First-Order and Second-Order Model Counting Meet Stable Marriages, Stable Roommates, and Stable Diners

Václav Kůla
Faculty of Electrical Engineering,
Czech Technical University in Prague
Prague, Czech Republic
kulavacl@fel.cvut.cz

Jan Tóth
Faculty of Electrical Engineering,
Czech Technical University in Prague
Prague, Czech Republic
tothjan2@fel.cvut.cz

Yuanhong Wang
Jilin University
Changchun, China
lucienwang@jlu.edu.cn

Yuyi Wang
CRRC Zhuzhou Institute, China
Zhuzhou, China
yuyiwang920@gmail.com

Ondřej Kuželka
Faculty of Electrical Engineering,
Czech Technical University in Prague
Prague, Czech Republic
ondrej.kuzelka@fel.cvut.cz

ABSTRACT

We study the computational complexity of counting and sampling variants of classical stable matching problems, including the stable marriage, stable roommates, and stable seating arrangement problems. While these problems are generally intractable, we show that many become tractable when the agents' preferences are drawn from only k distinct classes. Our positive results build on recent advances in first-order and second-order model counting from the finite model theory literature. In particular, we demonstrate how these tools both extend recent positive results and resolve an open question about stable seating arrangements posed by Berriaud, Constantinescu, and Wattenhofer (Stable Dinner Party Seating Arrangements, WINE 2023).

KEYWORDS

stable matching; first-order logic; GTEP

ACM Reference Format:
Václav Kůla, Jan Tóth, Yuanhong Wang, Yuyi Wang, and Ondřej Kuželka. 2026. First-Order and Second-Order Model Counting Meet Stable Marriages, Stable Roommates, and Stable Diners. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/SJZA5231

1 INTRODUCTION

In *Downton Abbey*, the seating arrangements at a formal dinner party were as carefully crafted as the outfits worn by the Crawleys. The family understood the significance of these arrangements. Who sat next to whom could make or break alliances, foster new relationships, or reignite old rivalries. A misplaced guest might insult their neighbor, while a clever arrangement could ensure harmony or at least avoid catastrophe. Every attendee's preference was considered because the stakes of a poorly planned seating chart extended beyond awkward silences to the broader social fabric of their world.

Such complexities may seem like the stuff of period dramas, but the challenges of organizing stable seating arrangements are far from fiction. Whether it's aforementioned placing guests at a dinner party (a stable seating problem, which was recently introduced by [6]), matching taxi drivers to passengers (a stable marriage problem), or assigning students to dormitories (stable roommate problem), the underlying principle remains the same: **stability**. Stability ensures that no two individuals (or agents) would prefer to abandon their assigned partners or seats in favor of someone or something else. This deceptively simple idea has profound implications for multi-agent systems and has been a cornerstone of matching theory since Gale and Shapley introduced the stable marriage problem in 1962 [13].

While the decision versions of these problems, whether we can arrange everyone stably, have been extensively studied [13, 16], real-world applications often demand more: **How many stable solutions are there? Can we efficiently sample them to explore the diversity of possibilities?** For instance, in settings where individuals frequently interact, e.g., multi-day training workshops, recurring business dinners, or regular meetings, it is undesirable to repeat seating arrangements, as this limits opportunities for meaningful exchanges. Diverse arrangements encourage broader communication and strengthen relationships over time. In schools, teachers often rotate desk partners to ensure a harmonious classroom environment while also encouraging students to interact with different classmates over time. Similarly, a scriptwriter, say, for the sitcom *The Office*, would undoubtedly prefer not to repeat conference room seatings across episodes. Our framework tackles this problem by offering means to randomly sample valid solutions, which we argue is better suited to finding diverse solutions than multiple runs of solution finding algorithms, which might come with biases.

Despite the importance of counting and sampling, several significant challenges have hindered progress in this area. One major difficulty lies in the computational complexity of these problems. For general preference profiles, counting stable marriages is #P-complete [9, 18], and counting stable roommates is #P-hard [7, 15]. These results imply that there are no polynomial-time algorithms

This work is licensed under a Creative Commons Attribution International 4.0 License.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/SJZA5231

to count these outcomes unless P = #P, making them infeasible for large sets of agents. Another challenge arises from the need to manage preference heterogeneity in real-world scenarios. Recent work has introduced the so-called k-class preference profiles [5], a framework where agents are grouped into k types, with all agents of the same type sharing identical preferences. This approach reduces preference complexity and enables polynomial-time algorithms to solve stable seating arrangements on simple structures such as paths and cycles. However, this framework has not yet been extended to stable marriage problems or stable roommate problems. Besides, it is unclear whether it remains powerful enough for counting or sampling stable configurations. The lack of systematic application of combinatorial frameworks has further limited progress in both counting and sampling tasks. Monadic Second-Order (MSO) Logic with Courcelle's Theorem [10, 11] provides powerful tools for efficient counting on bounded treewidth graphs, yet it has not been adapted to encode the constraints of stable seating arrangements in more complex seat graphs, such as $2 \times n$ grids. Meanwhile, First-Order Model Counting (FOMC) [4], known for its strength in counting relational structures satisfying logical constraints, has yet to be used to count or sample stable configurations. Leveraging its tractable fragments, such as the two-variable fragment with counting quantifiers (C^2) [14, 19], could potentially address both counting and sampling challenges in k-class stable marriages and stable roommates problems but remains largely unexplored.

This paper bridges the tractability gap in stable outcome counting and sampling by integrating k-class preference restrictions with logical counting tools. Our key contributions are as follows:

(1) **Unified k-Class Abstraction for Stable Matching Problems:** We extend the k-class preference framework [5], originally proposed for stable seating, to stable marriages and stable roommates.
(2) **MSO Encoding for Stable Seating on Bounded-Treewidth Graphs:** We show that k-class stable seating arrangements can be encoded as an MSO_2 formula (using edge adjacency and type set variables). By applying Courcelle's counting theorem [10], we prove that counting stable seatings is polynomial-time solvable for any seat graph of bounded treewidth and bounded degree, including $2 \times n$ grids. This result resolves an open problem from [5].
(3) **FOMC Encoding for k-Class Stable Matching Problems:** We model k-class stable marriages and stable roommates using the C^2 fragment of first-order logic (FOL with two variables and counting quantifiers). Since FOMC is polynomial-time tractable for C^2 [19], this provides the first polynomial-time algorithms for counting and sampling stable marriages and stable roommates under k-class preferences.

