\mathcal{M}_{\Gamma\wedge\forall x\exists_{=k}:P(x,y),\Delta}$: if $\mathcal{A} \models \Gamma \wedge \Pi$ then $f(\mathcal{A}) \models \Gamma \wedge \forall x \exists_{=k} y : P(x,y)$.

The sentence Π means that for every $c_1, c_2 \in \Delta$ such that $P(c_1, c_2)$ is true, there is exactly one $i \in [k]$ such that $R_i^P(c_1, c_2)$ is true. Thus we have that $\sum_{i\in[k]} |R_i^P| = |P| = k \cdot |\Delta|$, which together with $\bigwedge_{i\in[k]} \forall x \exists y : R_i^P(x,y)$ implies that $|R_i^P| = k$ for $i \in [k]$. We argue that each R_i^P is a function predicate in the sense that $\forall x \exists_{=1} y : R_i^P(x,y)$ holds in any model of $\Gamma \wedge \Pi$. Let us suppose, for contradiction, that $(\forall x \exists y : R_i^P(x,y)) \wedge (|R_i^P| = k)$ holds but there is some $a \in \Delta$ such that $R_i^P(a,b)$ and $R_i^P(a,b')$ are true for some $b \neq b' \in \Delta$. We have $|\{(x,y) \in \Delta^2 \mid R_i^P(x,y) \wedge x \neq a\}| \geq |\Delta| - 1$ by the fact $\forall x \exists y : R_i^P(x,y)$. It follows that $|R_i^P| \geq |\{(x,y) \in \Delta^2 \mid R_i^P(x,y) \wedge x \neq a\}| + 2 > |\Delta|$, which leads to a contradiction. Since all of R_i^P's are function predicates, it is easy to check $\forall x \exists_{=k} y : P(x,y)$ must be true in any model μ of $\Gamma \wedge \Pi$, i.e., $f(\mu) \models \Gamma \wedge \forall x \exists_{=k} y : P(x,y)$.

To finish the proof, one can easily show that, for every model $\mu \in \mathcal{M}_{\Gamma\wedge\forall x\exists_{=k}y:P(x,y),\Delta}$, there are exactly $(k!)^{|\Delta|}$ models $\mu' \in \mathcal{M}_{\Gamma\wedge\Pi,\Delta}$ such that $f(\mu') = \mu$. The reason for this is that 1) if, for any $a \in \Delta$, we permute $b_1, b_2, \ldots, b_k$ in $R_1^P(a,b_1), R_2^P(a,b_2), \ldots, R_k^P(a,b_k)$ in the model μ', we get another model of $\Gamma \wedge \Pi$, and 2) up to these permutations, the predicates R_i^P's in μ' are determined uniquely by μ. Finally, the weights of all these μ's are the same as those of μ, and we can write

$$\begin{aligned}
\sum_{\substack{\mu' \in \mathcal{M}_{\Gamma\wedge\Pi,\Delta}:\\ f(\mu')=\mu}} \mathbb{P}[\mu' \mid \Gamma \wedge \Pi] &= \frac{\sum_{\substack{\mu' \in \mathcal{M}_{\Gamma\wedge\Pi,\Delta}:\\ f(\mu')=\mu}} \mu'}{\mathsf{WFOMC}(\Gamma \wedge \Pi, \Delta, w, \bar{w})} \\
&= \frac{(k!)^{|\Delta|} \cdot \langle w, \bar{w} \rangle(\mu)}{(k!)^{|\Delta|} \cdot \mathsf{WFOMC}(\Gamma \wedge \forall x \exists_{=k} y : P(x,y), \Delta, w, \bar{w})} \\
&= \frac{\langle w, \bar{w} \rangle(\mu)}{\mathsf{WFOMC}(\Gamma \wedge \forall x \exists_{=k} y : P(x,y), \Delta, w, \bar{w})} \\
&= \mathbb{P}[\mu \mid \Gamma \wedge \forall x \exists_{=k} y : P(x,y)],
\end{aligned}$$

which completes the proof. □

Lemma 8. *Let Γ be a first-order logic sentence, Δ be a domain, and P be a predicate. Then the WFOMS $(\Gamma \wedge \forall_{=k} \forall y : P(x,y), \Delta, w, \bar{w})$ can be reduced to $(\Gamma \wedge (|U| = k) \wedge (\forall x : U(x) \Leftrightarrow (\forall y : P(x,y))), \Delta, w, \bar{w})$, where U is an auxiliary unary predicate with weight $w(U) = \bar{w}(U) = 1$, and the reduction is sound.*

