New Liftable Classes for First-Order Probabilistic Inference[*]

Seyed Mehran Kazemi
The University of British Columbia
smkazemi@cs.ubc.ca

Angelika Kimmig
KU Leuven
angelika.kimmig@cs.kuleuven.be

Guy Van den Broeck
University of California, Los Angeles
guyvdb@cs.ucla.edu

David Poole
The University of British Columbia
poole@cs.ubc.ca

Abstract

Statistical relational models provide compact encodings of probabilistic dependencies in relational domains, but result in highly intractable graphical models. The goal of lifted inference is to carry out probabilistic inference without needing to reason about each individual separately, by instead treating exchangeable, undistinguished objects as a whole. In this paper, we study the *domain recursion* inference rule, which, despite its central role in early theoretical results on domain-lifted inference, has later been believed redundant. We show that this rule is more powerful than expected, and in fact significantly extends the range of models for which lifted inference runs in time polynomial in the number of individuals in the domain. This includes an open problem called S4, the symmetric transitivity model, and a first-order logic encoding of the birthday paradox. We further identify new classes S^2FO^2 and S^2RU of domain-liftable theories, which respectively subsume FO^2 and *recursively unary theories*, the largest classes of domain-liftable theories known so far, and show that using domain recursion can achieve exponential speedup even in theories that cannot fully be lifted with the existing set of inference rules.

1 Introduction

Statistical relational learning (SRL) [8] aims at unifying logic and probability for reasoning and learning in noisy domains, described in terms of individuals (or objects), and the relationships between them. Statistical relational models [10], or template-based models [18] extend Bayesian and Markov networks with individuals and relations, and compactly describe probabilistic dependencies among them. These models encode exchangeability among the objects: individuals that we have the same information about are treated similarly.

A key challenge with SRL models is the fact that they represent highly intractable, densely connected graphical models, typically with millions of random variables. The aim of *lifted inference* [23] is to carry out probabilistic inference without needing to reason about each individual separately, by instead treating exchangeable, undistinguished objects as a whole. Over the past decade, a large number of lifted inference rules have been proposed [5, 9, 11, 14, 20, 22, 28, 30], often providing exponential speedups for specific SRL models. These basic exact inference techniques have applications in (tractable) lifted learning [32], where the main task is to efficiently compute *partition functions*, and in variational and over-symmetric approximations [29, 33]. Moreover, they provided the foundation for a rich literature on approximate lifted inference and learning [1, 4, 13, 17, 19, 21, 25, 34].

[*] This is an extended version of a NIPS-2016 paper including all the proofs.

The theoretical study of lifted inference began with the complexity notion of *domain-lifted* inference [31] (a concept similar to data complexity in databases). Inference is domain-lifted when it runs in time polynomial in the number of individuals in the domain. By identifying *liftable classes* of models, guaranteeing domain-lifted inference, one can characterize the theoretical power of the various inference rules. For example, the class FO^2, encoding dependencies among pairs of individuals (i.e., two logical variables), is liftable [30]. Kazemi and Poole [15] introduce a liftable class called *recursively unary*, capturing hierarchical simplification rules. Beame *et al.* [3] identify liftable classes of probabilistic database queries. Such results elevate the specific inference rules and examples to a general principle, and bring lifted inference in line with complexity and database theory [3].

This paper studies the *domain recursion* inference rule, which applies the principle of induction on the domain size. The rule makes one individual A in the domain explicit. Afterwards, the other inference rules simplify the SRL model up to the point where it becomes identical to the original model, except the domain size has decreased. Domain recursion was introduced by Van den Broeck [31] and was central to the proof that FO^2 is liftable. However, later work showed that simpler rules suffice to capture FO^2 [27], and the domain recursion rule was forgotten.

We show that domain recursion is more powerful than expected, and can lift models that are otherwise not amenable to domain-lifted inference. This includes an open problem by Beame *et al.* [3], asking for an inference rule for a logical sentence called S4. It also includes the symmetric transitivity model, and an encoding of the birthday paradox in first-order logic. There previously did not exist any efficient algorithm to compute the partition function of these SRL models, and we obtain exponential speedups. Next, we prove that domain recursion supports its own large *classes of liftable models* S^2FO^2 subsuming FO^2, and S^2RU subsuming recursive unary[2]. All existing exact lifted inference algorithms (e.g., [11, 15, 28]) resort to grounding the theories in S^2FO^2 or S^2RU that are not in FO^2 or recursively unary, and require time exponential in the domain size.

These results will be established using the weighted first-order model counting (WFOMC) formulation of SRL models [28]. WFOMC is close to classical first-order logic, and it can encode many other SRL models, including Markov logic [24], parfactor graphs [23], some probabilistic programs [7], relational Bayesian networks [12], and probabilistic databases [26]. It is a basic specification language that simplifies the development of lifted inference algorithms [3, 11, 28].

2 Background and Notation

A **population** is a set of constants denoting individuals (or objects). A **logical variable (LV)** is typed with a population. We represent LVs with lower-case letters, constants with upper-case letters, the population associated with a LV x with Δ_x, and its cardinality with $|\Delta_x|$. That is, a population Δ_x is a set of constants $\{X_1, \dots, X_n\}$, and we use $x \in \Delta_x$ as a shorthand for instantiating x with one of the X_i. A **parametrized random variable (PRV)** is of the form $\mathsf{F}(t_1, \dots, t_k)$ where F is a predicate symbol and each t_i is a LV or a constant. A **unary** PRV contains exactly one LV and a **binary** PRV contains exactly two LVs. A **grounding** of a PRV is obtained by replacing each of its LVs x by one of the individuals in Δ_x.

A **literal** is a PRV or its negation. A **formula** φ is a literal, a disjunction $\varphi_1 \vee \varphi_2$ of formulas, a conjunction $\varphi_1 \wedge \varphi_2$ of formulas, or a quantified formula $\forall x \in \Delta_x : \varphi(x)$ or $\exists x \in \Delta_x : \varphi(x)$ where x appears in $\varphi(x)$. A **sentence** is a formula with all LVs quantified. A **clause** is a disjunction of literals. A **theory** is a set of sentences. A theory is **clausal** if all its sentences are clauses. An **interpretation** is an assignment of values to all ground PRVs in a theory. An interpretation I is a **model** of a theory T, $I \models T$, if given its value assignments, all sentences in T evaluate to True.

Let $\mathcal{F}(T)$ be the set of predicate symbols in theory T, and $\Phi : \mathcal{F}(T) \rightarrow \mathbb{R}$ and $\overline{\Phi} : \mathcal{F}(T) \rightarrow \mathbb{R}$ be two functions that map each predicate F to weights. These functions associate a weight with assigning True or False to ground PRVs $\mathsf{F}(C_1, \dots, C_k)$. For an interpretation I of T, let ψ^{True} be the set of ground PRVs assigned True, and ψ^{False} the ones assigned False. The weight of I is given by $\omega(I) = \prod_{\mathsf{F}(C_1,\dots,C_k) \in \psi^{True}} \Phi(\mathsf{F}) \cdot \prod_{\mathsf{F}(C_1,\dots,C_k) \in \psi^{False}} \overline{\Phi}(\mathsf{F})$. Given a theory T and two functions Φ and $\overline{\Phi}$, the **weighted first-order model count (WFOMC)** of the theory given Φ and $\overline{\Phi}$ is: $\mathrm{WFOMC}(T|\Phi, \overline{\Phi}) = \sum_{I \models T} \omega(I)$.

[2] All proofs can be found in the extended version of the paper at: https://arxiv.org/abs/1610.08445

In this paper, we assume that all theories are clausal and do not contain existential quantifiers. The latter can be achieved using the Skolemization procedure of Van den Broeck *et al.* [30], which efficiently transforms a theory T with existential quantifiers into a theory T' without existential quantifiers that has the same weighted model count. That is, our theories are sets of finite-domain, function-free first-order clauses whose LVs are all universally quantified (and typed with a population). Furthermore, when a clause mentions two LVs x_1 and x_2 with the same population Δ_x, or a LV x with population Δ_x and a constant $C \in \Delta_x$, we assume they refer to different individuals.[3]

Example 1. Consider the theory $\forall x \in \Delta_x : \neg\mathsf{Smokes}(x) \vee \mathsf{Cancer}(x)$ having only one clause and assume $\Delta_x = \{A, B\}$. The assignment $\mathsf{Smokes}(A) = \mathsf{True}, \mathsf{Smokes}(B) = \mathsf{False}, \mathsf{Cancer}(A) = \mathsf{True}, \mathsf{Cancer}(B) = \mathsf{True}$ is a model. Assuming $\Phi(\mathsf{Smokes}) = 0.2$, $\Phi(\mathsf{Cancer}) = 0.8$, $\overline{\Phi}(\mathsf{Smokes}) = 0.5$ and $\overline{\Phi}(\mathsf{Cancer}) = 1.2$, the weight of this model is $0.2 \cdot 0.5 \cdot 0.8 \cdot 0.8$. This theory has eight other models. The WFOMC can be calculated by summing the weights of all nine models.

2.1 Converting Inference for SRL Models into WFOMC

For many SRL models, (lifted) inference can be converted into a WFOMC problem. As an example, consider a Markov logic network (MLN) [24] with weighted formulae $(w_1 : F_1, \dots, w_k : F_k)$. For every weighted formula $w_i : F_i$ of this MLN, let theory T have a sentence $\mathsf{Aux}_i(x, \dots) \Leftrightarrow F_i$ such that Aux_i is a predicate having all LVs appearing in F_i. Assuming $\Phi(\mathsf{Aux}_i) = \exp(w_i)$, and Φ and $\overline{\Phi}$ are 1 for the other predicates, the *partition function* of the MLN is equal to $\mathrm{WFOMC}(T)$.