The results reveal an interesting parameterized complexity landscape for counting and sampling stable outcomes. When k, the number of agent types, is fixed, all three problems are polynomial-time solvable, even as the number of agents, n, increases. Conversely, when k is unbounded, the problems remain NP-hard, consistent with previous findings [5]. This establishes k as a critical parameter for achieving tractability in practical applications.

2 BACKGROUND

In this section, we briefly recall key notions from graph theory and logic used throughout the paper.

2.1 Graph Theory

For a graph $G = (V, E)$, we denote by $|V(G)|$ and $|E(G)|$ the number of vertices and edges, respectively. For $v \in V(G)$, we write $N_G(v)$ for the set of its neighbors (v excluded).

2.1.1 Treewidth and Tree Decompositions. Many results connecting logic and graph algorithms rely on the notion of *treewidth*. A *tree decomposition* of a graph $G = (V, E)$ is a pair $(T, \mathcal{B})$, where T is a tree and $\mathcal{B} = \{B_t \subseteq V \mid t \in V(T)\}$ is a family of subsets of vertices, called *bags*, satisfying the following properties:

(1) for every vertex $v \in V$, there exists a bag B_t such that $v \in B_t$;
(2) for every edge $\{u, v\} \in E$, there exists a bag B_t such that $\{u, v\} \subseteq B_t$;
(3) for each vertex $v \in V$, the set of nodes $\{t \in V(T) \mid v \in B_t\}$ forms a connected subtree of T.

The *width* of a tree decomposition $(T, \mathcal{B})$ is $\max_{t \in V(T)} |B_t| - 1$. The *treewidth* of G, denoted by $\mathrm{tw}(G)$, is the minimum width over all its tree decompositions. Many computationally hard problems become tractable on graphs of bounded treewidth, a fact that underpins Courcelle's theorem and its extensions [3, 10], which we discuss in Section 2.2.3.

2.2 Logic

Here we briefly recall some definitions and tools from first-order and second-order logic which we will need in this paper.

2.2.1 First-Order Logic under Herbrand Semantics. We recall the standard definition of first-order logic (FOL) using Herbrand semantics, which provides a convenient representation for finite relational structures.

Let $\sigma = \{R_1, \ldots, R_\ell\}$ be a finite relational vocabulary where each R_i has fixed arity $r_i \geq 1$. We consider a finite *domain* $D = \{c_1, \ldots, c_n\}$ of distinct constant symbols. The *Herbrand base* over D and σ is the set

$$\mathrm{HB}(D, \sigma) = \{\, R_i(t_1, \ldots, t_{r_i}) \mid R_i \in \sigma,\ t_j \in D \,\},$$

that is, the set of all ground atoms that can be formed from the relation symbols in σ and the constants in D.

A *Herbrand structure* (or *model*) $\mathcal{A}$ over D is then simply a subset of this base, $\mathcal{A} \subseteq \mathrm{HB}(D, \sigma)$, indicating which ground atoms are true. Thus, the space of all possible structures on D corresponds to the power set of the Herbrand base.

Formulas of FOL are built in the usual way from atomic formulas using Boolean connectives and quantifiers $\exists, \forall$. Truth of a formula in a Herbrand structure is defined inductively: an atomic formula $R_i(t_1, \ldots, t_{r_i})$ is true in $\mathcal{A}$ if and only if it belongs to $\mathcal{A}$; Boolean connectives and quantifiers are interpreted in the standard way over the finite domain D. We write $\mathcal{A} \models \varphi$ if the formula φ is true in the structure $\mathcal{A}$ and $\mathcal{A} \not\models \varphi$ if it is not.

Example 2.1. Let $\sigma = \{E\}$ with E a binary relation, and let $D = \{a, b, c\}$. Then $\mathrm{HB}(D, \sigma)$ contains all nine ground atoms $E(a, a)$, $E(a, b)$, $\ldots$, $E(c, c)$. For instance, consider the Herbrand structure

$\mathcal{A} = \{E(a,b), E(b,c)\}$, which represents a directed graph with edges $a \to b$ and $b \to c$. The sentence

$$\varphi = \forall x \, \exists y \, E(x,y)$$

is true in $\mathcal{A}$ if and only if every vertex has at least one outgoing edge, so in this example, it is false because c has none, i.e., in this case $\mathcal{A} \not\models \varphi$.

2.2.2 Monadic Second-Order Logic on Graphs. *Monadic second-order logic* (MSO) extends first-order logic (FOL) by allowing quantification over sets of elements. When interpreted over graphs, the graph structure is fixed, and the only binary relation symbol in the vocabulary is the adjacency relation $E(x,y)$. The domain of interpretation is therefore the vertex set $V(G)$. This is called the *monadic second-order logic on graphs*.

In the MSO_1 fragment, quantification is allowed over individual vertices (this is the classical first-order quantification) and over sets of vertices. In the MSO_2 fragment, quantification extends also to sets of edges, and the logic includes the standard incidence relation $I(v,e)$ connecting vertices and edges. Importantly, such edge sets are interpreted as subsets of the actual edge set $E(G)$; quantification does not range over arbitrary subsets of $V(G) \times V(G)$. No additional predicates or function symbols are included.

The restrictions of MSO_1 and MSO_2 above are essential for Courcelle's theorem, discussed in more detail in Section 2.2.3, which applies to properties definable in MSO over the fixed relational vocabulary of graphs. Intuitively, it ensures that formulas can only refer to the given graph G itself (i.e., its vertices, edges, and their relations) rather than about possible supergraphs on the same vertex set.

An MSO formula $\varphi(\bar{X})$ may contain free set variables $\bar{X} = (X_1, \ldots, X_k)$. Given a graph G and an assignment $\bar{A}$ of subsets of $V(G)$ (and, in the case of MSO_2, subsets of $E(G)$) to these variables, we write $(G, \bar{A}) \models \varphi$ to denote that φ holds in G under this interpretation.