Proof. The proof is straightforward. □

Proof of Lemma 6. We can first get rid of all formulas of the form $\exists_{=k} x \forall y : P(x,y)$ by repeatedly using Lemma 8. Then we can use Lemma 7 repeatedly to eliminate the formulas of the form $\forall x \exists_{=k} y : P(x,y)$. The whole reduction is sound due to the transitivity of soundness. □

A.3 Applying Domain Recursion Scheme on $\mathbf{C}^2$ is Possible

The $\mathbf{C}^2$ sentences that we need to handle are of the form

$$\Gamma \wedge \bigwedge_{k\in[q]} (\forall x : A_k(x) \Leftrightarrow (\exists_{=m_k} y : R_k(x,y))), \tag{18}$$

where Γ is a $\mathbf{FO}^2$ sentence, each $R_k(x,y)$ is an atomic formula, and each A_k is an auxiliary Tseitin predicate. Any WFOMS problem on $\mathbf{C}^2$ can be reduced to a new one, whose input sentence is of the above form and $\max_{k\in[q]} m_k \le |\Delta|$, by the following steps:

- Convert each counting-quantified formula of the form $\exists_{\ge m} y : \varphi(x,y)$ to $\neg(\exists_{\le m-1} y : \varphi(x,y))$.
- Decompose each $\exists_{\le m} y : \varphi(x,y)$ into $(\forall y : \neg\varphi(x,y)) \land \bigvee_{i\in[m]}(\exists_{=i} y : \varphi(x,y))$.
- Replace each subformula $\exists_{=m} y : \varphi(x,y)$, where $m > |\Delta|$, with False.
- Starting from the atomic level and working upwards, replace any subformula $\exists_{=m}\varphi(x,y)$, where $\varphi(x,y)$ is a formula that does not contain any counting quantifier, with $A(x)$; and append $\forall x \forall y : R(x,y) \Leftrightarrow \varphi(x,y)$ and $\forall x : A(x) \Leftrightarrow (\exists_{=m} y : R(x,y))$, where R is an auxiliary binary predicate, to the original sentence.

It is easy to check that the reduction presented above is sound and independent of the domain size if the domain size is greater than the maximum counting parameter m in the input sentence.[6]

We first sample the 1-types of each element from the sentence (18) so that all the predicates A_k's will be eliminated. The resulting WFOMS is then defined on the following sentence:

$$\Gamma_0 \land \bigwedge_{k\in[q]} \left(\bigwedge_{e\in\Delta_k^{\exists}} \exists_{=m_k} y : R_k(e,y) \land \bigwedge_{e\in\Delta_k^{\nexists}} \neg(\exists_{=k} y : R_k(e,y)) \right),$$

where Γ_0 is the simplified sentence of Γ by replacing all its unary literals with their truth values, $\Delta_k^{\exists}$ contains precisely the elements with positive sampled literals $A_k(e)$'s, and $\Delta_k^{\nexists} = \Delta \setminus \Delta_k^{\exists}$.

We need to consider a more general WFOMS problem to apply our domain recursion scheme. For each counting quantified formula $\exists_{=k} y : R_k(x,y)$, we introduce $2m_k$ new unary predicates $Z_{k,1}^{\exists}, Z_{k,2}^{\exists}, \dots, Z_{k,m_k}^{\exists}, Z_{k,1}^{\nexists}, Z_{k,2}^{\nexists}, \dots, Z_{k,m_k}^{\nexists}$, and append the conjunction of

$$\forall x : \left(Z_{k,t}^{\exists}(x) \Leftrightarrow (\exists_{=t} y : R_k(x,y)) \right) \land \left(Z_{k,t}^{\nexists}(x) \Leftrightarrow \neg(\exists_{=t} y : R_k(x,y)) \right)$$

