# Domain-Lifted Sampling for Universal Two-Variable Logic and Extensions

**Yuanhong Wang**[1], **Timothy van Bremen**[2], **Yuyi Wang**[3, 4*], **Ondřej Kuželka**[5]
[1] Beihang University, China
[2] KU Leuven, Belgium
[3] CRRC Zhuzhou Institute, China
[4] ETH Zurich, Switzerland
[5] Czech Technical University in Prague, Czech Republic
lucienwang@buaa.edu.cn, timothy.vanbremen@cs.kuleuven.be, yuyiwang920@gmail.com, ondrej.kuzelka@fel.cvut.cz

## Abstract

Given a first-order sentence $\Gamma$ and a domain size $n$, how can one sample a model of $\Gamma$ on the domain $\{1, \ldots, n\}$ efficiently as $n$ scales? We consider two variants of this problem: the *uniform* sampling regime, in which the goal is to sample a model uniformly at random, and the *symmetric weighted* sampling regime, in which models are weighted according to the number of groundings of each predicate appearing in them. Solutions to this problem have applications to the scalable generation of combinatorial structures, as well as sampling in several statistical-relational models such as Markov logic networks and probabilistic logic programs. In this paper, we identify certain classes of sentences that are *domain-liftable under sampling*, in the sense that they admit a sampling algorithm that runs in time polynomial in $n$. In particular, we prove that every sentence of the form $\forall x \forall y : \psi(x, y)$ for some quantifier-free formula $\psi(x, y)$ is domain-liftable under sampling. We then further show that this result continues to hold in the presence of one or more *cardinality constraints* as well as a single *tree axiom* constraint.

## Introduction

We introduce and study the *first-order sampling* problem: given a sentence in first-order logic $\Gamma$ and a domain size $n$, sample a model of $\Gamma$ on the domain $\{1, \ldots, n\}$. Clearly, naive approaches such as rejection sampling are intractable for all but the smallest domains here, so we focus our attention on designing *domain-liftable* sampling algorithms, meaning their runtime is polynomial in $n$. Our approach is inspired by the *first-order model counting* literature (Van den Broeck et al. 2011; Beame et al. 2015), in which the goal is to *count* the number of models of a sentence over a fixed domain size $n$ in time polynomial in $n$; however, as we shall see, new challenges arise when translating results there to the sampling setting.

The first-order sampling problem has applications in several areas of computer science. Many problems related to the generation of combinatorial structures can easily be expressed as first-order sampling problems. Indeed, a great deal of research in the combinatorics community has gone into designing sampling schemes for particular classes of structures (Jerrum, Valiant, and Vazirani 1986); by viewing these problems through the lens of first-order sampling, one can obtain a more generalizable approach. For example, suppose we are interested in uniformly sampling labelled 2-colored trees with $n$ nodes, where one color is used $k$ times and the other $n - k$ times. This is precisely the type of problems that the results in our paper allow us to solve; it can be modelled with the sentence:

$$
\begin{aligned}
&\forall x : \neg R(x, x) \wedge \\
&\forall x \forall y : R(x, y) \rightarrow R(y, x) \wedge \\
&\forall x : C_1(x) \vee C_2(x) \wedge \\
&\forall x : \neg C_1(x) \vee \neg C_2(x) \wedge \\
&\forall x \forall y : R(x, y) \rightarrow \\
&\qquad \neg(C_1(x) \wedge C_1(y)) \wedge \neg(C_2(x) \wedge C_2(y)) \wedge \\
&\qquad \mathsf{T}_R \wedge \\
&\qquad |C_1| = k \wedge |C_2| = n - k
\end{aligned}
$$

where $\mathsf{T}_R$ enforces that the binary relation $R$ forms a tree, and $|C_1| = k$ and $|C_2| = n - k$ enforces cardinality constraints on the number of groundings of the relations $C_1$ and $C_2$.

There are also applications of first-order sampling in the domain of *statistical-relational learning* (De Raedt et al. 2016). For instance, Markov logic networks (MLNs) (Richardson and Domingos 2006) are a flexible formalism for mixing uncertainty with first-order logic. Probabilistic inference in an MLN is known to be reducible to weighted first-order model counting (Van den Broeck et al. 2011); as it turns out, essentially the same reduction can be used to sample possible worlds from an MLN. Domain-liftability under sampling is particularly important in these applications, as MLNs are often modelled over large populations.

In this paper, we prove the domain-liftability under sampling of $\mathbf{UFO}^2$, the universally-quantified two-variable fragment of first-order logic, comprising sentences of the form $\forall x \forall y : \psi(x, y)$ for some quantifier-free formula $\psi(x, y)$. We then show how to further extend this result in two different directions: one allowing the addition of *cardinality constraints* (Kuželka 2021), and another allowing the addition of a *tree constraint* (van Bremen and Kuželka 2021).

*partially supported by Huawei TCS Lab

## Related Work

The approach taken in this paper, as well as the formal *liftability* notions considered here, are inspired by the *lifted inference* literature (Van den Broeck et al. 2021). In lifted inference, the goal is to perform probabilistic inference in statistical-relational models in a way that exploits symmetries in the high-level structure of the model. Models that are amenable to scalable inference as domain size increases are dubbed *domain-liftable*, in a similar spirit to the notion of *domain-liftability under sampling* presented here. There exists a very extensive literature on lifted inference, both viewed from the logical as well as graphical models perspective (Poole 2003; de Salvo Braz, Amir, and Roth 2005; Gogate and Domingos 2011; Van den Broeck et al. 2011; Taghipour et al. 2013), which we do not cover exhaustively here for brevity. We do, however, draw the reader's attention to Beame et al. (2015, Appendix C), which studies the data complexity of weighted first-order model counting (WFOMC) of the two-variable fragment of first-order logic. The general argument used there—namely, the analysis of a two-variable sentence in terms of its cell types—forms a basis for our sampling approach discussed later in the paper.

On a related note, sampling from *propositional* logic formulas is a relatively well-studied area (Gomes, Sabharwal, and Selman 2006; Chakraborty, Meel, and Vardi 2013; Chakraborty et al. 2014). However, many real-world problems can achieve a far more natural and compact representation when expressed in first-order logic, and suffer from a significant blow-up in formula size when grounded out to propositional logic. Due to the lack of immediately identifiable symmetries in propositional logic formulas, most sampling approaches for propositional logic instead seek to design *approximate* samplers, typically through the imposition of random parity constraints. It is important to note as well that these approaches are not polynomial-time in the input formula length, and rely on efficient access to an NP oracle (typically realised through a SAT solver). In this paper, we instead aim to exploit the structure inherent in first-order logic to obtain *exact* lifted sampling schemes, a setting that to the best of our knowledge remains largely unstudied.

Gogate, Jha, and Venugopal (2012) studied *lifted importance sampling*, building on earlier work by Gogate and Domingos (2011). However, the algorithms here are approximate with a view to estimating the partition function of an MLN, unlike the exact and more general first-order logic setting considered in this paper. Van den Broeck and Niepert (2015) also propose an approximate sampling-based inference framework for graphical models in a similar spirit.

Finally, Lin, Lu, and Tan (2021) recently proposed an efficient algorithm for satisfiability testing of the two-variable fragment of first-order logic, that exhibits a finite model for a given formula if one exists. However, no guarantees are made as to the model size (other than an exponential upper bound) or the distribution of the algorithm's output.

## Background

In this section, we describe all the necessary background for the paper.

### First-Order Logic

We consider the function-free, finite domain fragment of first-order logic defined by a set of constants $\Delta$, called a *domain*, a set of variables $\mathcal{V}$ and a set of predicates $\mathcal{P}$. For any $P \in \mathcal{P}$, denote by $arity(P)$ the arity of $P$. An *atom* takes the form $P(x_1, \ldots, x_k)$ where $x_1, \ldots, x_k \in \Delta \cup \mathcal{V}$, $P \in \mathcal{P}$ and $arity(P) = k$. 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*. A formula is *ground* if it contains no logical variables. The *groundings* of a quantifier-free formula is the set of formulas obtained by instantiating the free variables with any possible combination of constants from $\Delta$. A *possible world* $\omega$ is represented as the set of ground atoms that are true in $\omega$. A possible world $\omega$ is called a *model* of a sentence $\Gamma$ if it satisfies $\Gamma$ (i.e., $\omega \models \Gamma$), according to the usual semantics of first-order logic. We denote the set of all models of a sentence $\Gamma$ over a fixed domain $\Delta = [n] = \{1, \ldots, n\}$ by $\mathrm{models}_{\mathrm{n}}(\Gamma)$. The two-variable syntactic fragment of first-order logic ($\mathbf{FO}^2$) is obtained by restricting the set of variables allowed in a formula to $\mathcal{V} = \{x, y\}$. We denote the subfragment of $\mathbf{FO}^2$ comprising sentences of the form $\forall x \forall y : \psi(x, y)$, where $\psi(x, y)$ is some quantifier-free formula, by $\mathbf{UFO}^2$.