Example 2.2. Consider the MSO_2 formula

$$\varphi(M) \;=\; \forall e_1, e_2 \subseteq E \, \Big(e_1 \in M \wedge e_2 \in M \wedge e_1 \neq e_2 \;\to\; \neg \, \exists v \; (v \in e_1 \wedge v \in e_2) \Big),$$

which states that no two distinct edges in M share a common vertex. Here, M is a free set variable interpreted as a subset of edges. For a given graph G and assignment $A \subseteq E(G)$, the structure (G, A) satisfies $\varphi(M)$ if and only if A forms a *matching* in G. This illustrates how MSO_2 formulas with free set variables can describe combinatorial structures (such as matchings, vertex covers, or independent sets).

2.2.3 Courcelle's Theorem and Its Counting Variant. A central result linking logical definability and graph structure is *Courcelle's theorem* [10]. It states that every graph property definable in MSO_2 can be decided in linear time on graphs of bounded treewidth. Formally, for each fixed MSO_2 sentence φ, there exists a function f such that, given a graph G of treewidth at most t, whether $G \models \varphi$ can be decided in time $f(|\varphi|, t) \cdot |G|$.

An important extension of Courcelle's theorem concerns *counting*. The *counting version of Courcelle's theorem* (see, e.g., [3, 12]) states that for every fixed MSO formula $\varphi(\bar{X})$, the number of satisfying assignments of $\bar{X}$ in G,

$$\#_\varphi(G) = |\{\bar{A} \mid (G, \bar{A}) \models \varphi\}|,$$

can also be computed in time $f(|\varphi|, \mathrm{tw}(G)) \cdot |G|$ for some computable function f. This counting extension has been used to express and efficiently solve various combinatorial and probabilistic problems on structured graph classes.

A further generalization considers the *weighted* version of the counting problem, where vertices or edges of the input graph are assigned numerical weights and the goal is to compute the total weight of all satisfying assignments rather than just their number. Formally, given a weight function $w : V(G) \cup E(G) \to \mathbb{R}$ and an MSO formula $\varphi(\bar{X})$ with free set variables, the *weighted count* is defined as

$$\mathrm{WCount}_{\varphi,w}(G) = \sum_{\bar{A} \subseteq 2^{V(G) \cup E(G)}} \mathbf{1}[(G, \bar{A}) \models \varphi] \cdot w(\bar{A}),$$

where $w(\bar{A})$ denotes the product of weights of all elements selected by $\bar{A}$.

This weighted evaluation problem is also fixed-parameter tractable on graphs of bounded treewidth [2]. The weighted version generalizes the classical counting problem.

2.3 First-Order Model Counting and Tractable Fragments

We will use *first-order model counting* (*FOMC*) as a tool for expressing and analyzing counting tasks arising in this work when Courcelle's theorem cannot be applied. In contrast to Courcelle's theorem, which concerns properties of a *fixed* graph structure with a predefined adjacency relation, FOMC counts over *all* structures on a given finite domain that satisfy a logical specification. This allows FOMC to capture, for example, the number of all undirected graphs without isolated vertices or the number of all k-regular graphs on n vertices—properties that cannot be solved in polynomial time using Courcelle's theorem.

Let σ be a relational vocabulary without function symbols, and let φ be a sentence in first-order logic (FOL), i.e., a formula without free variables. For a finite domain D, a structure $\mathcal{A}$ over D assigns each predicate $P \in \sigma$ a relation $P^{\mathcal{A}} \subseteq D^{\mathrm{arity}(P)}$. In what follows, the domain will always be, w.l.o.g., $[n] = \{1, 2, 3, \ldots, n\}$. The *first-order model count* of φ on a domain of size n is defined as

$$\mathrm{FOMC}(\varphi, n) = \big|\{\, \mathcal{A} \mid D(\mathcal{A}) = [n], \; \mathcal{A} \models \varphi \,\}\big|.$$

Intuitively, FOMC counts the number of finite relational structures on the given domain of size n which satisfy the logical constraints specified by φ.

In general, computing $\mathrm{FOMC}(\varphi, n)$ is intractable [4]. However, several fragments of first-order logic admit polynomial-time algorithms. One of the most important is the *two-variable fragment* FO^2, which restricts formulas to use only two variable symbols (reused via quantification). The seminal works of Van den Broeck [20] and Van den Broeck et al. [21] established that FOMC for FO^2 sentences can be computed in time polynomial in the domain size.

The fragment C^2 extends FO^2 by allowing counting quantifiers of the form $\exists^{=k} x \, \psi(x)$, meaning "there exist exactly k elements x such that $\psi(x)$ holds." The fragment C^2 is strictly more expressive

than FO^2 [14]. Importantly, the tractability of FOMC also carries over from FO^2 to C^2 [19]. In particular, for every C^2 sentence φ, we can compute $\mathrm{FOMC}(\varphi, n)$ in time polynomial in n. We will heavily rely on the tractability of C^2 in this paper.

2.4 Stable Marriage, Stable Roommates, and Stable Seating Arrangements

In this section we briefly recall the classical results on stable matching [13] problems, in particular the stable marriage and stable roommates problems, and their recent generalizations to stable seating arrangements [5]. We also discuss the counting complexity of these classical tasks, which turns out to be less well known.

The *stable marriage* (SM) problem, introduced by Gale and Shapley [13], involves two equally sized sets of agents, each with strict preferences over the members of the opposite set. A matching is *stable* if there is no *blocking pair*, which refers to two agents who both prefer each other to their assigned partners. Gale and Shapley proved that a stable matching always exists and presented the deferred-acceptance algorithm, which computes one in time $O(n^2)$.

The *stable roommates* (SR) problem [16] generalizes SM to the non-bipartite case: $2n$ agents each rank all others, and a stable matching is a partition into n disjoint pairs with no blocking pair, where *blocking pair* is defined in the same way as for the stable marriage problem. In contrast to the stable marriage problem, a stable matching need not exist for the stable roommates problem. Irving [16] gave a quadratic-time algorithm that decides existence and, if successful, produces a stable matching, under the condition that there are no ties in the agents' preferences (if there are ties, the problem becomes NP-hard) [17].

Recently, Bodlaender et al. [6] introduced the *stable seating arrangement* problem, in which agents are assigned to seats along a fixed *seat graph* (for example, a path or a cycle). Each agent has preferences over their possible neighbours, and stability requires that no two agents would both strictly prefer to exchange their seats. Since the unrestricted problem is NP-hard for non-trivial cases [6], Bodlaender et al. [6], Ceylan et al. [8], and Berriaud et al. [5] study the problem under various parametrizations. Berriaud et al. [5] introduce a polynomial-time solvable case, in which the agents come from a bounded number of classes.