over $t \in [m_k]$ to Γ_0, resulting in a new sentence Γ_1. The more general WFOMS is then defined on

$$\Gamma_1 \land \bigwedge_{i\in[n]} \nu_i(e_i), \tag{19}$$

where each $\nu_i(x)$ is a quantifier-free conjunction over a subset of $\{Z_{k,t}^{\exists}(x)\}_{t\in[m_k]} \cup \{Z_{k,t}^{\nexists}(x)\}_{t\in[m_k]}$. It is easy to check that the original WFOMS of (18) is reducible to the more general WFOMS problem, and the reduction is sound and independent of the domain size.

We show that the domain recursion scheme is still applicable to the WFOMS of (19). We only provide an intuition here while leaving the details for the future version of this paper. Additionally, we hope to discover a more practical and efficient solution in the future that would introduce fewer unary predicates, despite the current approach being domain-lifted.

The intuition is that we can view $\nu_i(x)$ as the "block type" similar to what we have done in the WMS of $\mathbf{FO}^2$. Then the domain recursion strategy is applied, first sampling the substructure

[6] This condition does not change the data complexity of the problem, as all the counting parameters in the sentence are considered constants but not the input of the problem.

of an element, and then updating each block type accordingly. The updated block types can still be represented by the unary predicates $Z^{\exists}_{k,t}$'s and $Z^{\nexists}_{k,t}$'s, and the new sampling problem is reducible to a new WFOMS of the general form. Following the similar argument for sampling $\mathbf{FO}^2$, the corresponding WMS for $\mathbf{C}^2$ is also lifted, which means that the full fragment of $\mathbf{C}^2$ is liftable under sampling.

A.4 Missing Details of Experiments

A.4.1 Experiment Settings

Sampling Combinatorial Structures The corresponding WFOMS problems for the uniform generation of combinatorial structures used in our experiments are presented as follows. The weighting functions w and $\bar{w}$ map all predicates to 1.

- Functions:
$$\forall x \exists_{=1} y : f(x, y).$$
- Functions w/o fix points:
$$(\forall x \exists_{=1} y : f(x, y)) \land (\forall x : \neg f(x, x)).$$
- Permutations:
$$(\forall x \exists_{=1} y : Per(x, y)) \land (\forall y \exists_{=1} x : Per(x, y)).$$
- Permutation without fix-points:
$$(\forall x \exists_{=1} y : Per(x, y)) \land (\forall y \exists_{=1} x : Per(x, y)) \land (\forall x : \neg Per(x, x)).$$

Sampling from MLNs An MLN is a finite set of weighted first-order formulas $\{(w_i, \alpha_i)\}_{i \in [m]}$, where each w_i is either a real-valued weight or ∞, and α_i is a first-order formula. Let $\mathcal{P}$ be the vocabulary of $\alpha_1, \alpha_2, \ldots, \alpha_m$. An MLN Φ paired with a domain Δ induces a probability distribution over $\mathcal{P}$-structures (also called possible worlds):

$$p_{\Phi,\Delta}(\omega) := \begin{cases} \frac{1}{Z_{\Phi,\Delta}} \exp\left(\sum_{(\alpha,w)\in\Phi_{\mathbb{R}}} w \cdot \#(\alpha, \omega)\right) & \text{if } \omega \models \Phi_\infty \\ 0 & \text{otherwise} \end{cases}$$

where $\Phi_{\mathbb{R}}$ and Φ_∞ are the real-valued and ∞-valued formulas in Φ respectively, and $\#(\alpha, \omega)$ is the number of groundings of α satisfied in ω. The sampling problem on an MLN Φ over a domain Δ is to randomly generate a possible world ω according to the probability $p_{\Phi,\Delta}(\omega)$.