### Weighted First-Order Model Counting

The sampling problems investigated by this paper are inspired by the problem of weighted first-order model counting (WFOMC), which has been extensively studied in, among other works, (Van den Broeck et al. 2011; den Broeck, Meert, and Darwiche 2014; Beame et al. 2015; Kuusisto and Lutz 2018; van Bremen and Kuželka 2021).

**Definition 1** (Weighting)**.** *Denote the set of predicates appearing in the sentence $\Gamma$ by $\mathcal{P}_\Gamma$. A weighting on $\Gamma$ is a pair of mappings $w : \mathcal{P}_\Gamma \to \mathbb{R}_{\geqslant 0}$ and $\bar{w} : \mathcal{P}_\Gamma \to \mathbb{R}_{\geqslant 0}$*[1]*.*

Given a weighting $(w, \bar{w})$ on a sentence, for every possible world $\omega$, we define its weight as

$$W_{w,\bar{w}}(\omega) := \prod_{l \in \omega_T} w(\mathsf{pred}(l)) \cdot \prod_{l \in \omega_F} \bar{w}(\mathsf{pred}(l)),$$

where $\omega_T$ ($\omega_F$) denotes the set of true (false) ground atoms in the possible world $\omega$, and the notation $\mathsf{pred}(l)$ maps an atom $l$ to its corresponding predicate name. When the context is clear, we leave out $w$ and $\bar{w}$ in the subscript of $W_{w,\bar{w}}$.

**Definition 2** (WFOMC)**.** *Let $(w, \bar{w})$ be a weighting on a sentence $\Gamma$. The WFOMC of $\Gamma$ over a domain of size $n$ under $(w, \bar{w})$ is*

$$\mathsf{WFOMC}(\Gamma, n, w, \bar{w}) := \sum_{\mu \in \mathrm{models}_{\mathrm{n}}(\Gamma)} W_{w,\bar{w}}(\mu).$$

[1] We use a slightly different definition of weighting here than that used in the WFOMC literature, by requiring the range of the weighting function to be non-negative. This ensures that the probability of a model introduced later is well-defined.

Note that since these weightings are defined on the predicate level, all groundings of the same predicate get the same weights. For this reason, the notion of WFOMC defined here is also referred to as *symmetric* WFOMC.

Given a sentence, or a class of sentences, prior research has mainly focused on its *data complexity* for WFOMC: the complexity of computing $\mathsf{WFOMC}(\Gamma, n, w, \bar{w})$ when fixing the input sentence $\Gamma$ and weighting $(w, \bar{w})$, and treating the domain size $n$ as a unary input. A sentence, or class of sentences, that enjoys polynomial-time data complexity is said to be *domain-liftable*.

## Count Distribution

The distribution of predicate cardinalities among the models of a sentence, called the *count distribution*, plays an important role in our sampling algorithm. Given a possible world $\omega$ and a list of predicates $\Psi = (P_1, \ldots, P_m)$, we denote the vectors of *count-statistics*: $\mathbf{N}(\Psi, \omega) := (n_1, \ldots, n_m)$, where $\forall i \in [m], n_i = |P_i|$ is the cardinality of the predicate $P_i$, i.e., the number of ground atoms for $P_i$ that are true in $\omega$.

**Definition 3** (Count distribution)**.** *Given a sentence $\Gamma$, and a list of predicates $\Psi$, the count distribution of $\Psi$ conditioned on $\Gamma$ over a domain of size $n$ under $(w, \bar{w})$ is*

$$q_{\Gamma,\Psi}(\mathbf{n}) := \sum_{\substack{\mu \in \mathrm{models}_{\mathrm{n}}(\Gamma): \\ \mathbf{N}(\Psi,\mu)=\mathbf{n}}} \frac{W(\mu)}{\mathsf{WFOMC}(\Gamma, n, w, \bar{w})}. \quad (1)$$

We can efficiently compute a count distribution as long as there exists a domain-lifted algorithm for WFOMC on the sentence in question.

**Theorem 1** (Proposition 4 in Kuželka (2021))**.** *For any set of predicates $\Psi$, if there is a domain-lifted oracle for computing $\mathsf{WFOMC}(\Gamma, n, w, \bar{w})$, the count distribution $q_{\Gamma,\Psi}$ can be obtained by a polynomial number of queries to this oracle.*

## Graph Theory

We will need some tools from graph theory to handle the tree constraint. We deal with undirected weighted graphs without self-loops. We can characterise any such graph by its symmetric weighted adjacency matrix $A$, whose element at position $(a, b)$ denotes the weight on the edge $\{a, b\}$. Denote $\mathcal{V}(A)$ as the set of vertices in $A$, and $\mathcal{E}(A)$ the set of edges with non-zero weight. Let $\mathcal{S}(A)$ be the set of (weighted) spanning trees of $A$. Define

$$\mathsf{TS}(A, \mathcal{F}) := \sum_{\substack{T \in \mathcal{S}(A): \\ \mathcal{F} \subseteq T}} \prod_{\{a,b\} \in T} A_{(a,b)} \quad (2)$$

as the weighted sum of spanning trees of $A$ that contain all edges in $\mathcal{F} \subseteq \mathcal{E}(A)$. Here we give some intuitions about the constraint that certain edges must be present in the spanning trees: in our algorithm, we sample a spanning tree edge by edge, and at each step we need to compute the number of trees that must contain the already sampled edges. $\mathsf{TS}(A, \mathcal{F})$ can be efficiently computed in time polynomial in $|\mathcal{V}(A)|$ by contracting the edges that must be present, then applying Kirchhoff's theorem (Chaiken and Kleitman 1978). We refer the reader to van Bremen and Kuželka (2021) for the details.

# Problem Statement

We are now ready to formally define the problem of *uniform first-order sampling*, along with its weighted counterpart *symmetric weighted first-order sampling*.

**Definition 4** (Uniform first-order sampling)**.** *Given a first-order sentence $\Gamma$, the uniform first-order sampling problem on $\Gamma$ over a domain of size $n$ is to generate a random model $G(\Gamma, n)$ in $\mathrm{models}_{\mathrm{n}}(\Gamma)$ such that $\mathbb{P}[G(\Gamma, n) = \mu] = 1/|\mathrm{models}_{\mathrm{n}}(\Gamma)|$ for every $\mu \in \mathrm{models}_{\mathrm{n}}(\Gamma)$.*

In some cases—for example, when sampling from MLNs—we are more interested in a *weighted* model sampler of a sentence, that generates models with differing probabilities according to the atoms that appear in the model. We use the same weighting definition as in symmetric WFOMC, and define the problem of symmetric weighted first-order sampling.

**Definition 5** (Symmetric weighted first-order sampling)**.** *Let $(w, \bar{w})$ be a weighting on a sentence $\Gamma$. The symmetric weighted first-order sampling problem on $\Gamma$ over a domain of size $n$ under $(w, \bar{w})$ is to generate a model $G(\Gamma, n, w, \bar{w})$ of $\Gamma$ such that*

$$\mathbb{P}[G(\Gamma, n, w, \bar{w}) = \mu] = \frac{W_{w,\bar{w}}(\mu)}{\mathsf{WFOMC}(\Gamma, n, w, \bar{w})} \quad (3)$$

*for every $\mu \in \mathrm{models}_{\mathrm{n}}(\Gamma)$.*

We call a probabilistic algorithm that realises a solution to the uniform or symmetric weighted first-order sampling problem a *uniform model sampler* (UMS) or *weighted model sampler* (WMS) respectively. Several sampling problems in combinatorics can be directly solved given access to a UMS or WMS, as long as the characterizing properties of the structure in the question can be encoded as a first-order logic sentence.

**Example 1.** *A WMS of the sentence* [2] $\forall x \forall y : (E(x, y) \rightarrow E(y, x)) \wedge \neg E(x, x) \wedge \forall x \forall y : x \neq y \wedge cliq(x) \wedge cliq(y) \rightarrow E(x, y)$ *with cardinality constraint $|cliq| = k$ (i.e., the clique size is $k$) over a domain of size $n$ under the weighting $w(E) = 3, \bar{w}(E) = 1$ and $w(cliq) = \bar{w}(cliq) = 1$ samples an Erdös-Rényi graph (Erdös and Rényi 1960) $\mathcal{G}_{n,p=0.9}$ with planted clique of size $k$.*

We also adapt the notion of data complexity of WFOMC to the sampling problem, and say a UMS or WMS is *domain-lifted* (or simply lifted) if the model generation algorithm runs in time polynomial in the domain size $n$. We call a sentence, or class of sentences, that admit a domain-lifted UMS or WMS *domain-liftable under sampling*.