2.4.1 Counting stable outcomes. The decision versions of both the stable marriage and stable roommates problems without ties are solvable in polynomial time using the classical algorithms of Gale and Shapley [13] and Irving [16]. However, their counting variants are intractable: counting stable marriages is #P-complete [9, 18], and counting stable roommate matchings is likewise #P-hard, as observed in subsequent papers (see, e.g., [7, 15]).

For the stable seating arrangement problem, the known results primarily concern decision and existence complexity. The work of Berriaud et al. [5] provides polynomial-time algorithms for several restricted settings and NP-hardness for others—we come back to their positive results in Section 3.

2.5 Exchange Stability and Blocking Pairs

The stable matchings, stable roommates, and stable seating arrangements problems all study stability, but two notions are common. Gale and Shapley defined stability via *blocking pairs*, while Alcalde [1] introduced *exchange stability*. Under blocking-pair stability, no two agents prefer each other to their assigned partners and can deviate by forming a new pair (possibly leaving former partners unmatched). If deviations must preserve feasibility—e.g., in the roommates setting with n students and exactly $n/2$ two-person rooms—such a deviation may be inadmissible without moving others. Exchange stability captures a feasibility-preserving deviation: if two agents each prefer the other's partner, they swap (potentially harming their former partners). Blocking pairs are specific to matchings, whereas exchange stability also applies to seating arrangements. Our MSO_2 modeling framework supports both notions, so our results hold in either setting.

3 THE k-CLASS VERSIONS OF STABLE MATCHING AND SEATING PROBLEMS

We now introduce the k-class variants of the classical stable marriage and stable roommates problems, as well as the recently proposed stable seating arrangement problem. These restricted forms were first considered for stable seatings by Berriaud et al. [5]; here we extend the same abstraction to the other settings. The k-class framework bounds the heterogeneity of agents' preferences and serves as the structural parameter for our tractability results.

Definition 3.1 (k-class preference profiles). Let A be a finite set of agents, and let $\mathcal{T} = \{1, \dots, k\}$ denote a finite set of *types*. A *k-class (type-based) preference profile* consists of:

- a type assignment $\tau : A \to \mathcal{T}$, and
- for each type $i \in \mathcal{T}$, a *preference function* $p_i : \mathcal{T} \to \mathbb{R}$.

Given two tuples of types, $S = (s_1, \dots, s_u)$ and $T = (t_1, \dots, t_v)$, with each $s_\ell, t_r \in \mathcal{T}$, we define the preference relation of an agent of type $x \in \mathcal{T}$ over such tuples as:

$$S \succ_x T \quad \Longleftrightarrow \quad \sum_{\ell=1}^{u} p_x(s_\ell) > \sum_{r=1}^{v} p_x(t_r).$$

In the special case where $|S| = |T| = 1$, we write $s_1 \succ_x t_1$ as shorthand. Note that in classical definitions of *stable marriage* and *stable roommates*, agents express preferences by ranking all suitable matches in a strict total order. Our model generalizes this by allowing type-based preferences. This reduces to strict total orders when, for all $i, j, l \in \mathcal{T}$, we have $p_i(j) \neq p_i(l)$.

When $k = n$, we recover the general setting with arbitrary preferences; when $k = 1$, all agents are identical and stability becomes trivial. The k-class restriction thus interpolates between fully homogeneous and fully heterogeneous instances and allows a compact representation: a k-class instance can be described using $O(k^2)$ preference parameters rather than $O(n^2)$.

Berriaud, Constantinescu, and Wattenhofer [5] illustrate the k-class framework by describing several behavioural categories (e.g., charmer, entertainer, diva, politico, introvert, outsider). Each category represents a distinct interaction pattern of preferences over members of other categories.

Stable marriage. In the stable marriage problem, two disjoint sets M and W of n agents each have preferences over the opposite side. A matching $\mu : M \to W$ is *stable* if there is no blocking pair (m, w) such that m and w both strictly prefer each other to their partners

under μ. Gale and Shapley [13] proved that a stable matching always exists and can be found in $O(n^2)$ time via the deferred-acceptance algorithm. In the k-class version, there are again at most k types of men and l-types of women. Men from the same type and women from the same type have the same preferences, respectively, and the preferences do not distinguish men or women from the same type, following Definition 3.1 of k-class preference profiles.

Stable roommates. The stable roommates problem [16] removes the bipartition: a single set of $2n$ agents, each ranking all others, must be partitioned into n pairs with no blocking pair. Unlike in the stable marriage problem, a stable matching need not exist, but Irving [16] showed that existence can be decided and a stable matching constructed, assuming there are no ties, if one exists, in polynomial time. The k-class restriction again limits the number of distinct preference types to at most k. It does bring ties but as we will see, the k-class restriction will make not only the decision and construction problems but also the respective counting and sampling problems solvable in polynomial-time.

Stable seating arrangements. The stable seating arrangement problem [5] generalizes stability to agents placed on a fixed *seat graph* $G = (S, E)$—Berrieaud et al. studied path and cycle structures. Each agent has preferences over others as potential neighbors. A seating $\sigma : A \to S$ is *stable* if there is no pair of agents who both strictly prefer to swap seats. Berriaud et al. showed that the problem is polynomial-time solvable on paths and cycles when restricted to the k-class variant and NP-hard otherwise. As in the other k-class problems, following Definition 3.1, the k-class version assumes at most k distinct preference functions among all agents with agents from the same class being indistinguishable in the other agents' preferences.

The k-class framework provides a uniform abstraction across these three problems. It preserves the essential combinatorial structure of stability while bounding preference diversity and input size. In the following sections, we exploit this parameterization to establish tractable results for counting and sampling stable outcomes using the counting version of Courcelle's theorem and tools from first-order model counting.

3.1 Known Results for the k-Class Stable Seating Arrangement Problem

The k-class variant of the stable seating arrangement problem was introduced by Berriaud, Constantinescu, and Wattenhofer [5]. In this model, agents are seated on a fixed *seat graph* $G = (S, E)$, and each agent belongs to one of at most k preference classes, with all agents of the same class sharing the same preference function over potential neighbors. A seating $\sigma : A \to S$ is stable if no two agents would both strictly prefer to swap seats.