2.2 Calculating the WFOMC of a Theory

We now describe a set of rules $\mathcal{R}$ that can be applied to a theory to find its WFOMC efficiently; for more details, readers are directed to [28], [22] or [11]. We use the following theory T with two clauses and four PRVs ($\mathsf{S}(x, m)$, $\mathsf{R}(x, m)$, $\mathsf{T}(x)$ and $\mathsf{Q}(x)$) as our running example:

$$\forall x \in \Delta_x, m \in \Delta_m : \mathsf{Q}(x) \vee \mathsf{R}(x, m) \vee \mathsf{S}(x, m) \qquad \forall x \in \Delta_x, m \in \Delta_m : \mathsf{S}(x, m) \vee \mathsf{T}(x)$$

Lifted Decomposition Assume we ground x in T. Then the clauses mentioning an arbitrary $X_i \in \Delta_x$ are $\forall m \in \Delta_m : \mathsf{Q}(X_i) \vee \mathsf{R}(X_i, m) \vee \mathsf{S}(X_i, m)$ and $\forall m \in \Delta_m : \mathsf{S}(X_i, m) \vee \mathsf{T}(X_i)$. These clauses are totally disconnected from clauses mentioning $X_j \in \Delta_x$ ($j \neq i$), and are the same up to renaming X_i to X_j. Given the exchangeability of the individuals, we can calculate the WFOMC of only the clauses mentioning X_i and raise the result to the power of the number of connected components ($|\Delta_x|$). Assuming T_1 is the theory that results from substituting x with X_i, $\mathrm{WFOMC}(T) = \mathrm{WFOMC}(T_1)^{|\Delta_x|}$.

Case-Analysis The WFOMC of T_1 can be computed by a case analysis over different assignments of values to a ground PRV, e.g., $\mathsf{Q}(X_i)$. Let T_2 and T_3 represent $T_1 \wedge \mathsf{Q}(X_i)$ and $T_1 \wedge \neg\mathsf{Q}(X_i)$ respectively. Then, $\mathrm{WFOMC}(T_1) = \mathrm{WFOMC}(T_2) + \mathrm{WFOMC}(T_3)$. We follow the process for T_3 (the process for T_2 will be similar) having clauses $\neg\mathsf{Q}(X_i)$, $\forall m \in \Delta_m : \mathsf{Q}(X_i) \vee \mathsf{R}(X_i, m) \vee \mathsf{S}(X_i, m)$ and $\forall m \in \Delta_m : \mathsf{S}(X_i, m) \vee \mathsf{T}(X_i)$.

Unit Propagation When a clause in the theory has only one literal, we can propagate the effect of this clause through the theory and remove it[4]. In T_3, $\neg\mathsf{Q}(X_i)$ is a unit clause. Having this unit clause, we can simplify the second clause and get the theory T_4 having clauses $\forall m \in \Delta_m : \mathsf{R}(X_i, m) \vee \mathsf{S}(X_i, m)$ and $\forall m \in \Delta_m : \mathsf{S}(X_i, m) \vee \mathsf{T}(X_i)$.

Lifted Case-Analysis Case-analysis can be done for PRVs having one logical variable in a lifted way. Consider the $\mathsf{S}(X_i, m)$ in T_4. Due to the exchangeability of the individuals, we do not have to consider all possible assignments to all ground PRVs of $\mathsf{S}(X_i, m)$, but only the ones where the number of individuals $M \in \Delta_m$ for which $\mathsf{S}(X_i, M)$ is True (or equivalently False) is different. This means considering $|\Delta_m| + 1$ cases suffices, corresponding to $\mathsf{S}(X_i, M)$ being True for exactly $j = 0, \dots, |\Delta_m|$ individuals. Note that we must multiply by $\binom{|\Delta_m|}{j}$ to account for the number

[3] Equivalently, we can disjoin $x_1 = x_2$ or $x = C$ to the clause.

[4] Note that unit propagation may remove clauses and random variables from the theory. To account for them, *smoothing* multiplies the WFOMC by $2^{\#rv}$, where $\#rv$ represents the number of removed variables.

of ways one can select j out of $|\Delta_m|$ individuals. Let T_{4j} represent T_4 with two more clauses: $\forall m \in \Delta_{m_T} : \mathsf{S}(X_i, m)$ and $\forall m \in \Delta_{m_F} : \neg\mathsf{S}(X_i, m)$, where Δ_{m_T} represents the j individuals in Δ_m for which $\mathsf{S}(X_i, M)$ is True, and Δ_{m_F} represents the other $|\Delta_m| - j$ individuals. Then $\text{WFOMC}(T_4) = \sum_{j=0}^{|\Delta_m|} \binom{|\Delta_m|}{j} \text{WFOMC}(T_{4j})$.

Shattering In T_{4j}, the individuals in Δ_m are no longer exchangeable: we know different things about those in Δ_{m_T} and those in Δ_{m_F}. We need to shatter every clause having individuals coming from Δ_m to make the theory exchangeable. To do so, the clause $\forall m \in \Delta_m : \mathsf{R}(X_i, m) \vee \mathsf{S}(X_i, m)$ in T_{4j} must be shattered to $\forall m \in \Delta_{m_T} : \mathsf{R}(X_i, m) \vee \mathsf{S}(X_i, m)$ and $\forall m \in \Delta_{m_F} : \mathsf{R}(X_i, m) \vee \mathsf{S}(X_i, m)$ (and similarly for the other formulae). The shattered theory T_{5j} after unit propagation will have clauses $\forall m \in \Delta_{m_F} : \mathsf{R}(X_i, m)$ and $\forall m \in \Delta_{m_F} : \mathsf{T}(X_i)$.

Decomposition, Caching, and Grounding In T_{5j}, the two clauses have different PRVs, i.e., they are disconnected. In such cases, we apply decomposition, i.e., find the WFOMC of each connected component separately and return the product. The WFOMC of the theory can be found by continuing to apply the above rules. In all the above steps, after finding the WFOMC of each (sub-)theory, we store the results in a cache so we can reuse them if the same WFOMC is required again. By following these steps, one can find the WFOMC of many theories in polynomial time. However, if we reach a point where none of the above rules are applicable, we ground one of the populations which makes the process exponential in the number of individuals.

2.3 Domain-Liftability

The following notions allow us to study the power of a set of lifted inference rules.

Definition 1. A theory is **domain-liftable** [31] if calculating its WFOMC is polynomial in $|\Delta_{x_1}|, |\Delta_{x_2}|, \ldots, |\Delta_{x_k}|$ where $x_1, x_2, \ldots, x_k$ represent the LVs in the theory. A class C of theories is domain-liftable if $\forall T \in C$, T is domain-liftable.

So far, two main classes of domain-liftable theories have been recognized: FO^2 [30, 31] and *recursively unary* [15, 22].

Definition 2. A theory is in FO^2 if all its clauses have up to two LVs.

Definition 3. A theory T is *recursively unary (RU)* if for every theory T' resulting from applying rules in $\mathcal{R}$ except for lifted case analysis to T, until no more rules apply, there exists some unary PRV in T' and a generic case of lifted case-analysis on this unary PRV is itself *RU*.

Note that the time needed to check whether a theory is in FO^2 or RU is independent of the domain sizes in the theory. For FO^2, the membership check can be done in time linear in the size of the theory, whereas for *RU*, only a worst-case exponential procedure is known. Thus, FO^2 currently offers a faster membership check than *RU*, but as we show later, *RU* subsumes FO^2. This gives rise to a trade-off between fast membership checking and modeling power for, e.g., lifted learning purposes.

3 The Domain Recursion Rule

Van den Broeck [31] considered another rule called *domain recursion* in the set of rules for calculating the WFOMC of a theory. The intuition behind domain recursion is that it modifies a domain Δ_x by making one element explicit: $\Delta_x = \Delta_{x'} \cup \{A\}$ with $A \notin \Delta_{x'}$. Next, clauses are rewritten in terms of $\Delta_{x'}$ and A while removing Δ_x from the theory entirely. Then, by applying standard rules in $\mathcal{R}$ on this modified theory, the problem is reduced to a WFOMC problem on a theory identical to the original one, except that Δ_x is replaced by the smaller domain $\Delta_{x'}$. This lets us compute WFOMC using dynamic programming. We refer to $\mathcal{R}$ extended with the domain recursion rule as $\mathcal{R}^{\mathcal{D}}$.

Example 2. Suppose we have a theory whose only clause is $\forall x, y \in \Delta_p : \neg\mathsf{Friend}(x, y) \vee \mathsf{Friend}(y, x)$, stating if x is friends with y, y is also friends with x. One way to calculate the WFOMC of this theory is by grounding only one individual in Δ_p and then using $\mathcal{R}$. Let A be an individual in Δ_p and let $\Delta_{p'} = \Delta_p - \{A\}$. We can (using domain recursion) rewrite the theory as: $\forall x \in \Delta_{p'} : \neg\mathsf{Friend}(x, A) \vee \mathsf{Friend}(A, x)$, $\forall y \in \Delta_{p'} : \neg\mathsf{Friend}(A, y) \vee \mathsf{Friend}(y, A)$, and $\forall x, y \in \Delta_{p'} : \neg\mathsf{Friend}(x, y) \vee \mathsf{Friend}(y, x)$. Lifted case-analysis on $\mathsf{Friend}(p', A)$ and $\mathsf{Friend}(A, p')$, shattering and unit propagation give $\forall x, y \in \Delta_{p'} : \neg\mathsf{Friend}(x, y) \vee \mathsf{Friend}(y, x)$. This theory is equivalent to our initial theory, with the only difference being that the population of people has decreased by one. By keeping a cache of the values of each sub-theory, one can verify that this process finds the WFOMC of the above theory in polynomial time.