We remark that the uniform (resp. symmetric weighted) first-order sampling problem is not a trivial task, since the direct reduction from sampling to counting may be intractable. By the traditional reduction from sampling to counting, one

[2] The inequality $x \neq y$ can be easily realized by cardinality constraints (Beame et al. 2015, Lemma 3.5).

can sample a model $\mu$ of a sentence $\Gamma$ by iteratively determining whether an atom $a$ is true in $\mu$ by computing the conditional probability $\mathbb{P}[a \mid \Gamma \wedge \sigma] = \mathsf{WFOMC}(\Gamma \wedge \sigma \wedge a, n, w, \bar{w})/\mathsf{WFOMC}(\Gamma \wedge \sigma, n, w, \bar{w})$, where $\sigma$ is the conjunction of all sampled literal evidences. However, as stated by Van den Broeck and Davis (2012), there always exists a sentence $\Gamma$ in $\mathbf{UFO}^2$ and a conjunction $\sigma$ such that the WFOMC of $\Gamma \wedge \sigma$ cannot be computed by any algorithm whose runtime grows polynomially in the size of $\sigma$, unless $\mathsf{P} = \mathsf{NP}$.

## Domain-Liftability Under Sampling of $\mathbf{UFO}^2$

In this section, we show that any sentence in $\mathbf{UFO}^2$ is domain-liftable under sampling.

**Proposition 1.** *The fragment $\mathbf{UFO}^2$ is domain-liftable under sampling.*

To prove the proposition above, we will construct a WMS for any $\mathbf{UFO}^2$ sentence, and then prove it is domain-lifted.

### Main Algorithm

Fix an input sentence $\Gamma$ and domain $\Delta = \{c_1, c_2, \ldots, c_n\}$ of size $n$. Denote $\Omega = \{\{1,2\}, \{1,3\}, \ldots, \{n-1, n\}\}$ the set of all distinct index pairs over $[n]$. Let $\mathcal{P}_\Gamma$ be the set of all $m$ predicates in $\Gamma$.

Define a *unary literal* as an atomic predicate or its negation using only the variable $x$, and a *binary literal* an atomic predicate or its negation using both variables $x$ and $y$. Note that an atom like $R(x, x)$ or its negation is considered a unary literal, even though $R$ is a binary relation. A binary literal is always of the form $R(x, y)$ or $R(y, x)$, or their respective negations. A *unary type* (for short, U-type) $\tau$ is defined as a maximal consistent conjunction of unary literals, and a *binary type*[3] (for short, B-type) $\rho$ is a maximal consistent conjunction of binary literals. A *type* is either a U- or B-type. Note that a type can be viewed as a quantifier-free formula that is the conjunction of its elements. We use $\mathcal{U}$ and $\mathcal{B}$ to denote the set of all U-types and B-types, respectively, of a sentence.

**Example 2.** *Consider the formula $\psi(x, y) = (F(x, y) \vee G(x)) \vee (\neg G(x) \vee \neg H(x))$. Then there are $2^3 = 8$ U-types on $\psi(x, y)$; one such example is $\neg F(x, x) \wedge G(x) \wedge H(x)$. There are 4 B-types: $F(x, y) \wedge F(y, x)$, $F(x, y) \wedge \neg F(y, x)$, $\neg F(x, y) \wedge F(y, x)$ and $\neg F(x, y) \wedge \neg F(y, x)$.*

A type assignment $T$ over the domain $\Delta$ is an ordered list $(\tau_1, \tau_2, \ldots, \tau_n, \rho_{1,2}, \rho_{1,3}, \ldots, \rho_{n-1,n})$ that associates each domain element $c_i$ with a unary type $\tau_i \in \mathcal{U}$ and each distinct element pair $\{c_i, c_j\}$ with a binary type $\rho_{i,j} \in \mathcal{B}$. When associated with a domain element (resp. element pair), the U-type $\tau_i$ (resp. B-type $\rho_{i,j}$) can be grounded out to $\tau_i(c_i)$ (resp. $\rho_{i,j}(c_i, c_j)$). Any possible world $\omega$ is fully characterised by a unique type assignment $T$. Moreover, if we conjoin the grounding of all types in $T$ as $\Sigma_T = \bigwedge_{i \in [n]} \tau_i(c_i) \wedge \bigwedge_{\{i,j\} \in \Omega} \rho_{i,j}(c_i, c_j)$, $\mathsf{WFOMC}(\Gamma \wedge \Sigma_T, n, w, \bar{w})$ is exactly $W(\omega)$ if $\omega$ is a model of $\Gamma$, and 0 otherwise. Thus, sampling a model from the sentence $\Gamma$ according to (3) is equivalent to random generation of a type assignment $T$ with the probability

$$\mathbb{P}[T] = \frac{\mathsf{WFOMC}(\Gamma \wedge \Sigma_T, n, w, \bar{w})}{\mathsf{WFOMC}(\Gamma, n, w, \bar{w})}. \tag{4}$$

**Example 3.** *Let $\Gamma$ be the sentence from Example 1, $n = 2$ and $k = 2$. The type assignment*

$$T = (cliq(x) \wedge \neg E(x, x), cliq(x) \wedge \neg E(x, x), E(x, y) \wedge E(y, x))$$

*fully characterises a unique possible world $\omega = \{cliq(c_1), cliq(c_2), E(c_1, c_2), E(c_2, c_1)\}$. Since $\omega$ is a model of $\Gamma$, $\mathsf{WFOMC}(\Gamma \wedge \Sigma_T, n, w, \bar{w}) = W(\omega)$.*

For brevity, we simply use $\tau_i$ to denote both the U-type $\tau_i$ and its grounding $\tau_i(c_i)$ for a specific domain element $c_i$ in the rest of the paper. Similarly, $\rho_{i,j}$ denotes both $\rho_{i,j}$ and $\rho_{i,j}(c_i, c_j)$. We randomly assign all types $\tau_i$'s and $\rho_{i,j}$'s in an incremental manner:

- we start with a trivial sentence $\sigma = \top$,
- at the first step, we randomly assign the U-types $\tau_i$'s to all domain elements with the probability

  $$\mathbb{P}[\tau_1, \ldots, \tau_n] = \frac{\mathsf{WFOMC}(\Gamma \wedge \bigwedge_{i \in [n]} \tau_i, n, w, \bar{w})}{\mathsf{WFOMC}(\Gamma, n, w, \bar{w})},$$

  and conjoin the sampled corresponding grounding to $\sigma$, i.e., $\sigma \leftarrow \sigma \wedge \bigwedge_{i \in [n]} \tau_i$, and
- at each following step, we randomly select a B-type $\rho_{i,j}$ from $\mathcal{B}$ for the element pair $\{c_i, c_j\}$ with probability

  $$\mathbb{P}[\rho_{i,j}] = \frac{\mathsf{WFOMC}(\Gamma \wedge \sigma \wedge \rho_{i,j}, n, w, \bar{w})}{\mathsf{WFOMC}(\Gamma \wedge \sigma, n, w, \bar{w})} \tag{5}$$

  and conjoin the sampled $\rho_{i,j}$ to $\sigma$, i.e., $\sigma \leftarrow \sigma \wedge \rho_{i,j}$.

Once the algorithm has gone over all element pairs indexed in $\Omega$, the probability of the resulting type assignment is guaranteed to be (4) by the chain rule of probability.

The first step of randomly assigning all U-types is crucial to our algorithm. As mentioned before, computing the WFOMC of $\Gamma \wedge \sigma$ (and therefore (5)) may be intractable. However, later in the section, we shall see that if all U-types have been determined, we can compute (5) for any possible $\sigma$ and $\rho_{i,j}$ in a domain-lifted way.

The main sampling algorithm for $\mathbf{UFO}^2$ is presented in Algorithm 1. At the beginning of the algorithm, we randomly assign all U-types using a subroutine UnaryTypes which will be introduced in the next subsection. In lines 6-19, the algorithm traverses all pairs of distinct domain elements, and randomly assigns a B-type for each of them in lines 7-18.

### Complexity

The computation of Algorithm 1 comes mainly from UnaryTypes and computing the probability (5). We will analyse the computational complexity of each of them respectively.

[3] Note that although the definition presented here of a unary type is precisely the same as the notion of a *1-type* well known in the logic literature, binary types are defined differently from *2-types*, in that a binary type does not include information about the unary types of its two constituent elements.