Berriaud et al. presented a comprehensive analysis of the computational complexity of finding stable seatings under various assumptions on G and k. Their main findings can be summarized as follows:

- For a fixed number of classes k, the existence of a stable seating arrangement can be decided in polynomial time when the seat graph is a *path* or cycle. The corresponding algorithms rely on dynamic programming over the table structure and run in time polynomial in the number of agents and exponential only in k.
- When k is unbounded (i.e., part of the input), the problem becomes NP-hard even on very simple seat graphs such as paths or cycles.
- Even though not stated explicitly in their paper, for fixed k, enumeration and counting of all stable seatings are tractable on path and cycle tables, since stability constraints can be encoded locally and their dynamic programming approach can be extended to count feasible arrangements (even though it has not been done in their paper).

One of open questions listed by Berriaud et al. in their paper concerns the case of tables where the graph corresponds to a $2 \times n$ grid where agents are seated at a rectangular grid and they have preferences not only on their left and right neighbours but also on the agent sitting across the table from them. The authors leave as an open problem whether the existence of a stable seating arrangement on such a table remains polynomial-time decidable for fixed k, or whether it becomes NP-hard even when k is bounded.

4 AN MSO FORMULATION FOR STABLE SEATING ARRANGEMENTS ON BOUNDED-TREEWIDTH GRAPHS

We now show that the k-class variant of the stable seating arrangement problem can be uniformly expressed in monadic second-order logic of graphs (MSO_2) using only the edge relation $E(x, y)$. This allows the direct application of Courcelle's theorem and its counting version to obtain polynomial-time algorithms for decision, construction, and counting. This approach will among others resolve the open case of $2 \times n$ grid tables mentioned in the previous section.

Graph representation. Each instance is represented by a graph $G = (V, E)$, where vertices correspond to seats and edges correspond to adjacency in the seat graph. No additional binary relations are introduced; all other problem-specific information will be encoded using unary predicates over the vertices.

Set variables and preferences. Because agents are partitioned into at most k preference classes, we introduce vertex set-variables

$$X_1, X_2, \ldots, X_k$$

and constrain them to be disjoint

$$\bigwedge_{i \neq j} \forall x \, (x \in X_i \to \neg x \in X_j)$$

indicating the type of vertex x, i.e. $x \in X_i$ if and only if the agent seated at the node x is from the type i.[1] Each type corresponds to one fixed preference function over the set of possible neighbors given by the k-class preference profiles (Definition 3.1). These preferences will be encoded implicitly within the MSO_2 formula by using constant-size subformulas that refer to these type set variables.

As an example, we show how to encode preferences in the setting where each agent has exactly two neighbours, which corresponds

[1] $x \in X_i$ can be thought of as a unary predicate $P_i(x)$ that holds if and only if x is of type i.

to a circular table graph (and may also represent several disjoint circular tables):

$$\text{PrefersNeighbours}(x_{agent}, x_{left}, x_{right}, x'_{left}, x'_{right}) \equiv \bigvee_{(i,j,i',j'):(i,j)\succ_k(i',j')} (x_{agent} \in X_k \wedge x_{left} \in X_i \wedge x_{right} \in X_j \wedge x'_{left} \in X_{i'} \wedge x'_{right} \in X_{j'}).$$

Here, the notation $(i, j) \succ_k (i', j')$ means that agents from class k prefer having a neighbour from class i on the left and a neighbour from class j on the right over having a neighbour from class i' on the left and a neighbour from class j' on the right.[2]

Now we can use the above formula to define when an agent seated at x envies another agent seated at x'. We still assume the case where every agent has exactly two neighbours. Here, we need to be careful and consider two cases: (i) when x and x' are neighbouring seats and (ii) when they are not. For the second, easier, case we can write:

$$\begin{aligned}\text{Envies}_{(\text{ii})}(x, x') \equiv (\neg E(x, x') \wedge \neg E(x', x)) \wedge \\ \Big(\forall x_{left} \forall x_{right} \forall x'_{left} \forall x'_{right} \ \big(E(x_{left}, x) \wedge \\ E(x, x_{right}) \wedge E(x'_{left}, x') \wedge E(x', x'_{right})\big) \rightarrow \\ \text{PrefersNeighbours}(x, x'_{left}, x'_{right}, x_{left}, x_{right})\Big),\end{aligned}$$

where all universal quantifications are over vertices of the graph (we will also omit the domain of quantification in the following for brevity). This formula says precisely that the agent seated at x envies the agent seated at x' if the former would prefer the latter's neighbours over its own.

Next we tackle the case when x and x' are actually neighbouring seats in the graph. In this case we must be careful because when we swap the two agents' positions, they will not end up with each others' neighbours because one of them is the other agent being swapped. The resulting formula for this case is:

$$\begin{aligned}&\text{Envies}_{(\text{i})}(x, x') \equiv \\ &\Big(E(x, x') \wedge \Big(\forall x_{left} \forall x'_{right} \ \big(E(x_{left}, x) \wedge E(x', x'_{right})\big) \rightarrow \\ &\qquad \text{PrefersNeighbours}(x, x', x'_{right}, x_{left}, x')\Big)\Big) \vee \\ &\Big(E(x', x) \wedge \Big(\forall x'_{left} \forall x_{right} \ \big(E(x'_{left}, x') \wedge E(x, x_{right})\big) \rightarrow \\ &\qquad \text{PrefersNeighbours}(x, x_{left}, x', x', x_{right})\Big)\Big)\end{aligned}$$

Then we can use the above formulas to define a new formula for detecting if agents seated at x and at x' make a blocking pair (breaking stability of the arrangement):

$$\begin{aligned}\text{Unstable}(x, x') \equiv (\text{Envies}_{(\text{i})}(x, x') \vee \text{Envies}_{(\text{ii})}(x, x')) \wedge \\ (\text{Envies}_{(\text{i})}(x', x) \vee \text{Envies}_{(\text{ii})}(x', x)).\end{aligned}$$

[2]The same encoding can be adapted to any fixed bounded degree by introducing variables for all incident neighbours (and their alternatives) and taking the disjunction over the corresponding preference comparisons. If the seating graph has maximum degree at most Δ, one can also handle varying degrees by taking a disjunction over the possible local degrees (with the appropriate adjacency-pattern test as a guard) and enforcing only the corresponding preference constraint.

$\text{Unstable}(x, x')$ holds if and only if the agents seated at x and at x' would both prefer to swap.