The reduction from the sampling problems on MLNs to WFOMS can be performed as follows. For every real-valued formula $(\alpha_i, w_i) \in \Phi_{\mathbb{R}}$, where the free variables in α_i are $\mathbf{x}$, we introduce a novel auxiliary predicate ξ_i and create a new formula $\forall \mathbf{x} : \xi_i(\mathbf{x}) \Leftrightarrow \alpha_i(\mathbf{x})$. For formula α_i with infinity weight, we instead create a new formula $\forall \mathbf{x} : \alpha_i(\mathbf{x})$. Denote the conjunction of the resulting set of sentences by Γ, and set the weighting function to be $w(\xi_i) = \exp(w_i)$ and $\bar{w}(\xi_i) = 1$, and for all other predicates, we set both w and $\bar{w}$ to be 1. Then the sampling problem on Φ over Δ is reduced to the WFOMS $(\Gamma, \Delta, w, \bar{w})$.

By the reduction above, we can write the two MLNs used in our experiments to WFOMS problems. The weights of predicates are all set to be 1 unless otherwise specified.

- Friends-smokers MLN: the reduced sentence is

$$\begin{aligned}&(\forall x : \neg fr(x,x) \wedge sm(x))\\ &\wedge (\forall x \forall y : fr(x,y) \Leftrightarrow fr(y,x))\\ &\wedge (\forall x \forall y : \xi(x,y) \Leftrightarrow (fr(x,y) \wedge sm(x) \Rightarrow sm(y)))\\ &\wedge (\forall x \exists y : fr(x,y)),\end{aligned}$$

  and the weight of ξ is set to be $w(\xi) = \exp(0.2)$.

- Employment MLN: the corresponding sentence is

$$\forall x : \xi(x) \Leftrightarrow (\exists y : workfor(x,y) \vee boss(x)),$$

  and the weight of ξ is set to be $\exp(1.3)$.

A.4.2 More Experimental Results

The Kolmogorov–Smirnov Test We utilized the Kolmogorov-Smirnov (KS) test [29] to validate the conformity of the (count) distributions produced by our algorithm to the reference distributions. The KS test used here is based on the multivariate Dvoretzky–Kiefer–Wolfowitz (DKW) inequality recently proved by [30].

Let $\mathbf{X}_1 = (X_{1i})_{i\in[k]}, \mathbf{X}_2 = (X_{2i})_{i\in[k]}, \ldots, \mathbf{X}_n = (X_{ni})_{i\in[k]}$ be n real-valued independent and identical distributed multivariate random variables with cumulative distribution function (CDF) $F(\cdot)$. Let $F_n(\cdot)$ be the associated empirical distribution function defined by

$$F_n(\mathbf{x}) := \frac{1}{n}\sum_{i\in[n]} \mathbb{1}_{X_{i1}\le x_1, X_{i2}\le x_2, \ldots, X_{ik}\le x_k}, \qquad \mathbf{x} \in \mathbb{R}^k.$$

The DKW inequality states

$$\mathbb{P}\left[\sup_{\mathbf{x}\in\mathbb{R}^k} |F_n(\mathbf{x}) - F(\mathbf{x})| > \epsilon\right] \ge (n+1)ke^{-2n\epsilon^2} \tag{20}$$

for every $\epsilon, n, k > 0$. When the random variables are univariate, i.e., $k = 1$, we can replace $(n+1)k$ in the above probability bound by a tighter constant 2.

Table 1: The Kolmogorov-Smirnov Test

Problem	Maximum deviation	Upper bound
graphs w/o isolated vertices	0.0036	0.0049
2-regular graphs	0.0065	0.0069
functions	0.0013	0.0024
functions w/o fix-points	0.0027	0.0042
permutations	0.0071	0.0124
permutations w/o fix-points	0.019	0.02
friends-smokers	0.0021	0.0087
employmenet	0.0030	0.0087

In the KS test, the null hypothesize is that the samples $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n$ are distributed according to some reference distribution, whose CDF is $F(\cdot)$. Then by (20), with probability $1-\alpha$, the maximum deviation $\sup_{\mathbf{x}\in\mathbb{R}^k} |F_n(\mathbf{x}) - F(\mathbf{x})|$ between empirical and reference distributions is bounded by $\epsilon = \sqrt{\ln(k(n+1)/\alpha)/2n}$ ($\sqrt{\ln(2/\alpha)/2n}$ for the univariate case). If the actual value of the maximum deviation is larger than ϵ, we can reject the null hypothesis at the confidence level α. Otherwise, we cannot reject the null hypothesis, i.e., the empirical distribution of the samples is not statistically different from the reference one. In our experiments, we choose $\alpha = 0.05$ as a significant level.