Algorithm 1: Weighted Model Sampler for $\mathbf{UFO}^2$

**Input**: A $\mathbf{UFO}^2$ sentence $\Gamma$, a domain $\Delta$ of size $n$ and a weighting $(w, \bar{w})$
**Output**: A model $\mu$ of $\Gamma$ with the probability (3)

$T = (\tau_1, \ldots, \tau_n) \leftarrow \mathsf{UnaryTypes}(\Gamma, \Delta)$
$\sigma \leftarrow \bigwedge_{i \in [n]} \tau_i$
$W \leftarrow \mathsf{WFOMC}(\Gamma \wedge \sigma, n, w, \bar{w})$
$\Omega \leftarrow \{\{1, 2\}, \ldots, \{n-1, n\}\}$
$\mathcal{B} \leftarrow$ the set of all B-types of $\Gamma$
**for** $\{i, j\} \in \Omega$ **do**
  **for** $\rho \in \mathcal{B}$ **do**
    $W_\rho \leftarrow \mathsf{WFOMC}(\Gamma \wedge \sigma \wedge \rho(c_i, c_j), n, w, \bar{w})$
    // $\mathsf{Uniform}(0, 1)$ produces a random number from the uniform distribution over $[0, 1]$
    **if** $\mathsf{Uniform}(0, 1) < \frac{W_\rho}{W}$ **then**
      $\sigma \leftarrow \sigma \wedge \rho(c_i, c_j)$
      $W \leftarrow W_\rho$
      append $\rho$ to $T$
      **break**
    **else**
      $W \leftarrow W - W_\rho$
    **end if**
  **end for**
**end for**
**return** the unique possible world characterised by $T$

**UnaryTypes** We will provide a domain-lifted algorithm realizing $\mathsf{UnaryTypes}$. Denote $T_u = (\tau_1, \tau_2, \ldots, \tau_n)$ an assignment of U-types over $\Delta$. Like in the previous section, we denote by $\Sigma_{T_u}$ the conjunction of the groundings of types in $T_u$. Since $\mathcal{P}_\Gamma$ is of size $m$, the size of $\mathcal{U}$ is $2^m$. Let $\tau^{(j)}$ be the $j$th U-type in $\mathcal{U}$. For any U-types assignment $T_u$, we use $\mathbf{n}_{T_u} = (n_1, n_2, \ldots, n_{2^m})$ to denote the U-type cardinality vector of $T_u$, where each $n_j$ is the multiplicity of $\tau^{(j)}$ in $T_u$. Since predicate weightings are symmetric (i.e. they give the same weight to every ground atom of a given predicate), we have

$$\mathsf{WFOMC}(\Gamma \wedge \Sigma_{T_u}, n, w, \bar{w}) = \mathsf{WFOMC}(\Gamma \wedge \Sigma_{T'_u}, n, w, \bar{w}),$$

i.e., the probability of the U-types assignments $T_u$ and $T'_u$ is equal as long as $\mathbf{n}_{T_u} = \mathbf{n}_{T'_u}$. Hence, if the U-type cardinality vector $\mathbf{n}$ has already been sampled, we can randomly assign the U-types $\tau_1, \ldots, \tau_n$ by, for each $j \in [2^m]$, uniformly selecting $n_j$ elements from $\Delta$ without replacement and assigning them the U-type $\tau^{(j)}$.

The probability of every U-type cardinality vector $\mathbf{n}$ is proportional to $\sum_{T_u : \mathbf{n}_{T_u} = \mathbf{n}} \mathsf{WFOMC}(\Gamma \wedge \Sigma_{T_u}, n, w, \bar{w})$. We can sample it by

1. conjoining $\forall x : \xi_j(x) \leftrightarrow \tau^{(j)}$ to $\Gamma$ for each $j \in [2^m]$, where $\xi_j$ is a new predicate with neutral weighting $w_j(\xi) = \bar{w}_j(\xi) = 1$, to obtain a new sentence $\Gamma'$;
2. computing the count distribution of $\Psi = \{\xi_1, \ldots, \xi_{2^m}\}$ conditional on $\Gamma'$ as $q_{\Gamma', \Psi}$;
3. generating a vector $\mathbf{n}$ according to $q_{\Gamma', \Psi}$.

Since $\mathbf{UFO}^2$ is a subfragment of $\mathbf{FO}^2$ and thus domain-liftable (den Broeck, Meert, and Darwiche 2014), computing the count distribution $q_{\Gamma', \Psi}$ is tractable by Theorem 1.

Finally, we remark here that thanks to Theorem 1, the method proposed above for sampling U-types can generalise beyond $\mathbf{UFO}^2$ as long as the first-order sentence $\Gamma$ is liftable.

**Computing Probability** Next, let us deal with the probability (5). It is sufficient to prove that there exists a domain-lifted algorithm for computing $\mathsf{WFOMC}(\Gamma \wedge \sigma, n, w, \bar{w})$ for any $\sigma$ appearing in Algorithm 1. Denote the set of index pairs that have been processed or are being processed as $\Omega' \subseteq \Omega$.

For any $\sigma$ appearing in the algorithm, we can always write it as $\sigma = \Sigma_{T_u} \wedge \bigwedge_{\{i,j\} \in \Omega'} \rho_{i,j}$ where $T_u = \{\tau_1, \ldots, \tau_n\}$ is a U-types assignment and $\rho_{i,j} \in \mathcal{B}$ for all $\{i, j\} \in \Omega'$. The grounding of the sentence $\Gamma = \forall x \forall y : \psi(x, y)$ over the domain $\Delta$ can be written as

$$\eta = \bigwedge_{i \in [n]} \psi(c_i, c_i) \wedge \bigwedge_{\{i,j\} \in \Omega} (\psi(c_i, c_j) \wedge \psi(c_j, c_i)).$$

For all $\{i, j\} \in \Omega$, let $\psi'(c_i, c_j)$ denote the ground sentence $\psi(c_i, c_j) \wedge \psi(c_j, c_i)$, where all determined atoms have been replaced by true or false, according to $\tau_i$ and $\tau_j$. Then we have

$$\eta \wedge \sigma = \Sigma_{T_u} \wedge \bigwedge_{\{i,j\} \in \Omega'} (\rho_{i,j} \wedge \psi'(c_i, c_j)) \wedge \bigwedge_{\{i,j\} \in \Omega \setminus \Omega'} \psi'(c_i, c_j).$$

Observe that each conjunct in the formula above is independent, (that is, they do not share any propositional variables). Denote $w_i = \mathsf{WMC}(\tau_i, w, \bar{w})$, $\hat{r}_{i,j} = \mathsf{WMC}(\rho_{i,j} \wedge \psi'(c_i, c_j), w, \bar{w})$ and $r_{i,j} = \mathsf{WMC}(\psi'(c_i, c_j), w, \bar{w})$, where $\mathsf{WMC}(\cdot, \cdot, \cdot)$ denotes the weighted model count (Beame et al. 2015). We can thus write the WFOMC of $\Gamma \wedge \sigma$ as

$$\mathsf{WFOMC}(\Gamma \wedge \sigma, n, w, \bar{w}) = \prod_{i \in [n]} w_i \cdot \prod_{\{i,j\} \in \Omega'} \hat{r}_{i,j} \cdot \prod_{\{i,j\} \in \Omega \setminus \Omega'} r_{i,j}. \tag{6}$$

Computing each $w_i$, $\hat{r}_{i,j}$ and $r_{i,j}$ is independent of $n$, and the total number of $w_i$, $\hat{r}_{i,j}$ and $r_{i,j}$ is at most $n^2$. It is easy to check that evaluating (6) is domain-lifted.

*Proof of Proposition 1.* We prove that Algorithm 1 is a domain-lifted WMS for any $\mathbf{UFO}^2$ sentence. Algorithm 1 needs to call $\mathsf{UnaryTypes}$ once and compute $O(n^2)$ probabilities (5) to sample a model. By the above analysis of $\mathsf{UnaryTypes}$ and (5), the total runtime of Algorithm 1 is clearly polynomial in the domain size $n$. □

## Domain-Liftability Under Sampling of $\mathbf{UFO}^2$ with Tree and Cardinality Constraints

In addition to our sampling result for the fragment $\mathbf{UFO}^2$ discussed in the previous section, in this section, we will also be proving domain-liftability results under sampling for *tree* and *cardinality constraints*, which are not necessarily expressible in first-order logic without grounding out the

Algorithm 2: Weighted Model Sampler for $\mathbf{UFO}^2$ with Constraints

**Input**: An $\mathbf{UFO}^2$ sentence $\Gamma$, a constraint $\mathcal{C}$, a domain $\Delta$ of size $n$ and a weighting $(w, \bar{w})$
**Output**: A model $\mu$ from $\text{models}_{\text{n}}(\Gamma \wedge \mathcal{C})$ with the probability (7)

$T = (\tau_1, \ldots, \tau_n) \leftarrow \mathsf{UnaryTypes}(\Gamma, \mathcal{C}, \Delta)$
...
$W \leftarrow \mathsf{WFOMC}(\Gamma \wedge \sigma \wedge \mathcal{C}, n, w, \bar{w})$
...
$W_\rho \leftarrow \mathsf{WFOMC}(\Gamma \wedge \sigma \wedge \rho(c_i, c_j) \wedge \mathcal{C}, n, w, \bar{w})$
...

constraint in question. This is following recent analogous results for WFOMC (Kuželka 2021; van Bremen and Kuželka 2021).