Finally, we can write an MSO formula with free variables $X_1, X_2, \ldots, X_k$, that is satisfied if and only if the arrangement of agents to seats represented by the X_i sets is stable:

$$\text{StableArrangement}(X_1, X_2, \ldots, X_k) \equiv \forall x \forall x' \neg \text{Unstable}(x, x')$$

Note that $\text{Unstable}(x, x')$ is just a formula (not a new relation) that uses the set variables $X_1, X_2, \ldots, X_k$.

Finishing touches. We are almost done. The formula

$$\text{StableArrangement}(X_1, X_2, \ldots, X_k)$$

is an MSO formula and the table graphs are supposed to be of bounded treewidth and bounded degree. That means we can apply Courcelle's theorem and its counting extension directly. However, as we said above, we are almost done, but not quite. What we got so far is a polynomial-time algorithm (polynomial in the number of agents) for computing something slightly different than what we need. To see that, first, note that we have not specified the preference types of the agents at all. Moreover, we are not yet counting the arrangements of the agents but instead ways to label the seats by the agent types while maintaining stability.

Let us first address the former issue. Since the agents in each preference type T_i are indistinguishable, we will not specify the types of each of them, instead we will just constrain the number of agents in each class based on the input specification of the problem. How can we do that? The most straightforward way is as follows. We use the fact that, as we discussed in Section 2.2.3, Courcelle's theorem extends to weighted counting. We introduce a weight w_i for each of the set variables X_i. As a result, the weighted count

$$\text{WCount}_{\text{StableArrangement},w}(G)$$

is a polynomial in the variables $w_1, w_2, \ldots, w_k$. To get the "number" that corresponds to having exactly n_1 agents of type 1, exactly n_2 agents of type 2 and so on, we need to extract the coefficient of the monomial $w_1^{n_1} w_2^{n_2} \cdots w_k^{n_k}$, which can be done using multivariate polynomial interpolation (in polynomial time). Hence, we have solved the first issue—we can constrain the number of agents from each preference type.

Fixing the second issue, is then also simple: To get the number of stable seating arrangements of the agents, we multiply the result obtained so far by $n_1!n_2! \cdots n_k!$, exploiting the indistinguishability of the agents in each preference type (which means that we can permute them within each type).

Sampling. Finally, to also support sampling of stable seating arrangements, we notice that Courcelle's theorem and the corresponding tractability hold even if we add "unary evidence", which will make the strategy described next work.

We introduce k unary predicates

$$\text{Selected}_1(x), \text{Selected}_2(x), \ldots, \text{Selected}_k(x)$$

which will be used to denote the preference types of the agents already selected for the places x. Then we add constraints to connect these unary predicates with the set variables $X_1, X_2, \ldots, X_k$:

$$\bigwedge_{i=1}^{k} \forall x \ \text{Selected}_i(x) \rightarrow x \in X_i.$$

We then iterate over the seats x and sample the preference types of the agents seated in them one by one—each time we sample a preference type i of an agent A, we add the constraint $\text{Selected}_i(A)$, we use the fact that we can count the number of the stable seating arrangements with given evidence using Courcelle's theorem to compute the probability with which the next seat will be occupied by the agent with a given preference profile. After we are done with sampling the preference types of the agents in the individual seats, we permute the agents in each class uniformly and assign to the vertices. It is not difficult to see that this procedure produces uniform stable seating arrangements.

5 FOMC FORMULATION FOR STABLE MARRIAGES AND STABLE ROOMMATES

We now turn to the counting and sampling variants of the k-class stable marriage and stable roommates problems. Whereas our treatment of stable seating arrangements crucially exploits the bounded-treewidth structure of the seating graph, for SM and SR it is convenient to work directly with the (typically complete) preference relations in a logical formalism. In particular, we express these problems as instances of weighted first-order model counting (WFOMC) in the two-variable fragment with counting quantifiers (C^2). A further benefit of the WFOMC viewpoint is its modularity: it extends easily to settings with several (possibly interacting) matchings, e.g., simultaneously assigning roommate pairs and project-partner pairs subject to additional cross-constraints. Polynomial-time solvability for WFOMC in C^2 is known [19].

5.1 Encoding the Stable Marriage Problem in C^2

Domain and predicates. The domain D consists of the set of agents A. For the stable marriage problem, we introduce unary predicates $L(x)$ and $R(x)$ distinguishing the two sides. Agents are divided into k preference types represented by unary predicates $T_1(x), \ldots, T_k(x)$. We use a binary predicate $M(x, y)$ to represent the matching relation denoting that x, a member of the side L (e.g., corresponding to men in the classical formulation of the problem), is matched to y, a member of the other side R.

Matching constraints. To ensure that $M(x, y)$ represents a one-to-one matching, we express the following formula without free variables (i.e., a sentence):

$$\begin{aligned}\text{Matching} \equiv \big(\forall x\, L(x) \rightarrow (\exists^{=1} y\, M(x,y))\big) \wedge \\ \big(\forall y\, R(y) \rightarrow (\exists^{=1} x\, M(x,y))\big) \wedge \\ (\forall x \forall y\, (M(x,y) \rightarrow (L(x) \wedge R(y))))\,.\end{aligned}$$

The first two conjuncts assert that each agent has exactly one partner; and the third enforces that the matching relation should "point" from the side L to the side R (which is specified using unary evidence). All three statements use two variables with counting quantifiers, staying within C^2.

Encoding the Agent Types and Sides. Each agent has its type, which determines among others its preferences, specified as part of the input to the problem. In the encoding, we specify the agent types using the unary predicates T_i, where $T_i(A)$ means that the agent A has type preference type i. As shown in [22], FOMC remains tractable for C^2 sentences even if we add *unary evidence*, where *unary evidence* refers to a conjunction of ground unary literals. This means that we can specify the agent types of all agents without affecting the tractability of the model counting problem—as long as we can represent the rest of the constraints using just two logical variables (which we will show we can). Moreover, we can use the same idea to encode the information about the side (e.g., men or women) to which the agents belong. We denote the conjunction of all this evidence as UnaryEvidence.