Note that the theory in Example 2 is in FO^2 and as proved in [27], its WFOMC can be computed without using the domain recursion rule[5]. This proof has caused the domain recursion rule to be forgotten in the lifted inference community. In the next section, we revive this rule and identify a class of theories that are only domain-liftable when using the domain recursion rule.

4 Domain Recursion Makes More Theories Domain-Liftable

In this section, we show three example theories that are not domain-liftable when using $\mathcal{R}$, yet become domain-liftable with domain recursion.

S4 Clause: Beame *et al.* [3] identified a clause (S4) with four binary PRVs having the same predicate and proved that, even though the rules $\mathcal{R}$ in Section 2.2 cannot calculate the WFOMC of that clause, there is a polynomial-time algorithm for finding its WFOMC. They concluded that this set of rules $\mathcal{R}$ for finding the WFOMC of theories does not suffice, asking for new rules to compute their theory. We prove that adding domain recursion to the set achieves this goal.

Proposition 1. *The theory consisting of the S4 clause* $\forall x_1, x_2 \in \Delta_x, y_1, y_2 \in \Delta_y : \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) \vee \neg\mathsf{S}(x_1, y_2)$ *is domain-liftable using* $\mathcal{R}^{\mathcal{D}}$.

Symmetric Transitivity: Domain-liftable calculation of WFOMC for the transitivity formula is a long-standing open problem. Symmetric transitivity is easier as its model count corresponds to the Bell number, but solving it using general-purpose rules has been an open problem. Consider clauses $\forall x, y, z \in \Delta_p : \neg\mathsf{F}(x, y) \vee \neg\mathsf{F}(y, z) \vee \mathsf{F}(x, z)$ and $\forall x, y \in \Delta_p : \neg\mathsf{F}(x, y) \vee \mathsf{F}(y, x)$ defining a symmetric transitivity relation. For example, Δ_p may indicate the population of people and F may indicate friendship.

Proposition 2. *The symmetric-transitivity theory is domain-liftable using* $\mathcal{R}^{\mathcal{D}}$.

Birthday Paradox: The birthday paradox problem [2] is to compute the probability that in a set of n randomly chosen people, two of them have the same birthday. A first-order encoding of this problem requires computing the WFOMC for a theory with clauses $\forall p \in \Delta_p, \exists d \in \Delta_d : \mathsf{Born}(p, d)$, $\forall p \in \Delta_p, d_1, d_2 \in \Delta_d : \neg\mathsf{Born}(p, d_1) \vee \neg\mathsf{Born}(p, d_2)$, and $\forall p_1, p_2 \in \Delta_p, d \in \Delta_d : \neg\mathsf{Born}(p_1, d) \vee \neg\mathsf{Born}(p_2, d)$, where Δ_p and Δ_d represent the population of people and days. The first two clauses impose the condition that every person is born in exactly one day, and the third clause states the "no two people are born on the same day" query.

Proposition 3. *The birthday-paradox theory is domain-liftable using* $\mathcal{R}^{\mathcal{D}}$.

5 New Domain-Liftable Classes: S^2FO^2 and S^2RU

In this section, we identify new domain-liftable classes, enabled by the domain recursion rule.

Definition 4. Let $\alpha(\mathsf{S})$ be a clausal theory that uses a single binary predicate S, such that each clause has exactly two different literals of S. Let $\alpha = \alpha(\mathsf{S}_1) \wedge \alpha(\mathsf{S}_2) \wedge \cdots \wedge \alpha(\mathsf{S}_n)$ where the S_i are different binary predicates. Let β be a theory where all clauses contain at most one S_i literal, and the clauses that contain an S_i literal contain no other literals with more than one LV. Then, S^2FO^2 and S^2RU are the classes of theories of the form $\alpha \wedge \beta$ where $\beta \in FO^2$ and $\beta \in RU$ respectively.

Theorem 1. S^2FO^2 *and* S^2RU *are domain-liftable using* $\mathcal{R}^{\mathcal{D}}$.

It can be easily verified that membership checking for S^2FO^2 and S^2RU is not harder than for FO^2 and RU, respectively.

Example 3. Suppose we have a set Δ_j of jobs and a set Δ_v of volunteers. Every volunteer must be assigned to at most one job, and every job requires no more than one person. If the job involves

[5] This can be done by realizing that the theory is disconnected in the grounding for every pair (A, B) of individuals and applying the lifted case-analysis.

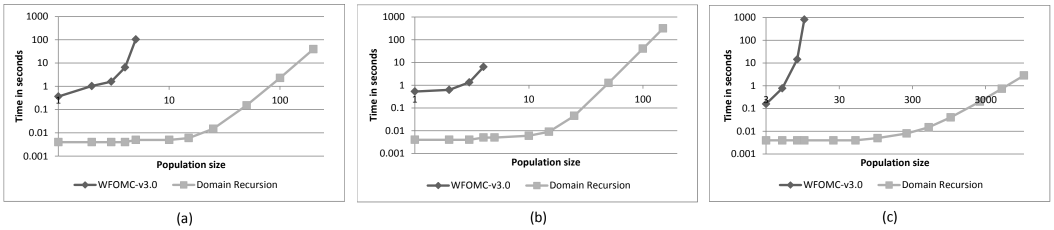

Figure 1: Run-times for calculating the WFOMC of (a) the theory in Example 3, (b) the S4 clause, and (c) symmetric transitivity, using the WFOMC-v3.0 software (which only uses $\mathcal{R}$) and comparing it to the case where we use the domain recursion rule, referred to as *Domain Recursion* in the diagrams.

working with gas, the assigned volunteer must be a non-smoker. And we know that smokers are most probably friends with each other. Then we will have the following first-order theory:

$$\forall v_1, v_2 \in \Delta_v, j \in \Delta_j : \neg\mathsf{Assigned}(v_1, j) \vee \neg\mathsf{Assigned}(v_2, j)$$
$$\forall v \in \Delta_v, j_1, j_2 \in \Delta_j : \neg\mathsf{Assigned}(v, j_1) \vee \neg\mathsf{Assigned}(v, j_2)$$
$$\forall v \in \Delta_v, j \in \Delta_j : \mathsf{InvolvesGas}(j) \wedge \mathsf{Assigned}(v, j) \Rightarrow \neg\mathsf{Smokes}(v)$$
$$\forall v_1, v_2 \in \Delta_v : \mathsf{Aux}(v_1, v_2) \Leftrightarrow (\mathsf{Smokes}(v_1) \wedge \mathsf{Friends}(v_1, v_2) \Rightarrow \mathsf{Smokes}(v_2))$$

Predicate Aux is added to capture the probability assigned to the last rule (as in MLNs). This theory is not in FO^2, not in RU, and is not domain-liftable using $\mathcal{R}$. However, the first two clauses are of the form described in Lemma 1, the third and fourth are in FO^2 (and also in RU), and the third clause, which contains $\mathsf{Assigned}(v, j)$, has no other PRVs with more than one LV. Therefore, this theory is in S^2FO^2 (and also in S^2RU) and domain-liftable based on Theorem 1.

Proposition 4. *$FO^2 \subset RU$, $FO^2 \subset S^2FO^2$, $FO^2 \subset S^2RU$, $RU \subset S^2RU$, $S^2FO^2 \subset S^2RU$.*

6 Experiments and Results

In order to see the effect of using domain recursion in practice, we find the WFOMC of three theories with and without using the domain recursion rule: (a) the theory in Example 3, (b) the S4 clause, and (c) the symmetric-transitivity theory. We implemented the domain recursion rule in C++ and compiled the codes using the g++ compiler. We compare our results with the WFOMC-v3.0 software[6]. Since this software requires domain-liftable input theories, for the first theory we grounded the jobs, for the second we grounded Δ_x, and for the third we grounded Δ_p. For each of these three theories, assuming $|\Delta_x| = n$ for all LVs x in the theory, we varied n and plotted the run-time as a function of n. All experiments were done on a 2.8GH core with 4GB RAM under MacOSX. The run-times are reported in seconds. We allowed a maximum of 1000 seconds for each run.

Obtained results can be viewed in Fig. 1. These results are consistent with our theory and indicate the clear advantage of using the domain recursion rule in practice. In Fig. 1(a), the slope of the diagram for domain recursion is approximately 4 which indicates the degree of the polynomial for the time complexity. Similar analysis can be done for the results on the $S4$ clause and the symmetric-transitivity clauses represented in Fig. 1(b), (c). The slope of the diagram in these two diagrams is around 5 and 2 respectively, indicating that the time complexity for finding their WFOMC are n^5 and n^2 respectively, where n is the size of the population.

7 Discussion

We can categorize theories with respect to the domain recursion rule as: (1) theories proved to be domain-liftable using domain recursion (e.g., S4, symmetric transitivity, and theories in S^2FO^2), (2) theories that are domain-liftable using domain recursion, but we have not identified them yet as such, and (3) theories that are not domain-liftable even when using domain recursion. We leave discovering and characterizing the theories in category 2 and 3 as future work. But here we show that

[6] Available at: https://dtai.cs.kuleuven.be/software/wfomc

even though the theories in category 3 are not domain-liftable using domain recursion, this rule may still result in exponential speedups for these theories.