We therefore define a *constraint* $\mathcal{C}$ on a sentence $\Gamma$ as a function mapping every possible world that can be formed on the predicates in $\Gamma$ to $\{\top, \bot\}$. For instance, a tree constraint on a binary predicate $R$ requires that the ground atoms of $R$ in $\omega$ form a tree (mapping such possible worlds to $\top$, and others $\bot$). Multiple constraints can always be combined into a single one by considering the logical "and" of their outputs. The satisfaction relation $\models$ and the logical and symbol $\wedge$ are extended naturally for constraints: $\omega \models \Gamma \wedge \mathcal{C}$ means that $\omega \models \Gamma$ and $\mathcal{C}(\omega) = \top$. Other notation and terminology is also extended accordingly by taking $\Gamma \wedge \mathcal{C}$ as a *pseudo-sentence*, e.g., $\text{models}_{\text{n}}(\Gamma \wedge \mathcal{C})$ denotes all possible worlds satisfying $\Gamma \wedge \mathcal{C}$. We note here that the reduction in Theorem 1 from computation of a count distribution to WFOMC still works for pseudo-sentences, and refer the reader to Appendix for the details.

## WMS for $\mathbf{UFO}^2$ with Constraints

The uniform and symmetric weighted sampling problems for $\mathbf{UFO}^2$ with constraints are defined analogous to Definition 4 and 5 respectively. In this section, we consider the symmetric weighted sampling problem (of which the uniform sampling problem is a special case). Recall that the symmetric weighted sampling problem on a given sentence $\Gamma$ with constraint $\mathcal{C}$ under $(w, \bar{w})$ requires the probability of any generated model $G(\Gamma, \mathcal{C}, n, w, \bar{w})$ to be

$$\mathbb{P}[G(\Gamma, \mathcal{C}, n, w, \bar{w}) = \mu] = \frac{W_{w,\bar{w}}(\mu)}{\mathsf{WFOMC}(\Gamma \wedge \mathcal{C}, n, w, \bar{w})} \tag{7}$$

for every $\mu \in \text{models}_{\text{n}}(\Gamma \wedge \mathcal{C})$. We generalise the terms *WMS*, *lifted* and *domain-liftable under sampling* to the constraint case in the natural way.

We construct a WMS for $\mathbf{UFO}^2$ with constraints in Algorithm 2, which is derived based on Algorithm 1 by considering the constraints when executing the UnaryTypes subprocedure and computing WFOMCs. Sampling cell types for a $\mathbf{UFO}^2$ sentence $\Gamma$ with constraint $\mathcal{C}$ can be realised in almost the same way described in Section , except that we sample the cardinality vector of U-types using the count distribution $q_{\Gamma' \wedge \mathcal{C}, \Psi}$ rather than $q_{\Gamma', \Psi}$.

## Tree Constraints

Let us first consider the tree constraint. For any possible world $\omega$ and predicate $P \in \mathcal{P}_\Gamma$, we denote by $\omega_P$ the set of ground atoms for $P$ that are true in $\omega$. Recall from (van Bremen and Kuželka 2021) that a tree constraint on some distinguished binary predicate $R$ on a possible world $\omega$, denoted by $\mathsf{T}_R(\omega)$, returns $\top$ i.f.f.

- the relation defined by $\omega_R$ is antireflexive and symmetric, and
- $\omega_R$ is a tree when interpreted as an undirected graph.

Though it may seem complicated, we show that a single tree constraint does not hurt the liftability of $\mathbf{UFO}^2$.

**Proposition 2.** *The fragment $\mathbf{UFO}^2$ with a single tree constraint is domain-liftable under sampling.*

*Proof.* It is sufficient to prove that Algorithm 2 is domain-lifted for any $\mathbf{UFO}^2$ sentence $\Gamma$ with any single tree constraint $\mathsf{T}_R$. According to the proof of Proposition 1, we only need to analyse the complexity of UnaryTypes and WFOMC of $\Gamma \wedge \sigma \wedge \mathsf{T}_R$. As discussed before, UnaryTypes is clearly domain-lifted, since computing the count distribution is reducible to the WFOMC problem of $\mathbf{UFO}^2$ with a single tree constraint, which has been proved to be domain-liftable in (van Bremen and Kuželka 2021).

Let us analyse $\mathsf{WFOMC}(\Gamma \wedge \sigma \wedge \mathsf{T}_R, n, w, \bar{w})$. Adopt all notations in Section . Let $\mathcal{F}_\sigma$ and $\mathcal{F}'_\sigma$ be the set of all index pairs $\{i, j\} \in \Omega'$ for which $R(c_i, c_j)$ and $\neg R(c_i, c_j)$ is true in $\sigma$ respectively. Denote $r^+_{i,j} = \mathsf{WMC}(\psi'(c_i, c_j) \wedge R(c_i, c_j), w, \bar{w})$ and $r^-_{i,j} = \mathsf{WMC}(\psi'(c_i, c_j) \wedge \neg R(c_i, c_j), w, \bar{w})$. Let $\mathcal{F}$ be the set of index pairs $\{i, j\} \in \Omega \backslash \Omega'$ such that $r^-_{i,j} = 0$. Let $\Omega'' = \Omega \backslash \Omega'$ be the set of indices that algorithm does not reach yet. Recall $w_i = \mathsf{WMC}(\tau_i, w, \bar{w})$ and $\hat{r}_{i,j} = \mathsf{WMC}(\rho_{i,j}, w, \bar{w})$. Consider some tree $T$ over $[n]$ such that $\mathcal{F}_\sigma \cup \mathcal{F} \subseteq T$ and does not contain any edges in $\mathcal{F}'_\sigma$. We will compute the weighted sum $W_T$ of every model $\mu$ of $\Gamma \wedge \sigma$ that satisfies $\mu_R = T$ (in other words, the WFOMC of $\Gamma \wedge \sigma$ limited to models that contain exactly the tree $T$). We can write:

$$\begin{aligned} W_T &= \prod_{i \in [n]} w_i \cdot \prod_{\{i,j\} \in \Omega'} \hat{r}_{i,j} \cdot \prod_{\{i,j\} \in T \cap \Omega''} r^+_{i,j} \cdot \prod_{\{i,j\} \in \Omega'' \backslash T} r^-_{i,j} \\ &= \prod_{i \in [n]} w_i \cdot \prod_{\{i,j\} \in \Omega'} \hat{r}_{i,j} \cdot \prod_{\{i,j\} \in \mathcal{F}} r^+_{i,j} \cdot \prod_{\{i,j\} \in \Omega'' \backslash \mathcal{F}} r^-_{i,j} \cdot \\ &\quad \prod_{\{i,j\} \in T \cap (\Omega'' \backslash \mathcal{F})} \frac{r^+_{i,j}}{r^-_{i,j}} \end{aligned}$$

Summing both sides of the equation across all trees containing $\mathcal{F}_\sigma \cup \mathcal{F}$ while not containing any edges in $\mathcal{F}'_\sigma$ is the