Stability constraint. Next we encode the stability constraints. What makes this more challenging than in the previous section, where we dealt with stable seating arrangements and where we could use Courcelle's theorem, is that we can only use two logical variables to do that in order to stay within the C^2 fragment. To aid the encoding, we introduce k auxiliary unary predicates

$$\text{MatchedTo}_1(x), \text{MatchedTo}_2(x), \ldots, \text{MatchedTo}_k(x)$$

which will be used to indicate what is the type of the agent matched to agent x (recall that there is exactly one such agent). In particular, we add to the encoding the sentence:

$$\text{MatchedTo} \equiv \bigwedge_{i=1}^{k} (\forall x\ (\text{MatchedTo}_i(x) \Leftrightarrow (\exists y\ ((M(x,y) \vee M(y,x)) \wedge T_i(y))))\,.$$

Now we have all the ingredients to encode the stability constraints. What remains is to show that there are no blocking pairs—there is a blocking pair if there exists a pair of agents who prefer each other to their matches. For two agents x and y where x is of type T_i and y is of type T_j, we can write a formula which is true if and only if x and y are a blocking pair as:

$$\text{BlockingPair}_{i,j}(x,y) \equiv \bigvee_{(k,l):(j \succ_i k \wedge i \succ_j l)} \big(\text{MatchedTo}_k(x) \wedge \text{MatchedTo}_l(y)\big),$$

where we recall that the notation $j \succ_i k$ denotes that i prefers j over k.

Finally, we can use the above to write the full condition for the stability of the matching:

$$\text{Stable} \equiv \neg\big(\exists x \exists y : \bigvee_{i,j \in [k]} \big(T_i(x) \wedge T_j(y) \wedge \text{BlockingPair}_{i,j}(x,y)\big)\big)$$

To count the number of solutions to the stable marriage problem, we need to compute

$$\text{FOMC}(\text{Matching} \wedge \text{MatchedTo} \wedge \text{Stable} \wedge \text{UnaryEvidence}, n),$$

where n is the number of agents; and we recall that UnaryEvidence contains information about the preference types and sides to which the individual agents belong.

Tractability. For each fixed k, the above FOMC problem is on a formula of constant size (dependent on the number of type combinations but not on the number of agents). By the result of [19], WFOMC for C^2 sentences can be computed in time polynomial in n. Consequently, the number of stable matchings in any k-class instance can be computed in polynomial time.

5.2 Modification for the Stable Roommates Problem

The same FOMC approach can be applied to the k-class *stable roommates* problem with only minor modifications to the logical encoding.

Eliminating the bipartite structure. In the stable roommates problem, there is only one set of agents rather than two disjoint sides. Accordingly, we remove the unary predicates $L(x)$ and $R(x)$ and the constraint enforcing bipartiteness in the matching relation. All agents are drawn from the same domain A, and the type predicates $T_1(x), \ldots, T_k(x)$ are retained as before.

Symmetric matching relation. The matching predicate $M(x, y)$ now denotes an *undirected* edge, meaning that x and y are matched to each other. We enforce this by adding a symmetry constraint:

$$\forall x \forall y \ \big(M(x, y) \leftrightarrow M(y, x)\big). \tag{1}$$

Each agent must still have exactly one partner, as before:

$$\forall y \, \exists^{=1} x \, M(x, y) \tag{2}$$

These two constraints together ensure that M is a perfect matching in the roommate setting.

Stability constraints. The stability condition is defined analogously to the bipartite case. Because the matching relation is symmetric, the auxiliary unary predicates

$$\text{MatchedTo}_1(x), \ldots, \text{MatchedTo}_k(x)$$

are defined using the same schema:

$$\bigwedge_{i \in [k]} \forall x \ \big(\text{MatchedTo}_i(x) \Leftrightarrow \exists y \, (M(x, y) \wedge T_i(y))\big). \tag{3}$$

The blocking-pair definition from the previous subsection remains unchanged, since it only depends on the types of the two agents and on their partners' types. Consequently, the stability formula

$$\text{Stable} \equiv \neg\big(\exists x \exists y : \bigvee_{i,j \in [k]} \big(T_i(x) \wedge T_j(y) \wedge \text{BlockingPair}_{i,j}(x, y)\big)\big)$$

applies verbatim to the roommates case as well.

Counting stable roommate matchings. The number of stable roommate matchings can thus be expressed as

$$\text{FOMC}\big(\text{Matching}_{\text{SR}} \wedge \text{Stable} \wedge \text{UnaryEvidence}, n\big),$$

where $\text{Matching}_{\text{SR}}$ denotes the conjunction of the one-to-one (2), symmetry (1) constraints, and the matched-to definitions (3). Because the resulting formula again lies within the C^2 fragment, the same tractability argument applies: WFOMC can be computed in polynomial time for each fixed k.

5.3 Sampling Stable Matchings

Beyond counting, sampling stable matchings is of considerable practical interest. In applications such as resource allocation, course assignments, or pairing mechanisms, it may be undesirable to always output a particular stable matching produced by a deterministic algorithm, since different algorithms can systematically favour certain agents or types of outcomes. Sampling a stable matching uniformly at random—or according to a specified probabilistic model—offers a fairer and often more interpretable alternative.

Recent progress in lifted probabilistic inference has shown that, for certain fragments of first-order logic, sampling satisfying structures can be performed efficiently without grounding. In particular, Wang, Pu, Wang, and Kuželka [22] establish that for any fixed sentence in the fragment C^2, models can be sampled in time polynomial in the size of the domain. Their method exploits the same symmetries that make weighted first-order model counting tractable in this fragment, but does so directly—without reducing sampling to counting. This direct approach is necessary because, unlike in the propositional setting, no general correspondence between counting and sampling is known to hold for lifted (first-order) representations.

The formulas we constructed for the k-class stable marriage and stable roommates problems are all C^2 sentences. Consequently, the lifted sampling results of [22] apply directly. Given the evidence specifying the agents' types (and sides, in the bipartite case), we can therefore sample a stable matching approximately uniformly at random in polynomial time, exploiting the same lifted representation used for counting. Moreover, in case we want to force diversity from already discovered scenarios manually, we can use *cardinality constraints* [19] to limit the setting further.

5.4 Multiple Matchings

Consider a scenario where students are seeking not only a roommate but also a partner for extracurricular activities. In this model, the Activity Matching directly influences the Roommate Matching. For instance, a student matched to a 'Late Night Gaming' friend may be incompatible with a student matched to an 'Early Morning Rowing' crew member, resulting in a veto constraint, while each student has preference over both their roommate and activity match. Modeling this interdependent, multi-layered problem is also possible in a straightforward manner in the k-class setting using our proposed FOMC framework.