Consider the (non-symmetric) transitivity rule: $\forall x, y, z \in \Delta_p : \neg\mathsf{Friend}(x, y) \vee \neg\mathsf{Friend}(y, z) \vee \mathsf{Friend}(x, z)$. Since none of the rules in $\mathcal{R}$ apply to the above theory, the existing lifted inference engines ground Δ_p and calculate the weighted model count (WMC) of the ground theory. By grounding Δ_p, these engines lose great amounts of symmetry. Suppose $\Delta_p = \{A, B, C\}$ and assume we select $\mathsf{Friend}(A, B)$ and $\mathsf{Friend}(A, C)$ as the first two random variables for case-analysis. Due to the exchangeability of the individuals, the case where $\mathsf{Friend}(A, B)$ and $\mathsf{Friend}(A, C)$ are assigned to True and False respectively has the same WMC as the case where they are assigned to False and True. However, the current engines fail to exploit this symmetry as they consider grounded individuals non-exchangeable.

By applying domain recursion to the above theory instead of fully grounding it, one can exploit the symmetries of the theory. Suppose $\Delta_{p'} = \Delta_p - \{P\}$. Then we can rewrite the theory as follows:

$$\begin{aligned}
\forall y, z \in \Delta_{p'} &: \neg\mathsf{Friend}(P, y) \vee \neg\mathsf{Friend}(y, z) \vee \mathsf{Friend}(P, z)\\
\forall x, z \in \Delta_{p'} &: \neg\mathsf{Friend}(x, P) \vee \neg\mathsf{Friend}(P, z) \vee \mathsf{Friend}(x, z)\\
\forall x, y \in \Delta_{p'} &: \neg\mathsf{Friend}(x, y) \vee \neg\mathsf{Friend}(y, P) \vee \mathsf{Friend}(x, P)\\
\forall x, y, z \in \Delta_{p'} &: \neg\mathsf{Friend}(x, y) \vee \neg\mathsf{Friend}(y, z) \vee \mathsf{Friend}(x, z)
\end{aligned}$$

Now if we apply lifted case analysis on $\mathsf{Friend}(P, y)$ (or equivalently on $\mathsf{Friend}(P, z)$), we do not get back the same theory with reduced population and calculating the WFOMC is still exponential. However, we only generate one branch for the case where $\mathsf{Friend}(P, y)$ is True only once. This branch covers both the symmetric cases mentioned above. Exploiting these symmetries reduces the time-complexity exponentially.

This suggests that for any given theory, when the rules in $\mathcal{R}$ are not applicable one may want to try the domain recursion rule before giving up and resorting to grounding a population.

8 Conclusion

We identified new classes of domain-liftable theories called S^2FO^2 and S^2RU by reviving the domain recursion rule. We also demonstrated how this rule is useful for theories outside these classes. Our work opens up a future research direction for identifying and characterizing larger classes of theories that are domain-liftable using domain recursion. It also helps us get closer to finding a dichotomy between the theories that are domain-liftable and those that are not, similar to the dichotomy result of Dalvi and Suciu [6] for query answering in probabilistic databases.

It has been shown [15, 16] that compiling the WFOMC rules into low-level programs (e.g., C++ programs) offers a (approx.) 175x speedup compared to other approaches. While compiling the previously known rules to low-level programs was straightforward, compiling the domain recursion rule to low-level programs without using recursion might be tricky as it relies on the population size of the logical variables. A future research direction would be finding if the domain recursion rule can be efficiently compiled into low-level programs, and measuring the amount of speedup it offers.

9 Proofs of the Theorems, Propositions, and Lemmas

9.1 Proof of Proposition 1

Proof. Let $\Delta_{x'} = \Delta_x - \{N\}$. Applying *domain recursion* on Δ_x (choosing Δ_y is analogous) gives the following *shattered* theory on the reduced domain $\Delta_{x'}$.

$$\begin{aligned}
\forall x_1, x_2 \in \Delta_{x'},\ y_1, y_2 \in \Delta_y &: \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_1, y_2) \vee \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) && (1)\\
\forall x_2 \in \Delta_{x'},\ y_1, y_2 \in \Delta_y &: \mathsf{S}(N, y_1) \vee \neg\mathsf{S}(N, y_2) \vee \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) && (2)\\
\forall x_1 \in \Delta_{x'},\ y_1, y_2 \in \Delta_y &: \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_1, y_2) \vee \neg\mathsf{S}(N, y_1) \vee \mathsf{S}(N, y_2) && (3)\\
\forall y_1, y_2 \in \Delta_y &: \mathsf{S}(N, y_1) \vee \neg\mathsf{S}(N, y_2) \vee \neg\mathsf{S}(N, y_1) \vee \mathsf{S}(N, y_2) && (4)
\end{aligned}$$

We now reach to the standard rules $\mathcal{R}$ to simplify the output of domain recursion. The last clause is a *tautology* and can be dropped. The theory contains a unary PRV, namely $\mathsf{S}(N, y)$, which is a candidate for *lifted case-analysis*. Let $\Delta_{y_T} \subseteq \Delta_y$ be the individuals of Δ_y for which $\mathsf{S}(N, y)$ is satisfied, and let $\Delta_{y_F} = \Delta_y \setminus \Delta_{y_T}$ be its complement in Δ_y. This gives

$$\begin{aligned}
\forall x_1, x_2 \in \Delta_{x'},\ y_1, y_2 \in \Delta_y &: \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_1, y_2) \vee \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) && (5)\\
\forall x_2 \in \Delta_{x'},\ y_1, y_2 \in \Delta_y &: \mathsf{S}(N, y_1) \vee \neg\mathsf{S}(N, y_2) \vee \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) && (6)\\
\forall x_1 \in \Delta_{x'},\ y_1, y_2 \in \Delta_y &: \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_1, y_2) \vee \neg\mathsf{S}(N, y_1) \vee \mathsf{S}(N, y_2) && (7)\\
\forall y \in \Delta_{y_T} &: \mathsf{S}(N, y) && (8)\\
\forall y \in \Delta_{y_F} &: \neg\mathsf{S}(N, y) && (9)
\end{aligned}$$

Unit propagation creates two independent theories: one containing the $\mathsf{S}(N, y)$ atoms, which is trivially liftable, and one containing the other atoms, namely

$$\begin{aligned}
\forall x_1, x_2 \in \Delta_{x'},\ y_1, y_2 \in \Delta_y &: \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_1, y_2) \vee \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) && (10)\\
\forall x_2 \in \Delta_{x'},\ y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_F} &: \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) && (11)\\
\forall x_1 \in \Delta_{x'},\ y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_F} &: \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_1, y_2) && (12)
\end{aligned}$$

The last two clauses are equivalent, hence, we have

$$\begin{aligned}
\forall x_1, x_2 \in \Delta_{x'},\ y_1, y_2 \in \Delta_y &: \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_1, y_2) \vee \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) && (13)\\
\forall x \in \Delta_{x'},\ y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} &: \neg\mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) && (14)
\end{aligned}$$

After *shattering*, we get four copies of the first clause:

$$\begin{aligned}
\forall x_1, x_2 \in \Delta_{x'},\ y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_T} &: \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_1, y_2) \vee \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) && (15)\\
\forall x_1, x_2 \in \Delta_{x'},\ y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_F} &: \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_1, y_2) \vee \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) && (16)\\
\forall x_1, x_2 \in \Delta_{x'},\ y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} &: \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_1, y_2) \vee \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) && (17)\\
\forall x_1, x_2 \in \Delta_{x'},\ y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_F} &: \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_1, y_2) \vee \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) && (18)\\
\forall x \in \Delta_{x'},\ y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} &: \neg\mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) && (19)
\end{aligned}$$

The second and third clauses are subsumed by the last clause, and can be removed:

$$\begin{aligned}
\forall x_1, x_2 \in \Delta_{x'},\ y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_T} &: \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_1, y_2) \vee \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) && (20)\\
\forall x_1, x_2 \in \Delta_{x'},\ y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_F} &: \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_1, y_2) \vee \neg\mathsf{S}(x_2, y_1) \vee \mathsf{S}(x_2, y_2) && (21)\\
\forall x \in \Delta_{x'},\ y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} &: \neg\mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) && (22)
\end{aligned}$$

Let us now consider the last clause, and ignore the first two for the time being. The last clause is actually in FO^2, and the *Skolemization*-rewriting of reused FO^2 variables [30] can be applied, for example to y_2 in its second PRV. The last clause is thus replaced by

$$\begin{aligned}
\forall x \in \Delta_{x'},\ y \in \Delta_{y_F} &: \neg\mathsf{S}(x, y) \vee \neg\mathsf{A}(x) && (23)\\
\forall x \in \Delta_{x'},\ y \in \Delta_{y_T} &: \mathsf{S}(x, y) \vee \mathsf{A}(x) && (24)\\
\forall x \in \Delta_{x'} &: \mathsf{A}(x) \vee \mathsf{B}(x) && (25)\\
\forall x \in \Delta_{x'},\ y \in \Delta_{y_T} &: \mathsf{S}(x, y) \vee \mathsf{B}(x) && (26)
\end{aligned}$$