For the uniform generation of combinatorial structures, we assigned each model a lexicographical number and treated the model index as a random variable with a discrete uniform distribution. For the sampling problems of MLNs, we test their count distributions against the true count distributions. Table 1 shows the maximum deviation between the empirical and reference cumulative distribution functions, along with the upper bound set by the DKW inequality. As shown in Table 1, all maximum deviations are within their respective upper bounds. Therefore, we cannot reject any null hypotheses, i.e., there is no statistically significant difference between the two sets of distributions.

A.5 Missing Details of WMS

A.5.1 Optimizations for WMS

There exist several optimizations to make it more practical. Here, we present some of them that are used in our implementation.

- The complexity of DRSampler heavily depends on the recursion depth. In our implementation, when selecting a domain element e_t for sampling its substructure, we always chose the element with the "strongest" existential constraint that contains the most Tseitin atoms $Z_k(x)$'s. It would help DRSampler fast reach the condition that the existential constraint for all elements is $\top$. In this case, DRSampler will invoke the more efficient WMS for $\mathbf{UFO}^2$ to sample the remaining substructures.

- Let $\mathcal{P}_\exists$ be the union of vocabularies of the existentially quantified formulas

$$\mathcal{P}_\exists := \mathcal{P}_{\varphi_1(x,y)} \cup \mathcal{P}_{\varphi_2(x,y)} \cup \cdots \cup \mathcal{P}_{\varphi_m(x,y)}.$$

  We further decomposed the sampling probability $\mathbb{P}[\mathcal{A} \mid \Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i)]$ into

$$\mathbb{P}\left[\mathcal{A} \mid \Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i) \wedge \mathcal{A}_\exists\right] \cdot \mathbb{P}\left[\mathcal{A}_\exists \mid \Gamma_T \wedge \bigwedge_{i\in[n]} \eta_i(e_i)\right],$$

  where $\mathcal{A}_\exists$ is a $\mathcal{P}_\exists$-structure over Δ. It decomposes the conditional sampling problem of $\mathcal{A}$ into two subproblems—one is to sample $\mathcal{A}_\exists$ and the other sample the remaining substructures conditional on $\mathcal{A}_\exists$. The advantage of this decomposition is that the latter subproblem can be reduced into a sampling problem on a $\mathbf{UFO}^2$ sentence, since all existentially-quantified formulas have been satisfied with $\mathcal{A}_\exists$. The first subproblem for sampling $\mathcal{A}_\exists$ can be solved by a similar algorithm of DRSampler. In this algorithm, the 2-tables used to partition cells are now defined over $\mathcal{P}_\exists$, whose size is exponentially smaller than the one in the original algorithm. As a result, the enumeration of 2-table configurations in Algorithm 2 will be exponentially faster.

- We cached $\mathcal{W}_{\mathbf{n}}$, the weight $\langle w, \bar{w} \rangle(\tau^i)$ of all 1-types and the weight $\langle w, \bar{w} \rangle(\pi^i)$ of all 2-tables, which are widely used in our WMS.

A.5.2 The Function **ExSat**$(\cdot, \cdot)$

The pseudo-code of ExSat is presented in Algorithm 4.

Algorithm 4 ExSat$(\mathbf{g}, \eta)$

```
Decompose $\mathbf{g}$ into $\{g_{\eta^i,\pi^j}\}_{i\in[N_c],j\in[N_b]}$
$(\beta, \tau) \leftarrow (\eta)$
// Check the coherence of 2-tables
for $i \in [N_u]$ do
    for $j \in [N_b]$ do
        // $\tau(\eta^i)$ is the 1-type in $\eta^i$
        if $\pi^j$ is not coherent with $\tau(\eta^i)$ and $\tau$ and $g_{\eta^i,\pi^j} > 0$ then
            return False
        end if
    end for
end for
// Check the satisfaction of existentially-quantified formulas
$\forall j \in [N_b], h_{\pi^j} \leftarrow \sum_{i\in[N_c]} g_{\eta^i,\pi^j}$
for $Z_k(x) \in \beta$ do
    for $j \in [N_b]$ do
        if $R_k(x, y) \in \pi^j$ and $h_{\pi^j} > 0$ then
            return True
        end if
    end for
end for
return False
```

A.5.3 A Lifted WMS for $\mathbf{FO}^2$ with Cardinality Constraints

We present the modified sampling algorithm for $\mathbf{FO}^2$ with cardinality constraints. The changes from the original WMS for $\mathbf{FO}^2$ are highlighted by the blue lines.