WFOMC of $\Gamma \wedge \sigma$ with the tree constraint $\mathsf{T}_R$:

$$
\begin{aligned}
&\mathsf{WFOMC}(\Gamma \wedge \sigma, \mathsf{T}_R, n, w, \bar{w}) \\
&= \sum_{\substack{T \subseteq \Omega \setminus \mathcal{F}'_\sigma \\ \mathcal{F}_\sigma \cup \mathcal{F} \subseteq T}} W_T \\
&= \prod_{i \in [n]} w_i \cdot \prod_{\{i,j\} \in \Omega'} \hat{r}_{i,j} \cdot \prod_{\{i,j\} \in \mathcal{F}} r^+_{i,j} \cdot \prod_{\{i,j\} \in \Omega'' \setminus \mathcal{F}} r^-_{i,j} \cdot \\
&\quad \sum_{\substack{T \subseteq \Omega \setminus \mathcal{F}'_\sigma \\ \mathcal{F}_\sigma \cup \mathcal{F} \subseteq T}} \prod_{\{i,j\} \in \mathcal{F}_\sigma \cup \mathcal{F}} 1 \prod_{\{i,j\} \in T \cap (\Omega'' \setminus \mathcal{F})} \frac{r^+_{i,j}}{r^-_{i,j}} \\
&= \prod_{i \in [n]} w_i \cdot \prod_{\{i,j\} \in \Omega'} \hat{r}_{i,j} \cdot \prod_{\{i,j\} \in \mathcal{F}} r^+_{i,j} \cdot \prod_{\{i,j\} \in \Omega'' \setminus \mathcal{F}} r^-_{i,j} \cdot \\
&\quad \mathsf{TS}(A, \mathcal{F}_\sigma \cup \mathcal{F})
\end{aligned}
\tag{8}
$$

where $A$ is the graph with weighted adjacency matrix:

$$
A_{(i,j)} = \begin{cases} \frac{r^+_{i,j}}{r^-_{i,j}} & \text{if } \{i,j\} \in \Omega'' \setminus \mathcal{F} \\ 0 & \text{if } \{i,j\} \in \mathcal{F}'_\sigma \\ 1 & \text{otherwise} \end{cases}
$$

Since the quantity $\mathsf{TS}(A, \mathcal{F}_\sigma \cup \mathcal{F})$ can be computed in time polynomial in the size $n$ of $A$, we can conclude that evaluating this WFOMC is also domain-lifted, which completes the proof. □

## Cardinality Constraints

Next, let us incorporate cardinality constraints into the sampling problem. Given a sentence $\Gamma$ containing predicates $\mathcal{P}_\Gamma$, a cardinality constraint w.r.t. $\Gamma$ is a function on possible worlds

$$
\mathcal{C}_f(\omega) := \begin{cases} \top & \text{if } f(\mathbf{N}(\mathcal{P}_\Gamma, \omega)) = 1 \\ \bot & \text{otherwise} \end{cases}
$$

where $f \in \mathbb{N}^{|\mathcal{P}_\Gamma|} \to \{0,1\}$. For brevity, when the function $f$ is in the form $\mathbb{1}[|P_i| = n_i]$ where $P_i$ is a predicate, $n_i \in \mathbb{N}$, and $\mathbb{1}[\cdot]$ is the indicator function, we can simply write the cardinality constraint $\mathcal{C}_f$ as $|P_i| = n_i$.

We are now ready to present our final result on the domain-liftability under sampling of $\mathbf{UFO}^2$, this time with tree and cardinality constraints.

**Theorem 2.** *The fragment $\mathbf{UFO}^2$ with a single tree constraint and multiple cardinality constraints is domain-liftable under sampling.*

*Proof.* Given a $\mathbf{UFO}^2$ sentence $\Gamma$, a tree constraint $\mathsf{T}_R$ and multiple cardinality constraints $\mathcal{C}_{f_1}, \ldots, \mathcal{C}_{f_M}$, we can first construct a new cardinality constraint $\mathcal{C}_f$, where $f = \min_{i=1}^{M} f_i$, and then combine it with $\mathsf{T}_R$ resulting a single constraint $\mathcal{C}$. We will prove Algorithm 2 is domain-lifted for any such constraint.

Following the same argument of Proposition 1 and 2, we investigate the complexity of $\mathsf{UnaryTypes}(\Gamma, \mathcal{C}, \Delta)$ and $\mathsf{WFOMC}(\Gamma \wedge \sigma \wedge \mathcal{C}, n, w, \bar{w})$ respectively. Similarly, $\mathsf{UnaryTypes}(\Gamma, \mathcal{C}, \Delta)$ is easily proved domain-lifted since computation of the necessary count distribution is reducible to computation of the WFOMC of $\Gamma \wedge \mathsf{T}_R \wedge \mathcal{C}_f$, which is domain-liftable (van Bremen and Kuželka 2021).

Next, let us consider $\mathsf{WFOMC}(\Gamma \wedge \sigma \wedge \mathcal{C}, n, w, \bar{w})$. Let $\mathcal{D} = \times_{P \in \mathcal{P}_\Gamma} \{0, \ldots, n^{arity(P)}\}$. Recall $f$ is a function mapping every $\mathbf{n} \in \mathcal{D}$ to $\{0,1\}$, and

$$
q_{\Gamma \wedge \sigma \wedge \mathsf{T}_R, \mathcal{P}_\Gamma}(\mathbf{n}) := \sum_{\substack{\mu \in \mathrm{models}_{\mathrm{n}}(\Gamma \wedge \sigma \wedge \mathsf{T}_\mathrm{R}): \\ \mathbf{N}(\mathcal{P}_\Gamma, \mu) = \mathbf{n}}} \frac{W(\mu)}{\mathsf{WFOMC}(\Gamma \wedge \sigma \wedge \mathsf{T}_R, n, w, \bar{w})}.
$$

The WFOMC of a sentence $\Gamma \wedge \sigma$ with constraint $\mathcal{C}$ can be written as

$$
\begin{aligned}
&\mathsf{WFOMC}(\Gamma \wedge \sigma \wedge \mathcal{C}, n, w, \bar{w}) \\
&\quad = \sum_{\mu \in \mathrm{models}_{\mathrm{n}}(\Gamma \wedge \sigma \wedge \mathcal{C})} W(\mu) \\
&\quad = \sum_{\mathbf{n} \in \mathcal{D}} f(\mathbf{n}) \cdot \sum_{\substack{\mu \in \mathrm{models}_{\mathrm{n}}(\Gamma \wedge \sigma \wedge \mathsf{T}_\mathrm{R}): \\ \mathbf{N}(\mathcal{P}_\Gamma, \mu) = \mathbf{n}}} W(\mu) \\
&\quad = \mathsf{WFOMC}(\Gamma \wedge \sigma \wedge \mathsf{T}_R, n, w, \bar{w}) \cdot \\
&\qquad \sum_{\mathbf{n} \in \mathcal{D}} f(\mathbf{n}) \cdot q_{\Gamma \wedge \sigma \wedge \mathsf{T}_R, \mathcal{P}_\Gamma}(\mathbf{n}).
\end{aligned}
$$

Computing the term $\mathsf{WFOMC}(\Gamma \wedge \sigma \wedge \mathsf{T}_R, n, w, \bar{w})$ has been proven domain-lifted by Proposition 2. The remaining summation is over the count distribution of $\mathcal{P}_\Gamma$ conditioned on $\Gamma \wedge \sigma \wedge \mathsf{T}_R$, which is however also reducible to the WFOMC of the liftable $\Gamma \wedge \sigma \wedge \mathsf{T}_R$. It is easy to check computing $\mathsf{WFOMC}(\Gamma \wedge \sigma \wedge \mathcal{C}, n, w, \bar{w})$ is in time polynomial in $n$, which the conclusion follows. □

## Algorithmic Improvements

We further optimise Algorithm 2 to make it more efficient in practice. Briefly, we devise a faster UnaryTypes and use caching to avoid recomputing WFOMCs and WMCs throughout the algorithm. Details can be found in the online technical report[4].

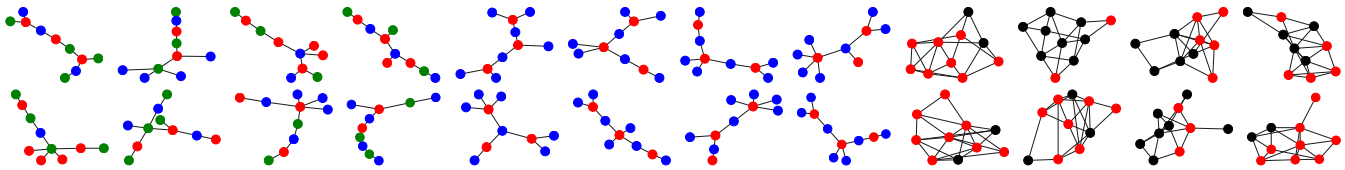
# Applications and Experiments

In this section, we present some applications of the first-order sampling problem and examine the scalability of our algorithm in practice. All experiments were performed on a computer with an 8-core Intel i7 3.60GHz processor and 32 GB of RAM.

## Sampling Combinatorial Structures

We first consider the problem of counting $k$-colored trees. Recall that a graph is said to be $k$-colorable if every vertex can be colored with one of $k$ colors such that no two vertices of the same color are adjacent to one another. A $k$-colored graph is a $k$-colorable graph together with a valid color assignment. We evaluated our algorithm on 2-, 3-, and 4-colored trees, and plotted a few examples of 3-colored trees of size 10 in Figure 1a.