Next, we perform *lifted case-analysis* on $\mathsf{A}(x')$. Let $\Delta_\alpha \subseteq \Delta_{x'}$ be the individuals in $\Delta_{x'}$ for which $\mathsf{A}(x')$ is satisfied, and let $\Delta_{\bar{\alpha}} = \Delta_{x'} \setminus \Delta_\alpha$ be its complement in $\Delta_{x'}$:

$$\begin{aligned}
\forall x \in \Delta_{x'},\ y \in \Delta_{y_F} &: \neg\mathsf{S}(x, y) \vee \neg\mathsf{A}(x) && (27)\\
\forall x \in \Delta_{x'},\ y \in \Delta_{y_T} &: \mathsf{S}(x, y) \vee \mathsf{A}(x) && (28)\\
\forall x \in \Delta_{x'} &: \mathsf{A}(x) \vee \mathsf{B}(x) && (29)\\
\forall x \in \Delta_{x'},\ y \in \Delta_{y_T} &: \mathsf{S}(x, y) \vee \mathsf{B}(x) && (30)\\
\forall x \in \Delta_\alpha &: \mathsf{A}(x) && (31)\\
\forall x \in \Delta_{\bar{\alpha}} &: \neg\mathsf{A}(x) && (32)
\end{aligned}$$

Unit propagation gives two independent theories: a theory containing the predicate A, which is trivially liftable, and the theory

$$\begin{aligned}
\forall x \in \Delta_\alpha,\ y \in \Delta_{y_F} &: \neg\mathsf{S}(x, y) && (33)\\
\forall x \in \Delta_{\bar{\alpha}},\ y \in \Delta_{y_T} &: \mathsf{S}(x, y) && (34)\\
\forall x \in \Delta_{\bar{\alpha}} &: \mathsf{B}(x) && (35)\\
\forall x \in \Delta_\alpha,\ y \in \Delta_{y_T} &: \mathsf{S}(x, y) \vee \mathsf{B}(x) && (36)
\end{aligned}$$

Next, we perform *atom counting* on $\mathsf{B}(x')$. Let $\Delta_\beta \subseteq \Delta_\alpha$ be the individuals of Δ_α for which $\mathsf{B}(x')$ is satisfied, and let $\Delta_{\bar\beta} = \Delta_\alpha \setminus \Delta_\beta$ be its complement in Δ_α. In other words, the original domain Δ_x is now split up into four parts: $\Delta_{\bar x} = \{N\}$, $\Delta_{\bar\alpha}$, Δ_β, and $\Delta_{\bar\beta}$. This gives the theory

$$\begin{aligned}
\forall x \in \Delta_\alpha,\ y \in \Delta_{y_F} &: \neg\mathsf{S}(x,y) && (37)\\
\forall x \in \Delta_{\bar\alpha},\ y \in \Delta_{y_T} &: \mathsf{S}(x,y) && (38)\\
\forall x \in \Delta_{\bar\alpha} &: \mathsf{B}(x) && (39)\\
\forall x \in \Delta_\alpha,\ y \in \Delta_{y_T} &: \mathsf{S}(x,y) \lor \mathsf{B}(x) && (40)\\
\forall x \in \Delta_\beta &: \mathsf{B}(x) && (41)\\
\forall x \in \Delta_{\bar\beta} &: \neg\mathsf{B}(x) && (42)
\end{aligned}$$

Unit propagation gives two independent theories: a theory containing the predicate B, which is trivially liftable, and the theory

$$\begin{aligned}
\forall x \in \Delta_\alpha,\ y \in \Delta_{y_F} &: \neg\mathsf{S}(x,y) && (43)\\
\forall x \in \Delta_{\bar\alpha},\ y \in \Delta_{y_T} &: \mathsf{S}(x,y) && (44)\\
\forall x \in \Delta_{\bar\beta},\ y \in \Delta_{y_T} &: \mathsf{S}(x,y) && (45)
\end{aligned}$$

We now reintroduce the first removed clause. Clause 20 has nine copies after *shattering*:

$$\begin{aligned}
\forall x_1 \in \Delta_{\bar\alpha}, x_2 \in \Delta_{\bar\alpha},\ y_1,y_2 \in \Delta_{y_T} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (46)\\
\forall x_1 \in \Delta_{\bar\alpha}, x_2 \in \Delta_{\beta},\ y_1,y_2 \in \Delta_{y_T} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (47)\\
\forall x_1 \in \Delta_{\bar\alpha}, x_2 \in \Delta_{\bar\beta},\ y_1,y_2 \in \Delta_{y_T} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (48)\\
\forall x_1 \in \Delta_{\beta}, x_2 \in \Delta_{\bar\alpha},\ y_1,y_2 \in \Delta_{y_T} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (49)\\
\forall x_1 \in \Delta_{\beta}, x_2 \in \Delta_{\beta},\ y_1,y_2 \in \Delta_{y_T} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (50)\\
\forall x_1 \in \Delta_{\beta}, x_2 \in \Delta_{\bar\beta},\ y_1,y_2 \in \Delta_{y_T} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (51)\\
\forall x_1 \in \Delta_{\bar\beta}, x_2 \in \Delta_{\bar\alpha},\ y_1,y_2 \in \Delta_{y_T} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (52)\\
\forall x_1 \in \Delta_{\bar\beta}, x_2 \in \Delta_{\beta},\ y_1,y_2 \in \Delta_{y_T} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (53)\\
\forall x_1 \in \Delta_{\bar\beta}, x_2 \in \Delta_{\bar\beta},\ y_1,y_2 \in \Delta_{y_T} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (54)
\end{aligned}$$

Unit propagation of clauses 44 and 45 satisfies any clause that has a positive literal whose x domain is $\Delta_{\bar\alpha}$ or $\Delta_{\bar\beta}$. This removes all clauses except for

$$\forall x_1 \in \Delta_{\beta}, x_2 \in \Delta_{\beta},\ y_1,y_2 \in \Delta_{y_T} : \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) \qquad (55)$$

We now reintroduce the second removed clause. Clause 21 has four copies after *shattering*:

$$\begin{aligned}
\forall x_1 \in \Delta_{\alpha}, x_2 \in \Delta_{\alpha},\ y_1,y_2 \in \Delta_{y_F} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (56)\\
\forall x_1 \in \Delta_{\alpha}, x_2 \in \Delta_{\bar\alpha},\ y_1,y_2 \in \Delta_{y_F} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (57)\\
\forall x_1 \in \Delta_{\bar\alpha}, x_2 \in \Delta_{\alpha},\ y_1,y_2 \in \Delta_{y_F} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (58)\\
\forall x_1 \in \Delta_{\bar\alpha}, x_2 \in \Delta_{\bar\alpha},\ y_1,y_2 \in \Delta_{y_F} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (59)
\end{aligned}$$

Unit propagation of clauses 43 satisfies any clause that has a negative literal whose x domain is α. This removes all clauses except for

$$\forall x_1 \in \Delta_{\bar\alpha}, x_2 \in \Delta_{\bar\alpha},\ y_1,y_2 \in \Delta_{y_F} : \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) \qquad (60)$$

Putting it all together, we have the theory

$$\begin{aligned}
\forall x_1, x_2 \in \Delta_{\beta},\ y_1,y_2 \in \Delta_{y_T} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_2,y_2) && (61)\\
\forall x_1, x_2 \in \Delta_{\bar\alpha},\ y_1,y_2 \in \Delta_{y_F} &: \mathsf{S}(x_1,y_1) \lor \neg\mathsf{S}(x_1,y_2) \lor \neg\mathsf{S}(x_2,y_1) \lor \mathsf{S}(x_1,y_2) && (62)\\
\forall x \in \Delta_\alpha,\ y \in \Delta_{y_F} &: \neg\mathsf{S}(x,y) && (63)\\
\forall x \in \Delta_{\bar\alpha},\ y \in \Delta_{y_T} &: \mathsf{S}(x,y) && (64)\\
\forall x \in \Delta_{\bar\beta},\ y \in \Delta_{y_T} &: \mathsf{S}(x,y) && (65)
\end{aligned}$$

These five clauses are all *independent*. The last three are trivially liftable. The first two are simply copies of S4 with modified domains Δ_β, Δ_{y_T}, $\Delta_{\bar\alpha}$ and Δ_{y_F} instead of Δ_x and Δ_y. However, we have that $|\Delta_\beta| < |\Delta_x|$, $|\Delta_{y_T}| \leq |\Delta_y|$, $|\Delta_{\bar\alpha}| < |\Delta_x|$, and $|\Delta_{y_F}| \leq |\Delta_y|$. The recursion is thus guaranteed to terminate with $\Delta_\beta = \Delta_{\bar\alpha} = \emptyset$. By keeping a cache of WFOMCs for all sizes of Δ_x and Δ_y, we can compute the WFOMC of S4 in PTIME. □

9.2 Proof of Proposition 2

Proof. Symmetric-transitivity has the following two sentences:

$$\forall x, y, z \in \Delta_p : \neg \mathsf{F}(x, y) \lor \neg \mathsf{F}(y, z) \lor \mathsf{F}(x, z) \tag{66}$$
$$\forall x, y \in \Delta_p : \neg \mathsf{F}(x, y) \lor \mathsf{F}(y, x) \tag{67}$$

Assuming $\Delta_q = \Delta_p - \{N\}$:

$$\forall y, z \in \Delta_q : \neg \mathsf{F}(N, y) \lor \neg \mathsf{F}(y, z) \lor \mathsf{F}(N, z) \tag{68}$$
$$\forall x, z \in \Delta_q : \neg \mathsf{F}(x, N) \lor \neg \mathsf{F}(N, z) \lor \mathsf{F}(x, z) \tag{69}$$
$$\forall x, y \in \Delta_q : \neg \mathsf{F}(x, y) \lor \neg \mathsf{F}(y, N) \lor \mathsf{F}(x, N) \tag{70}$$
$$\forall x, y, z \in \Delta_q : \neg \mathsf{F}(x, y) \lor \neg \mathsf{F}(y, z) \lor \mathsf{F}(x, z) \tag{71}$$
$$\forall y \in \Delta_q : \neg \mathsf{F}(N, y) \lor \mathsf{F}(y, N) \tag{72}$$
$$\forall x \in \Delta_q : \neg \mathsf{F}(x, N) \lor \mathsf{F}(N, x) \tag{73}$$
$$\forall x, y \in \Delta_q : \neg \mathsf{F}(x, y) \lor \mathsf{F}(y, x) \tag{74}$$