Algorithm 5 $\mathsf{WMS}(\Gamma, \Upsilon, \Delta, w, \bar{w})$

INPUT: An $\mathbf{FO}^2$ sentence Γ of the form (3), a set of cardinality constraints Υ of the form (16), a domain $\Delta = \{e_i\}_{i\in[n]}$ of size n, a weighting $(w, \bar{w})$
OUTPUT: A model μ of $\Gamma \wedge \Upsilon$ over Δ

```
Construct $\Gamma$ to $\Gamma_T$ by (4)
$\forall i \in [n], \beta_i(x) \leftarrow \bigwedge_{k\in[m]} Z_k(x)$
$\widehat{\Gamma} \leftarrow \Gamma_T \wedge \bigwedge_{i\in[n]} \beta_i(e_i)$
$(\tau_i)_{i\in[n]} \leftarrow \mathsf{OneTypeSampler}(\widehat{\Gamma}, \Upsilon, \Delta, w, \bar{w})$
$\forall i \in [n], \eta_i \leftarrow (\beta_i, \tau_i)$
$\mu \leftarrow \mathsf{DRSampler}(\Gamma_T, \Upsilon, \Delta, w, \bar{w}, (\eta_i)_{i\in[n]})$
return $\mu$
```

Algorithm 6 $\text{OneTypeSampler}(\widehat{\Gamma}, \Upsilon, \Delta, w, \bar{w})$

```
$W \leftarrow \mathsf{WFOMC}(\widehat{\Gamma} \wedge \Upsilon, \Delta, w, \bar{w})$
Obtain the blocks $B_{\beta^1}, B_{\beta^2}, \ldots, B_{\beta^{2^m}}$ from $\widehat{\Gamma}$
for $\left(\mathbf{n}_{\beta^i}\right)_{i\in[2^m]} \in \mathsf{Prod}\left(\mathcal{T}_{|B_{\beta^1}|,N_u}, \ldots, \mathcal{T}_{|B_{\beta^{2^m}}|,N_u}\right)$ do
    $\mathbf{n} \leftarrow \bigoplus_{i\in[2^m]} \mathbf{n}_{\beta^i}$
    Compute $\mathcal{W}_{\mathbf{n},\Upsilon}$
    $W' \leftarrow \mathcal{W}_{\mathbf{n},\Upsilon} \cdot \prod_{t=1}^{2^m} \binom{|B_{\beta^t}|}{\mathbf{n}_{\beta^t}}$
    if $\mathsf{Uniform}(0, 1) < \frac{W'}{W}$ then
        $\mathbf{n}^* \leftarrow \mathbf{n}$
        break
    else
        $W \leftarrow W - W'$
    end if
end for
for $i \in [2^m]$ do
    Fetch the cell configuration $\mathbf{n}^*_{\beta^i}$ in $\beta^i$ from $\mathbf{n}^*$
    Randomly partition $B_{\beta^i}$ into $\{C_{\beta^i,\tau^j}\}_{j\in[N_u]}$ according to $\mathbf{n}^*_{\beta^i}$
    for $j \in [N_u]$ do
        Assign the 1-type $\tau^j$ to all elements in $C_{\beta^i,\tau^j}$
    end for
end for
```