[4] https://github.com/lucienwang1009/lifted_sampling_ufo2



(a) (b) (c)

Figure 1: (a) Sampled 3-colored (red, blue, and green) trees of size 10. (b) Sampled 2-colored (red and blue) trees of size 10 with cardinality constraint $|red| = 3$. (c) Examples of the sampled possible worlds from "friends-smokers" MLN over a domain of size 10. Red vertices denote smokers, and black vertices non-smokers. Two vertices are connected if the corresponding pair of people are friends (best viewed in color).

We next consider the problem posed in the introduction of the paper: uniform generation of a labelled 2-colored tree of size $n$, where the first color is used $k$ times and the other $n-k$ times. This problem can be encoded as a $\mathbf{UFO}^2$ sentence with tree and cardinality constraints, as already shown in the introduction. We present a few such trees generated by our algorithm with $n = 10$ and $k = 3$ in Figure 1b.

## Sampling from MLNs

We now turn to applications of our approach to sampling possible worlds from MLNs. Recall that an MLN is a finite set of weighted first-order formulas $\{(w_1, \alpha_1), \ldots, (w_m, \alpha_m)\}$, where each $w_i$ is either a real-valued weight or $\infty$, and $\alpha_i$ is a quantifier-free first-order formula. An MLN $\Phi$ paired with a domain $\Delta$ induces a probability distribution over possible worlds:

$$p_{\Phi,\Delta}(\omega) = \begin{cases} \frac{1}{Z_{\Phi,\Delta}} \exp\left(\sum_{(\alpha,w)\in\Phi_{\mathbb{R}}} w \cdot N(\alpha, w)\right) & \text{if } \omega \models \Phi_\infty \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where $\Phi_{\mathbb{R}}$ and $\Phi_\infty$ denote the real-valued and $\infty$-valued formulas in $\Phi$ respectively, $N(\alpha, w)$ is the number of groundings of $\alpha$ satisfied in $\omega$, and $Z_{\Phi,\Delta}$, called the partition function, is a normalisation constant. The sampling problem on an MLN $\Phi$ over a domain $\Delta$ is to randomly generate a possible world $\omega$ with the probability $\mathbb{P}[\omega] = p_{\Phi,\Delta}(\omega)$.

Following the reduction from the partition function to WFOMC (Van den Broeck et al. 2011), we can reduce the sampling problem on MLNs to a symmetric weighted first-order sampling problem introduced in this paper. Given an MLN $\Phi$, for every real-valued formula $(\alpha_i, w_i) \in \Phi_{\mathbb{R}}$, where the free variables in $\alpha_i$ are $\mathbf{x} = \{x_1, \ldots, x_k\}$, we create a new formula $\forall \mathbf{x}, \xi_i(\mathbf{x}) \leftrightarrow \alpha_i(\mathbf{x})$, where $\xi_i$ is an auxiliary predicate. When $\alpha_i$ has $w_i = \infty$, we instead create a new formula $\forall \mathbf{x}, \alpha_i(\mathbf{x})$. We denote the conjunction of the resulting set of sentences by $\Gamma$ 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. One can check that any possible world satisfying $\Phi_\infty$ corresponds to a model $\mu$ of $\Gamma$ by removing all atoms with auxiliary predicates $\xi_i$'s, and the required probability (3) of $\mu$ is equal to (9).

We note here, the aforementioned reduction to symmetric weighted first-order sampling has the helpful property that the number of variables in the sentence produced by the reduction is the same as the maximum number of variables appearing in any formula of the original MLN. In particular, this means that if the number of variables used in each of the formulas in the MLN is limited to two, the reduced sentence is in $\mathbf{UFO}^2$ and thus liftable under sampling.

We consider the sampling problem on the classic "friends-smokers" MLN:

$$\Phi = \{(\infty, \neg fr(x,x)), (\infty, fr(x,y) \rightarrow fr(y,x)), \\ (0.2, fr(x,y) \wedge sm(x) \rightarrow sm(y)), (0, sm(x))\}.$$

This MLN models that people who are friends with smokers are likely to smoke themselves. Examples of the sampled possible worlds of $\Phi$ over a domain of size 10 is illustrated in Figure 1c.

## Performance Analysis

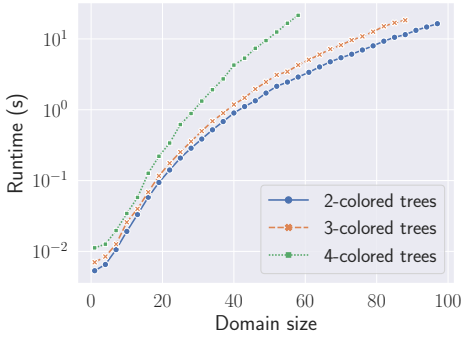
We further analyse the scalability of our algorithm on the examples of $k$-colored trees and the "friends-smokers" MLN. For the experiment on $k$-colored trees, we evaluated the average runtime across different domain sizes (with 10 samples for every size) for 2-, 3- and 4-colored trees and plotted it in Figure 2a. Similarly, the runtime for sampling possible worlds from the "friends-smokers" MLN over different domain sizes is shown in Figure 2b.

To the best of our knowledge, this is the first exact lifted sampling algorithm for first-order logic, so the closest comparison would be against a state-of-the-art propositional sampler like UniGen (Soos, Gocht, and Meel 2020). We ran UniGen on the "friends-smokers" MLN, but it could only scale to domains of size 30, whereas our approach takes less than 1 second on a domain of size 100, as shown in Fig. 2b.

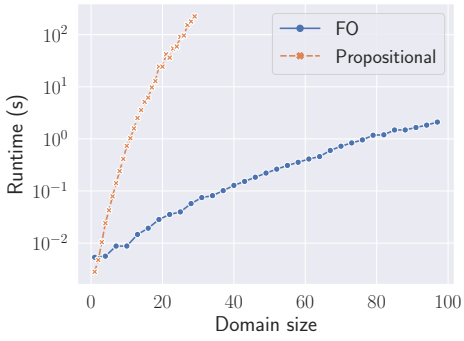
## Discussion and Conclusion

We introduced the problem of *first-order sampling*, as well as the notion of *domain-liftability under sampling*. We further showed that the fragment $\mathbf{UFO}^2$ is domain-liftable under sampling, even with the addition of cardinality constraints and a single tree constraint.

A natural direction for future work is to see if our results can be extended from $\mathbf{UFO}^2$ to $\mathbf{FO}^2$, or even further beyond to the two-variable fragment with counting quantifiers $\mathbf{C}^2$. Analogous results have been obtained for domain-liftability in the *counting* setting, however, we conjecture




(a)




(b)

Figure 2: (a) Runtime for sampling 2-, 3-, and 4-colored trees. (b) Comparison of computation between our algorithm (FO) and propositional sampling method (Propositional) for the "friends-smokers" MLN for various domain size.

that transferring this to the sampling setting would require heavier machinery or may not even be possible at all: this raises questions to what extent the classic equivalence between counting and sampling holds in the first-order context. For example, a UMS for $\mathbf{C}^2$ can be used to solve the uniform generation problem of $k$-regular graphs, on which there have been highly non-trivial works, e.g., (Gao and Wormald 2017). On the other hand, we rigorously show in Appendix that directly applying our sampling algorithm on an $\mathbf{FO}^2$ sentence with existential quantifiers is intractable (not domain-lifted) unless $\mathsf{FP} = \#\mathsf{P}$, which suggests that an entirely different approach is necessary. Thus, sampling for the full fragment of $\mathbf{FO}^2$, and even more so for $\mathbf{C}^2$, is an interesting challenge for the future.

## Acknowledgments

TvB is supported by the Research Foundation – Flanders (G095917N), and the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 694980, project SYNTH). OK is supported by the Czech Science Foundation project "Generative Relational Models" (20-19104Y).

# Appendix

## Reduction of Count Distribution for First-order Sentence with Constraints

It is easy to prove the reduction from computation of count distributions to WFOMC for sentences with constraints, following the proof of Theorem 1 in (Kuželka 2020).