Lifted case-analysis on $\mathsf{F}(N, q)$ assuming Δ_{q_T} contains individuals in Δ_q for which $\mathsf{F}(N, q)$ is true and Δ_{q_F} is the other individuals in Δ_q:

$$\forall y, z \in \Delta_q : \neg \mathsf{F}(N, y) \lor \neg \mathsf{F}(y, z) \lor \mathsf{F}(N, z) \tag{75}$$
$$\forall x, z \in \Delta_q : \neg \mathsf{F}(x, N) \lor \neg \mathsf{F}(N, z) \lor \mathsf{F}(x, z) \tag{76}$$
$$\forall x, y \in \Delta_q : \neg \mathsf{F}(x, y) \lor \neg \mathsf{F}(y, N) \lor \mathsf{F}(x, N) \tag{77}$$
$$\forall x, y, z \in \Delta_q : \neg \mathsf{F}(x, y) \lor \neg \mathsf{F}(y, z) \lor \mathsf{F}(x, z) \tag{78}$$
$$\forall y \in \Delta_q : \neg \mathsf{F}(N, y) \lor \mathsf{F}(y, N) \tag{79}$$
$$\forall x \in \Delta_q : \neg \mathsf{F}(x, N) \lor \mathsf{F}(N, x) \tag{80}$$
$$\forall x, y \in \Delta_q : \neg \mathsf{F}(x, y) \lor \mathsf{F}(y, x) \tag{81}$$
$$\forall x \in \Delta_{q_T} : \mathsf{F}(N, x) \tag{82}$$
$$\forall x \in \Delta_{q_F} : \neg \mathsf{F}(N, x) \tag{83}$$

Unit propagation:

$$\forall y \in \Delta_{q_T}, z \in \Delta_{q_F} : \neg \mathsf{F}(y, z) \tag{84}$$
$$\forall x \in \Delta_q, z \in \Delta_{q_T} : \neg \mathsf{F}(x, N) \lor \mathsf{F}(x, z) \tag{85}$$
$$\forall x, y \in \Delta_q : \neg \mathsf{F}(x, y) \lor \neg \mathsf{F}(y, N) \lor \mathsf{F}(x, N) \tag{86}$$
$$\forall x, y, z \in \Delta_q : \neg \mathsf{F}(x, y) \lor \neg \mathsf{F}(y, z) \lor \mathsf{F}(x, z) \tag{87}$$
$$\forall y \in \Delta_{q_T} : \mathsf{F}(y, N) \tag{88}$$
$$\forall x \in \Delta_{q_F} : \neg \mathsf{F}(x, N) \tag{89}$$
$$\forall x, y \in \Delta_q : \neg \mathsf{F}(x, y) \lor \mathsf{F}(y, x) \tag{90}$$

Shattering:

$$\forall y \in \Delta_{q_T}, z \in \Delta_{q_F} : \neg\mathsf{F}(y,z) \tag{91}$$
$$\forall x \in \Delta_{q_T}, z \in \Delta_{q_T} : \neg\mathsf{F}(x,N) \vee \mathsf{F}(x,z) \tag{92}$$
$$\forall x \in \Delta_{q_F}, z \in \Delta_{q_T} : \neg\mathsf{F}(x,N) \vee \mathsf{F}(x,z) \tag{93}$$
$$\forall x \in \Delta_{q_T}, y \in \Delta_{q_T} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,N) \vee \mathsf{F}(x,N) \tag{94}$$
$$\forall x \in \Delta_{q_T}, y \in \Delta_{q_F} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,N) \vee \mathsf{F}(x,N) \tag{95}$$
$$\forall x \in \Delta_{q_F}, y \in \Delta_{q_T} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,N) \vee \mathsf{F}(x,N) \tag{96}$$
$$\forall x \in \Delta_{q_F}, y \in \Delta_{q_F} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,N) \vee \mathsf{F}(x,N) \tag{97}$$
$$\forall x \in \Delta_{q_T}, y \in \Delta_{q_T}, z \in \Delta_{q_T} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,z) \vee \mathsf{F}(x,z) \tag{98}$$
$$\forall x \in \Delta_{q_T}, y \in \Delta_{q_T}, z \in \Delta_{q_F} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,z) \vee \mathsf{F}(x,z) \tag{99}$$
$$\forall x \in \Delta_{q_T}, y \in \Delta_{q_F}, z \in \Delta_{q_T} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,z) \vee \mathsf{F}(x,z) \tag{100}$$
$$\forall x \in \Delta_{q_T}, y \in \Delta_{q_F}, z \in \Delta_{q_F} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,z) \vee \mathsf{F}(x,z) \tag{101}$$
$$\forall x \in \Delta_{q_F}, y \in \Delta_{q_T}, z \in \Delta_{q_T} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,z) \vee \mathsf{F}(x,z) \tag{102}$$
$$\forall x \in \Delta_{q_F}, y \in \Delta_{q_T}, z \in \Delta_{q_F} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,z) \vee \mathsf{F}(x,z) \tag{103}$$
$$\forall x \in \Delta_{q_F}, y \in \Delta_{q_F}, z \in \Delta_{q_T} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,z) \vee \mathsf{F}(x,z) \tag{104}$$
$$\forall x \in \Delta_{q_F}, y \in \Delta_{q_F}, z \in \Delta_{q_F} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,z) \vee \mathsf{F}(x,z) \tag{105}$$
$$\forall y \in \Delta_{q_T} : \mathsf{F}(y,N) \tag{106}$$
$$\forall x \in \Delta_{q_F} : \neg\mathsf{F}(x,N) \tag{107}$$
$$\forall x \in \Delta_{q_T}, y \in \Delta_{q_T} : \neg\mathsf{F}(x,y) \vee \mathsf{F}(y,x) \tag{108}$$
$$\forall x \in \Delta_{q_T}, y \in \Delta_{q_F} : \neg\mathsf{F}(x,y) \vee \mathsf{F}(y,x) \tag{109}$$
$$\forall x \in \Delta_{q_F}, y \in \Delta_{q_T} : \neg\mathsf{F}(x,y) \vee \mathsf{F}(y,x) \tag{110}$$
$$\forall x \in \Delta_{q_F}, y \in \Delta_{q_F} : \neg\mathsf{F}(x,y) \vee \mathsf{F}(y,x) \tag{111}$$

Unit propagation:

$$\forall y \in \Delta_{q_T}, z \in \Delta_{q_F} : \neg\mathsf{F}(y,z) \tag{112}$$
$$\forall x \in \Delta_{q_T}, z \in \Delta_{q_T} : \mathsf{F}(x,z) \tag{113}$$
$$\forall x \in \Delta_{q_F}, y \in \Delta_{q_T} : \neg\mathsf{F}(x,y) \tag{114}$$
$$\forall x \in \Delta_{q_F}, y \in \Delta_{q_F}, z \in \Delta_{q_F} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,z) \vee \mathsf{F}(x,z) \tag{115}$$
$$\forall y \in \Delta_{q_T} : \mathsf{F}(y,N) \tag{116}$$
$$\forall x \in \Delta_{q_F} : \neg\mathsf{F}(x,N) \tag{117}$$
$$\forall x \in \Delta_{q_F}, y \in \Delta_{q_F} : \neg\mathsf{F}(x,y) \vee \mathsf{F}(y,x) \tag{118}$$

The first, second, third, fifth, and sixth clauses are independent of the other clauses and can be reasoned about separately. They can be trivially lifted. The remaining clauses are:

$$\forall x \in \Delta_{q_F}, y \in \Delta_{q_F}, z \in \Delta_{q_F} : \neg\mathsf{F}(x,y) \vee \neg\mathsf{F}(y,z) \vee \mathsf{F}(x,z) \tag{119}$$
$$\forall x \in \Delta_{q_F}, y \in \Delta_{q_F} : \neg\mathsf{F}(x,y) \vee \mathsf{F}(y,x) \tag{120}$$

The above clauses are an instance of our initial clauses but with smaller domain sizes. We can reason about them by following a similar process, and if we keep sub-results in a cache, the process will be polynomial. □

9.3 Proof of Proposition 3

Proof. After skolemization [30] for removing the existential quantifier, the birthday-paradox theory contains $\forall p \in \Delta_p, \forall d \in \Delta_d : \mathsf{S}(p) \vee \neg\mathsf{Born}(p,d)$, $\forall p \in \Delta_p, d_1, d_2 \in \Delta_d : \neg\mathsf{Born}(p,d_1) \vee \neg\mathsf{Born}(p,d_2)$, and $\forall p_1, p_2 \in \Delta_p, d \in \Delta_d : \neg\mathsf{Born}(p_1,d) \vee \neg\mathsf{Born}(p_2,d)$, where S is the Skolem predicate. This theory is not in FO^2 and not in RU and is not domain-liftable using $\mathcal{R}$. However, this theory is both S^2FO^2 and S^2RU as the last two clauses are instances of $\alpha Born$, the first one is in FO^2 and also in RU and has no PRVs with more than one LV other than Born. Therefore, this theory is domain-liftable based on Theorem 1. □

9.4 Proof of Theorem 1

Proof. The case where $\alpha = \emptyset$ is trivial. Let $\alpha = \alpha(\mathsf{S}_1) \wedge \alpha(\mathsf{S}_2) \wedge \cdots \wedge \alpha(\mathsf{S}_n)$. Once we remove all PRVs having none or one LV by (lifted) case-analysis, the remaining clauses can be divided into $n+1$ components: the i-th component in the first n components only contains S_i literals, and the $(n+1)$-th component contains no S_i literals. These components are disconnected from each other, so we can consider each of them separately. The $(n+1)$-th component comes from clauses in β and is domain-liftable by definition. The following two Lemmas prove that the clauses in the other components are also domain-liftable. The proofs of both lemmas rely on domain recursion.