Algorithm 7 $\mathsf{DRSampler}(\Gamma_T, \Upsilon, \Delta, w, \bar{w}, (\eta_i)_{i\in[n]})$

```
if $n = 1$ then
    return $\emptyset$
end if
if only the block of type $\beta = \top$ is nonempty then
    return a model $\mu$ sampled by a $\mathbf{UFO}^2$ WMS from $\forall x \forall y : \psi(x, y) \wedge \bigwedge_{i\in[n]} \tau_i(e_i) \wedge \Upsilon$ over $\Delta$ under $(w, \bar{w})$
end if
Choose $t \in [n]$; $\mu \leftarrow \tau_t(e_t)$; $\Delta' \leftarrow \Delta \setminus \{e_t\}$
Get the cell configuration $\mathbf{n} = \left(n_{\eta^i}\right)_{i\in[N_c]}$ of $(\eta_i)_{i\in[n]\setminus\{t\}}$
$W \leftarrow \mathcal{W}_{\mathbf{n},\Upsilon}$
$\forall i \in [N_c], T_i \leftarrow \mathcal{T}_{n_{\eta^i}, N_b}$
for $\left(\mathbf{g}_{\eta^i}\right)_{i\in[N_c]} \leftarrow \mathsf{Prod}(\mathcal{T}_{n_{\eta^1}, N_b}, \ldots, \mathcal{T}_{n_{\eta^{N_c}}, N_b})$ do
    $\mathbf{g} \leftarrow \bigoplus_{i\in[N_c]} \mathbf{g}_{\eta^i}$
    if $\mathsf{ExSat}\,(\mathbf{g}, \eta_t, \Upsilon)$ then
        Get the new cell configuration $\mathbf{n}'$ w.r.t. $\mathbf{g}$ by (15)
        Get the new cardinality constraints $\Upsilon'$ w.r.t. $\mathbf{g}$ by (17)
        Compute $\mathcal{W}_{\mathbf{n}',\Upsilon}$
        $W' \leftarrow \mathcal{W}_{\mathbf{n}',\Upsilon} \cdot \langle w, \bar{w}\rangle(\tau_t) \cdot \prod_{i\in[N_c]} \binom{n_{\eta^i}}{\mathbf{g}_{\eta^i}} \cdot \mathbf{w}^{\mathbf{g}_{\eta^i}}$
        if $\mathsf{Uniform}(0, 1) < \frac{W'}{W}$ then
            $\mathbf{g}^* \leftarrow \mathbf{g}$
            break
        else
            $W \leftarrow W - W'$
        end if
    end if
end for
Obtain the cell partition $\{C_{\eta^i}\}_{i\in[N_c]}$ from $(\eta_i)_{i\in[n]\setminus\{t\}}$
for $i \in [N_c]$ do
    Fetch the cardinality vector $\mathbf{g}^*_{\eta^i}$ of $\eta^i$ from $\mathbf{g}^*$
    Randomly partition the cell $C_{\eta^i}$ into $\left\{G_{\eta^i,\pi^j}\right\}_{j\in[N_b]}$ according to $\mathbf{g}^*_{\eta^i}$
    for $j \in [N_b]$ do
        $\mu \leftarrow \mu \cup \left\{\pi^j(e_t, e)\right\}_{e\in G_{\eta^i,\pi^j}}$
        $\forall e_s \in G_{\eta^i,\pi^j}, \eta'_s \leftarrow \eta_{s|\pi^j}$
    end for
end for
Obtain the reduced cardinality constraints $\Upsilon'$ w.r.t. $\mathbf{g}^*$ by (17)
$\mu \leftarrow \mu \cup \mathsf{DRSampler}(\Gamma_T, \Upsilon', \Delta', w, \bar{w}, (\eta'_i)_{i\in[n-1]})$
return $\mu$
```

References

[1] Y. Wang, T. Van Bremen, Y. Wang, and O. Kuželka, "Domain-lifted sampling for universal two-variable logic and extensions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 9, 2022, pp. 10 070–10 079.

[2] L. D. Raedt, K. Kersting, S. Natarajan, and D. Poole, "Statistical relational artificial intelligence: Logic, probability, and computation," *Synthesis lectures on artificial intelligence and machine learning*, vol. 10, no. 2, pp. 1–189, 2016.

[3] G. Van den Broeck, N. Taghipour, W. Meert, J. Davis, and L. De Raedt, "Lifted probabilistic inference by first-order knowledge compilation," in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence*, 2011, pp. 2178–2185.

[4] G. Van den Broeck, W. Meert, and A. Darwiche, "Skolemization for weighted first-order model counting," in *Fourteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2014.