6 CONCLUSIONS

We established a unifying framework for reasoning about the counting and sampling complexity of classical stable matching problems through the lens of logical model counting. Our results show that the combination of k-class preference structures and tractable logical fragments—MSO_2 on bounded-treewidth graphs and C^2 for dense structures—yields a principled route to polynomial-time algorithms for several variants previously thought intractable.

On the structural side, we resolved an open problem from [5] by showing that counting stable seating arrangements remains tractable on any bounded-treewidth seating graph, including the previously unresolved case of rectangular tables. On the logical side, we demonstrated that both the k-class stable marriage and k-class stable roommates problems can be formulated as first-order model counting instances in C^2, implying polynomial-time solvability of their counting and sampling variants.

Beyond their theoretical interest, these results open new directions for the analysis of matching markets and agent-based systems. Counting and sampling stable outcomes provide a quantitative view of stability—how constrained a market or social configuration truly is—while lifted model counting and sampling offer scalable computational tools to explore this space systematically.

ACKNOWLEDGMENTS

Václav Kůla is supported by the Central Europe Leuven Strategic Alliance (CELSA) project *Towards Scalable Algorithms for Neuro-Symbolic AI*. Yuanhong Wang is supported by National Natural Science Foundation of China (No.62506141). Ondřej Kuželka is supported by the Czech Science Foundation project 24-11820S (*Automatic Combinatorialist*).

REFERENCES

[1] José Alcalde. 1994. Exchange-proofness or divorce-proofness? Stability in one-sided matching markets. *Review of Economic Design* 1 (02 1994), 275–287. https://doi.org/10.1007/BF02716626

[2] Antoine Amarilli, Pierre Bourhis, and Pierre Senellart. 2015. Provenance circuits for trees and treelike instances. In *International Colloquium on Automata, Languages, and Programming*. Springer, 56–68.

[3] Stefan Arnborg, Jens Lagergren, and Detlef Seese. 1991. Easy Problems for Tree-Decomposable Graphs. *Journal of Algorithms* 12, 2 (1991), 308–340. https://doi.org/10.1016/0196-6774(91)90006-K

[4] Paul Beame, Guy Van den Broeck, Eric Gribkoff, and Dan Suciu. 2015. Symmetric weighted first-order model counting. In *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 313–328.

[5] Damien Berriaud, Andrei Constantinescu, and Roger Wattenhofer. 2023. Stable Dinner Party Seating Arrangements. In *Proceedings of the 19th International Conference on Web and Internet Economics (WINE 2023) (Lecture Notes in Computer Science)*. Springer. Extended version available as arXiv:2310.03494.

[6] Hans L. Bodlaender, Tesshu Hanaka, Lars Jaffke, Hirotaka Ono, Yota Otachi, and Tom C. van der Zanden. 2020. *Hedonic Seat Arrangement Problems*. WorkingPaper. arXiv. https://doi.org/10.48550/arXiv.2002.10898

[7] Robert Bredereck, Klaus Heeger, Dušan Knop, and Rolf Niedermeier. 2022. Parameterized complexity of stable roommates with ties and incomplete lists through the lens of graph parameters. *Information and Computation* 289 (2022), 104943.

[8] Esra Ceylan, Jiehua Chen, and Sanjukta Roy. 2023. Optimal Seat Arrangement: What Are the Hard and Easy Cases? 2563–2571. https://doi.org/10.24963/ijcai.2023/285

[9] Pavithra Chebolu, Leslie Ann Goldberg, and Russell Martin. 2012. The complexity of approximately counting stable marriages. *Theoretical Computer Science* 437 (2012), 35–47. https://doi.org/10.1016/j.tcs.2012.02.029

[10] Bruno Courcelle. 1990. The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs. *Information and Computation* 85, 1 (1990), 12–75. https://doi.org/10.1016/0890-5401(90)90043-H

[11] Bruno Courcelle and Mohamed Mosbah. 1993. Monadic second-order evaluations on tree-decomposable graphs. *Theoretical Computer Science* 109, 1-2 (1993), 49–82.

[12] Jörg Flum and Martin Grohe. 2004. The Parameterized Complexity of Counting Problems. *SIAM J. Comput.* 33, 4 (2004), 892–922. https://doi.org/10.1137/S0097539703427203

[13] David Gale and Lloyd S. Shapley. 1962. College Admissions and the Stability of Marriage. *Amer. Math. Monthly* 69, 1 (1962), 9–15. https://doi.org/10.2307/2312726

[14] Erich Grädel, Martin Otto, and Eric Rosen. 1997. Two-Variable Logic with Counting is Decidable. In *Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science (LICS 1997)*. 306–317. https://doi.org/10.1109/LICS.1997.614955

[15] Dan Gusfield and Robert W. Irving. 1989. *The Stable Marriage Problem: Structure and Algorithms*. MIT Press, Cambridge, MA.

[16] Robert W. Irving. 1985. An Efficient Algorithm for the "Stable Roommates" Problem. *Journal of Algorithms* 6, 4 (1985), 577–595. https://doi.org/10.1016/0196-6774(85)90033-1

[17] Robert W. Irving. 1994. Stable marriage and indifference. *Discrete Applied Mathematics* 48, 3 (1994), 261–272. https://doi.org/10.1016/0166-218X(92)00179-P

[18] Robert W. Irving and Peter Leather. 1986. The Complexity of Counting Stable Marriages. *SIAM J. Comput.* 15, 3 (1986), 655–667. https://doi.org/10.1137/0215045

[19] Ondrej Kuzelka. 2021. Weighted first-order model counting in the two-variable fragment with counting quantifiers. *Journal of Artificial Intelligence Research* 70 (2021), 1281–1307.

[20] Guy Van den Broeck. 2011. On the completeness of first-order knowledge compilation for lifted probabilistic inference. *Advances in Neural Information Processing Systems* 24 (2011).

[21] Guy Van den Broeck, Wannes Meert, and Adnan Darwiche. 2014. Skolemization for weighted first-order model counting. In *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR)*. ASSOC ADVANCEMENT ARTIFICIAL INTELLIGENCE, 1–10.

[22] Yuanhong Wang, Juhua Pu, Yuyi Wang, and Ondřej Kuželka. 2024. Lifted algorithms for symmetric weighted first-order model sampling. *Artificial Intelligence* 331 (2024), 104114.