Lemma 1. *A clausal theory $\alpha(\mathsf{S})$ with only one predicate S where all clauses have exactly two different literals of S is domain-liftable.*

Lemma 2. *Suppose $\{\Delta_{p_1}, \Delta_{p_2}, \ldots, \Delta_{p_n}\}$ are mutually exclusive subsets of Δ_x and $\{\Delta_{q_1}, \Delta_{q_2}, \ldots, \Delta_{q_m}\}$ are mutually exclusive subsets of Δ_y. We can add any unit clause of the form $\forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_j} : \mathsf{S}(p_i, q_j)$ or $\forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_j} : \neg\mathsf{S}(p_i, q_j)$ to the $\alpha(\mathsf{S})$ theory in Lemma 1 and the theory is still domain-liftable.*

Therefore, theories in S^2FO^2 and S^2RU are domain-liftable. □

9.5 Proof of Lemma 1

Proof. A theory in this form has a *subset* of the following clauses:

$$\forall x \in \Delta_x, y_1, y_2 \in \Delta_y : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{121}$$
$$\forall x \in \Delta_x, y_1, y_2 \in \Delta_y : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{122}$$
$$\forall x \in \Delta_x, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{123}$$
$$\forall x_1, x_2 \in \Delta_x, y \in \Delta_y : \mathsf{S}(x_1, y) \vee \mathsf{S}(x_2, y) \tag{124}$$
$$\forall x_1, x_2 \in \Delta_x, y \in \Delta_y : \mathsf{S}(x_1, y) \vee \neg\mathsf{S}(x_2, y) \tag{125}$$
$$\forall x_1, x_2 \in \Delta_x, y \in \Delta_y : \neg\mathsf{S}(x_1, y) \vee \neg\mathsf{S}(x_2, y) \tag{126}$$
$$\forall x_1, x_2 \in \Delta_x, y_1, y_2 \in \Delta_y : \mathsf{S}(x_1, y_1) \vee \mathsf{S}(x_2, y_2) \tag{127}$$
$$\forall x_1, x_2 \in \Delta_x, y_1, y_2 \in \Delta_y : \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_2, y_2) \tag{128}$$
$$\forall x_1, x_2 \in \Delta_x, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_2, y_2) \tag{129}$$

Let N be an individual in Δ_x. Applying domain recursion on $\Delta_{x'} = \Delta_{x'} - \{N\}$ for all clauses gives:

for (1):

$$\forall y_1, y_2 \in \Delta_y : \mathsf{S}(N, y_1) \vee \mathsf{S}(N, y_2) \tag{130}$$
$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{131}$$

for (2):

$$\forall y_1, y_2 \in \Delta_y : \mathsf{S}(N, y_1) \vee \neg\mathsf{S}(N, y_2) \tag{132}$$
$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{133}$$

for (3):

$$\forall y_1, y_2 \in \Delta_y : \neg\mathsf{S}(N, y_1) \vee \neg\mathsf{S}(N, y_2) \tag{134}$$
$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{135}$$

for (4):

$$\forall x \in \Delta_{x'}, y \in \Delta_y : \mathsf{S}(N, y) \vee \mathsf{S}(x, y) \tag{136}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_y : \mathsf{S}(x_1, y) \vee \mathsf{S}(x_2, y) \tag{137}$$

for (5):

$$\forall x \in \Delta_{x'}, y \in \Delta_y : \mathsf{S}(N, y) \vee \neg\mathsf{S}(x, y) \tag{138}$$
$$\forall x \in \Delta_{x'}, y \in \Delta_y : \neg\mathsf{S}(N, y) \vee \mathsf{S}(x, y) \tag{139}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_y : \mathsf{S}(x_1, y) \vee \neg\mathsf{S}(x_2, y) \tag{140}$$

for (6):

$$\forall x \in \Delta_{x'}, y \in \Delta_y : \neg\mathsf{S}(N, y) \vee \neg\mathsf{S}(x, y) \tag{141}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_y : \neg\mathsf{S}(x_1, y) \vee \neg\mathsf{S}(x_2, y) \tag{142}$$

for (7):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(N, y_1) \vee \mathsf{S}(x, y_2) \tag{143}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x_1, y_1) \vee \mathsf{S}(x_2, y_2) \tag{144}$$

for (8):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(N, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{145}$$
$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(N, y_1) \vee \mathsf{S}(x, y_2) \tag{146}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_2, y_2) \tag{147}$$

for (9):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(N, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{148}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_2, y_2) \tag{149}$$

Then we can perform lifted case-analysis on $\mathsf{S}(N, y)$. For the case where $\mathsf{S}(N, y)$ is true for exactly k of the individuals in Δ_y, we update all clauses assuming Δ_{y_T} and Δ_{y_F} represent the individuals for which $\mathsf{S}(N, y)$ is True and False respectively, and assuming $\forall y \in \Delta_{y_T} : \mathsf{S}(N, y)$ and $\forall y \in \Delta_{y_F} : \neg\mathsf{S}(N, y)$:

for (1):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{150}$$

for (2):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{151}$$

for (3):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{152}$$

for (4):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \mathsf{S}(x, y) \tag{153}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_y : \mathsf{S}(x_1, y) \vee \mathsf{S}(x_2, y) \tag{154}$$

for (5):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \neg\mathsf{S}(x, y) \tag{155}$$
$$\forall x \in \Delta_{x'}, y \in \Delta_{y_T} : \mathsf{S}(x, y) \tag{156}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_y : \mathsf{S}(x_1, y) \vee \neg\mathsf{S}(x_2, y) \tag{157}$$

for (6):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_T} : \neg\mathsf{S}(x, y) \tag{158}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_y : \neg\mathsf{S}(x_1, y) \vee \neg\mathsf{S}(x_2, y) \tag{159}$$

for (7):

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_y : \mathsf{S}(x, y_2) \tag{160}$$

$$\forall x_1, x_2 \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x_1, y_1) \vee \mathsf{S}(x_2, y_2) \tag{161}$$

for (8):

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_y : \neg\mathsf{S}(x, y_2) \tag{162}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_y : \mathsf{S}(x, y_2) \tag{163}$$

$$\forall x_1, x_2 \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_2, y_2) \tag{164}$$

$$\tag{165}$$

for (9):

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_y : \neg\mathsf{S}(x, y_2) \tag{166}$$

$$\forall x_1, x_2 \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_2, y_2) \tag{167}$$

After subsumptions and shattering:

for (1):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{168}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{169}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{170}$$

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{171}$$

$$\tag{172}$$

for (2):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{173}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{174}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{175}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{176}$$

for (3):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_T} : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{177}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_F} : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{178}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{179}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_F} : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{180}$$

for (4): (For the second clause, the case where $y \in \Delta_{y_F}$ becomes subsumed by the first clause)

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \mathsf{S}(x, y) \tag{181}$$

$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_{y_T} : \mathsf{S}(x_1, y) \vee \mathsf{S}(x_2, y) \tag{182}$$

for (5): (The third clause was subsumed by the first two)

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \neg\mathsf{S}(x, y) \tag{183}$$

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_T} : \mathsf{S}(x, y) \tag{184}$$

for (6): (Similar to (4))

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_T} : \neg\mathsf{S}(x, y) \tag{185}$$

$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_{y_F} : \neg\mathsf{S}(x_1, y) \vee \neg\mathsf{S}(x_2, y) \tag{186}$$

for (7):

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_2) \tag{187}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_2) \tag{188}$$

$$\forall x_1, x_2 \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_T} : \mathsf{S}(x_1, y_1) \vee \mathsf{S}(x_2, y_2) \tag{189}$$

$$\forall x_1, x_2 \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_F} : \mathsf{S}(x_1, y_1) \vee \mathsf{S}(x_2, y_2) \tag{190}$$

for (8):

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} : \neg\mathsf{S}(x, y_2) \tag{191}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_F} : \neg\mathsf{S}(x, y_2) \tag{192}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_2) \tag{193}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_2) \tag{194}$$

for (9):

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_T} : \neg\mathsf{S}(x, y_2) \tag{195}$$

$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_F} : \neg\mathsf{S}(x, y_2) \tag{196}$$

$$\forall x_1, x_2 \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} : \neg\mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_2, y_2) \tag{197}$$

$$\forall x_1, x_2 \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_F} : \neg\mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_2, y_2) \tag{198}$$

Looking at the first clause for (7) (and some other clauses), we see that there exists a $\forall y_1 \in \Delta_{y_T}$ but y_1 does not appear in the formula. If $\Delta_{y_T} = \emptyset$, we can ignore this clause. Otherwise, we can ignore $\forall y_1 \in \Delta_{y_T}$. So we consider three cases. When $k = 0$ (i.e. $\Delta_{y_F} = \Delta_y$, $\Delta_{y_T} = \emptyset$):

for (1):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{199}$$

for (2):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{200}$$

for (3):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{201}$$

for (4):

$$\forall x \in \Delta_{x'}, y \in \Delta_y : \mathsf{S}(x, y) \tag{202}$$

for (5):

$$\forall x \in \Delta_{x'}, y \in \Delta_y : \neg\mathsf{S}(x, y) \tag{203}$$

for (6):