[5] A. J. Robinson and A. Voronkov, *Handbook of automated reasoning.* Elsevier, 2001, vol. 1.

[6] A. Kuusisto and C. Lutz, "Weighted model counting beyond two-variable logic," in *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, 2018, pp. 619–628.

[7] O. Kuzelka, "Weighted first-order model counting in the two-variable fragment with counting quantifiers," *Journal of Artificial Intelligence Research*, vol. 70, pp. 1281–1307, 2021.

[8] C. Cooper, M. Dyer, and C. Greenhill, "Sampling regular graphs and a peer-to-peer network," *Combinatorics, Probability and Computing*, vol. 16, no. 4, pp. 557–593, 2007.

[9] P. Gao and N. Wormald, "Uniform generation of random regular graphs," in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science.* IEEE, 2015, pp. 1218–1230.

[10] D. Poole, "First-order probabilistic inference," in *IJCAI*, vol. 3, 2003, pp. 985–991.

[11] M. Richardson and P. Domingos, "Markov logic networks," *Machine learning*, vol. 62, no. 1, pp. 107–136, 2006.

[12] P. Beame, G. Van den Broeck, E. Gribkoff, and D. Suciu, "Symmetric weighted first-order model counting," in *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, 2015, pp. 313–328.

[13] G. Van den Broeck, "On the completeness of first-order knowledge compilation for lifted probabilistic inference," *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[14] S. M. Kazemi, A. Kimmig, G. Van den Broeck, and D. Poole, "New liftable classes for first-order probabilistic inference," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[15] S. M. Kazemi, A. Kimmig, G. V. d. Broeck, and D. Poole, "Domain recursion for lifted inference with existential quantifiers," *arXiv preprint arXiv:1707.07763*, 2017.

[16] J. Tóth and O. Kuželka, "Lifted inference with linear order axiom," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023, to appear.

[17] C. P. Gomes, A. Sabharwal, and B. Selman, "Model counting," in *Handbook of satisfiability.* IOS press, 2021, pp. 993–1014.

[18] S. Chakraborty, K. S. Meel, and M. Y. Vardi, "A scalable and nearly uniform generator of sat witnesses," in *International Conference on Computer Aided Verification.* Springer, 2013, pp. 608–623.

[19] S. Chakraborty, D. J. Fremont, K. S. Meel, S. A. Seshia, and M. Y. Vardi, "On parallel scalable uniform sat witness generation," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems.* Springer, 2015, pp. 304–319.

[20] G. Van den Broeck and J. Davis, "Conditioning in first-order knowledge compilation and lifted probabilistic inference," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[21] G. Van den Broeck and A. Darwiche, "On the complexity and approximation of binary evidence in lifted inference," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[22] D. Scott, "A decision method for validity of sentences in two variables," *Journal of Symbolic Logic*, vol. 27, no. 377, p. 74, 1962.

[23] E. Gradel, M. Otto, and E. Rosen, "Two-variable logic with counting is decidable," in *Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science.* IEEE, 1997, pp. 306–317.

[24] E. Grädel, P. G. Kolaitis, and M. Y. Vardi, "On the decision problem for two-variable first-order logic," *Bulletin of symbolic logic*, vol. 3, no. 1, pp. 53–69, 1997.

[25] M. Soos, S. Gocht, and K. S. Meel, "Tinted, detached, and lazy cnf-xor solving and its applications to counting and sampling," in *International Conference on Computer Aided Verification.* Springer, 2020, pp. 463–484.

[26] S. Chakraborty, D. Fried, K. S. Meel, and M. Y. Vardi, "From weighted to unweighted model counting," in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[27] T. Van Bremen and O. Kuželka, "Lifted inference with tree axioms," in *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, vol. 18, no. 1, 2021, pp. 599–608.

[28] M. Jaeger, "Lower complexity bounds for lifted inference," *Theory and Practice of Logic Programming*, vol. 15, no. 2, pp. 246–263, 2015.

[29] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[30] M. Naaman, "On the tight constant in the multivariate dvoretzky–kiefer–wolfowitz inequality," *Statistics & Probability Letters*, vol. 173, p. 109088, 2021.