**Corollary 1.** *For any set of predicates $\Psi$ and constraint $\mathcal{C}$, if there is a domain-lifted oracle for computing $\mathsf{WFOMC}(\Gamma \wedge \mathcal{C}, n, w, \bar{w})$, the count distribution $q_{\Gamma\wedge\mathcal{C},\Psi}$ can be obtained by a polynomial number of queries to this oracle.*

*Proof.* Given a sentence $\Gamma$ with constraint $\mathcal{C}$, a list of predicates $\Psi = \{P_1, \ldots, P_m\}$ and a domain of size $n$, the domain[5] of its count distribution $q_{\Gamma\wedge\mathcal{C},\Psi}$ is $\mathcal{D} = \times_{j=1}^{m}\{0, 1, \ldots, n^{arity(P_j)}\}$. Let $\mathbf{M} = [n^{arity(P_1)} + 1, \ldots, n^{arity(P_m)} + 1]$. Extending the range of a weighting from non-negative real values to complex numbers, and applying Discrete Fourier Transform (DFT) and inverse DFT (Kuželka, Kungurtsev, and Wang 2020) on $q_{\Gamma\wedge\mathcal{C},\Psi}$, we have

$$g_{\Gamma\wedge\mathcal{C},\Psi}(\mathbf{k}) = \frac{\mathsf{WFOMC}(\Gamma \wedge \mathcal{C}, n, w_{\mathbf{k}}, \bar{w}_{\mathbf{k}})}{\mathsf{WFOMC}(\Gamma \wedge \mathcal{C}, n, w, \bar{w})}, \tag{10}$$

where for all $j \in [m]$, $w_{\mathbf{k}}(P_j) = w(P_j) \cdot e^{-i2\pi\mathbf{k}_j/\mathbf{M}_j}$ and $\bar{w}_{\mathbf{k}}(P_j) = \bar{w}(P_j)$, and

$$q_{\Gamma\wedge\mathcal{C},\Psi}(\mathbf{n}) = \frac{\sum_{\mathbf{k}\in\mathcal{D}} g_{\Gamma,\Psi}(\mathbf{k}) \cdot \exp(i2\pi\langle\mathbf{n}, \mathbf{k}/\mathbf{M}\rangle)}{\prod_{j=1}^{m}\mathbf{M}_j} \tag{11}$$

where $\mathbf{k}/\mathbf{M}$ denotes component-wise division, and $\langle\cdot,\cdot\rangle$ is the inner product. By (10) and (11), the number of WFOMC calls required to compute the count distribution is in polynomial in $n$. □

## A Counterexample with Existential Quantifiers

We provide an $\mathbf{FO}^2$ sentence with existential quantifiers that does not admit efficient sampling by our algorithm:

$$\begin{aligned}\Gamma =& \forall x \forall y : E(x,y) \rightarrow E(y,x) \wedge \\ & \forall x : \neg E(x,x) \wedge \\ & \forall x \exists y : E(x,y).\end{aligned}$$

Uniform sampling here is equivalent to uniform generation of an undirected graph with no isolated vertices. Note there is an efficient rejection sampling algorithm for this problem. However, we will show that Algorithm 1 fails to process $\Gamma$ in time polynomial in the domain size.

When we run Algorithm 1 with $\Gamma$, we always need to compute the WFOMC of $\Gamma \wedge \bigwedge_{\{i,j\}\notin\mathcal{E}} \neg E(i,j)$ for all possible sets $\mathcal{E}$ of element pairs. By viewing every $\mathcal{E}$ as an undirected graph $G$ indexed in the domain, whose edges are exactly $\mathcal{E}$, the WFOMC of $\Gamma \wedge \bigwedge_{\{i,j\}\notin\mathcal{E}} \neg E(i,j)$ under the weighting mapping all predicates to 1 corresponds to counting the edge coverings of $G$, which is known to be #P-complete. In other words, there always exists a set of element pairs $\mathcal{E}$ such that the WFOMC of $\Gamma \wedge \bigwedge_{\{i,j\}\notin\mathcal{E}} \neg E(i,j)$ cannot be computed by any domain-lifted algorithm, unless $\mathsf{FP} = \#\mathsf{P}$.

[5]Here, domain refers to the domain of a mathematical function, not to a domain as a set of domain elements.

## References


Beame, P.; Van den Broeck, G.; Gribkoff, E.; and Suciu, D. 2015. Symmetric Weighted First-Order Model Counting. In *PODS*, 313–328. ACM.

Chaiken, S.; and Kleitman, D. J. 1978. Matrix tree theorems. *J. Comb. Theory Ser. A.*, 24(3): 377–381.

Chakraborty, S.; Fremont, D. J.; Meel, K. S.; Seshia, S. A.; and Vardi, M. Y. 2014. Distribution-Aware Sampling and Weighted Model Counting for SAT. In *AAAI*, 1722–1730. AAAI Press.

Chakraborty, S.; Meel, K. S.; and Vardi, M. Y. 2013. A Scalable and Nearly Uniform Generator of SAT Witnesses. In *CAV*, volume 8044 of *Lecture Notes in Computer Science*, 608–623. Springer.

De Raedt, L.; Kersting, K.; Natarajan, S.; and Poole, D. 2016. *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

de Salvo Braz, R.; Amir, E.; and Roth, D. 2005. Lifted First-Order Probabilistic Inference. In *IJCAI*, 1319–1325. Professional Book Center.

den Broeck, G. V.; Meert, W.; and Darwiche, A. 2014. Skolemization for Weighted First-Order Model Counting. In *KR*. AAAI Press.

Erdös, P.; and Rényi, A. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1): 17–60.

Gao, P.; and Wormald, N. C. 2017. Uniform Generation of Random Regular Graphs. *SIAM J. Comput.*, 46(4): 1395–1427.

Gogate, V.; and Domingos, P. M. 2011. Probabilistic Theorem Proving. In *UAI*, 256–265. AUAI Press.

Gogate, V.; Jha, A. K.; and Venugopal, D. 2012. Advances in Lifted Importance Sampling. In *AAAI*. AAAI Press.

Gomes, C. P.; Sabharwal, A.; and Selman, B. 2006. Near-Uniform Sampling of Combinatorial Spaces Using XOR Constraints. In *NIPS*, 481–488. MIT Press.

Jerrum, M.; Valiant, L. G.; and Vazirani, V. V. 1986. Random Generation of Combinatorial Structures from a Uniform Distribution. *Theor. Comput. Sci.*, 43: 169–188.

Kuusisto, A.; and Lutz, C. 2018. Weighted model counting beyond two-variable logic. In *LICS*, 619–628. ACM.

Kuželka, O. 2020. Complex Markov Logic Networks: Expressivity and Liftability. In *UAI*, volume 124 of *Proceedings of Machine Learning Research*, 729–738. AUAI Press.

Kuželka, O. 2021. Weighted First-Order Model Counting in the Two-Variable Fragment With Counting Quantifiers. *J. Artif. Intell. Res.*, 70: 1281–1307.

Kuželka, O.; Kungurtsev, V.; and Wang, Y. 2020. Lifted Weight Learning of Markov Logic Networks (Revisited One More Time). In *PGM*, volume 138 of *Proceedings of Machine Learning Research*, 269–280. PMLR.

Lin, T.; Lu, C.; and Tan, T. 2021. Towards a more efficient approach for the satisfiability of two-variable logic. In *LICS*, 1–13. IEEE.

Poole, D. 2003. First-order probabilistic inference. In *IJCAI*, 985–991. Morgan Kaufmann.

Richardson, M.; and Domingos, P. M. 2006. Markov logic networks. *Mach. Learn.*, 62(1-2): 107–136.

Soos, M.; Gocht, S.; and Meel, K. S. 2020. Tinted, Detached, and Lazy CNF-XOR solving and its Applications to Counting and Sampling. In *Proceedings of International Conference on Computer-Aided Verification (CAV)*.

Taghipour, N.; Fierens, D.; Davis, J.; and Blockeel, H. 2013. Lifted Variable Elimination: Decoupling the Operators from the Constraint Language. *J. Artif. Intell. Res.*, 47: 393–439.

van Bremen, T.; and Kuželka, O. 2021. Lifted Inference with Tree Axioms. In *KR*, 599–608.

van Bremen, T.; and Kuželka, O. 2021. Faster lifting for two-variable logic using cell graphs. In *Uncertainty in Artificial Intelligence*, 1393–1402. PMLR.

Van den Broeck, G.; and Davis, J. 2012. Conditioning in First-Order Knowledge Compilation and Lifted Probabilistic Inference. In *AAAI*. AAAI Press.

Van den Broeck, G.; Kersting, K.; Natarajan, S.; and Poole, D., eds. 2021. *An Introduction to Lifted Probabilistic Inference*. Neural Information Processing series. The MIT Press.

Van den Broeck, G.; and Niepert, M. 2015. Lifted Probabilistic Inference for Asymmetric Graphical Models. In *AAAI*, 3599–3605. AAAI Press.

Van den Broeck, G.; Taghipour, N.; Meert, W.; Davis, J.; and De Raedt, L. 2011. Lifted Probabilistic Inference by First-Order Knowledge Compilation. In *IJCAI*, 2178–2185. IJCAI/AAAI.