$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_y : \neg\mathsf{S}(x_1, y) \vee \neg\mathsf{S}(x_2, y) \tag{204}$$

for (7):

$$\forall x \in \Delta_{x'}, y_2 \in \Delta_y : \mathsf{S}(x, y_2) \tag{205}$$

for (8):

$$\forall x \in \Delta_{x'}, y_2 \in \Delta_y : \neg\mathsf{S}(x, y_2) \tag{206}$$

for (9):

$$\forall x_1, x_2 \in \Delta_{x'}, y_1 \in \Delta_y, y_2 \in \Delta_y : \neg\mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_2, y_2) \tag{207}$$

If clause #4 is one of the clauses in the theory, then unit propagation either gives False, or satisfies all the clauses. The same is true for clauses #5, #7, and #8. In a theory not having any of these four clauses, we will be left with a set of clauses that are again a subset of the initial 9 clauses that we started with, but with a smaller domain size. By applying the same procedure, we can count the number of models. When $k = |\Delta_y|$ (i.e. $\Delta_{y_F} = \emptyset$, $\Delta_{y_T} = \Delta_y$), everything is just similar to the $k = 0$ case.

When $0 < k < |\Delta_y|$ (i.e. neither Δ_{y_T} nor Δ_{y_F} are empty):

for (1):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{208}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{209}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{210}$$
$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{211}$$

for (2):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{212}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{213}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{214}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{215}$$

for (3):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_T} : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{216}$$
$$\forall x \in \Delta_{x'}, y_1 \Delta_{y_T}, y_2 \in \Delta_{y_F} : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{217}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{218}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_F} : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{219}$$

for (4):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \mathsf{S}(x, y) \tag{220}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_{y_T} : \mathsf{S}(x_1, y) \vee \mathsf{S}(x_2, y) \tag{221}$$

for (5):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \neg\mathsf{S}(x, y) \tag{222}$$
$$\forall x \in \Delta_{x'}, y \in \Delta_{y_T} : \mathsf{S}(x, y) \tag{223}$$

for (6):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_T} : \neg\mathsf{S}(x, y) \tag{224}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_{y_F} : \neg\mathsf{S}(x_1, y) \vee \neg\mathsf{S}(x_2, y) \tag{225}$$

for (7):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_T} : \mathsf{S}(x, y) \tag{226}$$
$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \mathsf{S}(x, y) \tag{227}$$

for (8):

$$False \tag{228}$$

for (9):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_T} : \neg\mathsf{S}(x, y) \tag{229}$$
$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \neg\mathsf{S}(x, y) \tag{230}$$

If either one of clauses #5, #7, #8 or #9 are in the theory, then unit propagation either gives $False$ or satisfies all clauses. Assume none of these four clauses are in the theory. If both clauses #4 and #6 are in a theory, again unit propagation gives either $False$ or satisfies all clauses. If none of them are in the theory, then the other clauses are a subset of the initial 9 clauses that we started with. So let's consider the case where we have clause #4 and a subset of the first three clauses (the case with #6 instead of #4 is similar). In this case, if clauses #2 or #3 are in the theory, unit propagation either gives $False$ or satisfies all the clauses. If none of them are in the theory and only #1 is in the theory, we will have the following clauses after unit propagation:

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \mathsf{S}(x, y) \tag{231}$$

$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_{y_T} : \mathsf{S}(x_1, y) \vee \mathsf{S}(x_2, y) \tag{232}$$

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{233}$$

The first clause is independent of the other two clauses. The second and third clauses are just similar to clauses #4 and #1 in the initial list of clauses and we can handle them using the same procedure.

If we use a cache to store computations for all subproblems, WFOMC is domain-liftable, i.e. polynomial in the population sizes. □

9.6 Proof of Lemma 2

Proof. Let ψ be the set of pairs (i, j) such that the singleton clause $\forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_j} : \mathsf{S}(p_i, q_j)$ is in the theory, and $\overline{\psi}$ be the set of pairs (i, j) such that the singleton clause $\forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_j} : \neg\mathsf{S}(p_i, q_j)$ is in the theory. Then the singleton clauses can be written as follows:

$$\forall (i, j) \in \psi : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_j} : \mathsf{S}(p_i, q_j) \tag{234}$$

$$\forall (i, j) \in \overline{\psi} : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_j} : \neg\mathsf{S}(p_i, q_j) \tag{235}$$

And the $\alpha(\mathsf{S})$ clauses are as in Lemma 1. Without loss of generality, let's assume we select an individual $N \in \Delta_{p_1}$ for domain recursion, and re-write all clauses to separate N from Δ_{p_1}. Assuming $\Delta_{p'_1}, = \Delta_{p_1} - \{N\}$ and $\Delta_{x'} = \Delta_x - \{N\}$, the theory will be:

For singletons:

$$\forall (1, j) \in \psi : \forall q_j \in \Delta_{q_j} : \mathsf{S}(N, q_j) \tag{236}$$

$$\forall (1, j) \in \overline{\psi} : \forall q_j \in \Delta_{q_j} : \neg\mathsf{S}(N, q_j) \tag{237}$$

$$\forall (1, j) \in \psi : \forall p'_1 \in \Delta_{p'_1}, , q_j \in \Delta_{q_j} : \mathsf{S}(p'_1, q_j) \tag{238}$$

$$\forall (1, j) \in \overline{\psi} : \forall p'_1 \in \Delta_{p'_1}, , q_j \in \Delta_{q_j} : \neg\mathsf{S}(p'_1, q_j) \tag{239}$$

$$\forall (i \neq 1, j) \in \psi : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_j} : \mathsf{S}(p_i, q_j) \tag{240}$$

$$\forall (i \neq 1, j) \in \overline{\psi} : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_j} : \neg\mathsf{S}(p_i, q_j) \tag{241}$$

for (1):

$$\forall y_1, y_2 \in \Delta_y : \mathsf{S}(N, y_1) \vee \mathsf{S}(N, y_2) \tag{242}$$

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{243}$$

for (2):

$$\forall y_1, y_2 \in \Delta_y : \mathsf{S}(N, y_1) \vee \neg\mathsf{S}(N, y_2) \tag{244}$$

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{245}$$

for (3):

$$\forall y_1, y_2 \in \Delta_y : \neg\mathsf{S}(N, y_1) \vee \neg\mathsf{S}(N, y_2) \tag{246}$$

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{247}$$

for (4):

$$\forall x \in \Delta_{x'}, y \in \Delta_y : \mathsf{S}(N, y) \vee \mathsf{S}(x, y) \tag{248}$$

$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_y : \mathsf{S}(x_1, y) \vee \mathsf{S}(x_2, y) \tag{249}$$

for (5):

$$\forall x \in \Delta_{x'}, y \in \Delta_y : \mathsf{S}(N, y) \vee \neg\mathsf{S}(x, y) \tag{250}$$
$$\forall x \in \Delta_{x'}, y \in \Delta_y : \neg\mathsf{S}(N, y) \vee \mathsf{S}(x, y) \tag{251}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_y : \mathsf{S}(x_1, y) \vee \neg\mathsf{S}(x_2, y) \tag{252}$$

for (6):

$$\forall x \in \Delta_{x'}, y \in \Delta_y : \neg\mathsf{S}(N, y) \vee \neg\mathsf{S}(x, y) \tag{253}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_y : \neg\mathsf{S}(x_1, y) \vee \neg\mathsf{S}(x_2, y) \tag{254}$$

for (7):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(N, y_1) \vee \mathsf{S}(x, y_2) \tag{255}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x_1, y_1) \vee \mathsf{S}(x_2, y_2) \tag{256}$$

for (8):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(N, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{257}$$
$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(N, y_1) \vee \mathsf{S}(x, y_2) \tag{258}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_2, y_2) \tag{259}$$

for (9):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(N, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{260}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(x_1, y_1) \vee \neg\mathsf{S}(x_2, y_2) \tag{261}$$

We apply lifted case-analysis on each $\mathsf{S}(N, \Delta_{q_j})$. For each j, let $\Delta_{q_{Tj}}$ represent the individuals in Δ_{q_j} for which $\mathsf{S}(N, q_j)$ is true and $\Delta_{q_{Fj}}$ be the other individuals. For each j, lifted case-analysis adds two clauses to the theory as follows:

$$\forall q_j \in \Delta_{q_{Tj}} : \mathsf{S}(N, q_j) \tag{262}$$
$$\forall q_j \in \Delta_{q_{Fj}} : \neg\mathsf{S}(N, q_j) \tag{263}$$

We shatter all other singleton clauses based on these newly added singletons. If the singletons are inconsistent, there is no model. Otherwise, let y_T represent $\cup_j q_{Tj}$ and y_F represent $\cup_j q_{Fj}$. We add the following two singleton clauses to the theory:

$$\forall y \in \Delta_{y_T} : \mathsf{S}(N, y) \tag{264}$$
$$\forall y \in \Delta_{y_F} : \neg\mathsf{S}(N, y) \tag{265}$$

We shatter all clauses in $\alpha(\mathsf{S})$ form based on these two singletons (not considering the shattering caused by the other singletons) and apply unit propagation. Then the theory will be as follows (the details can be checked in Lemma 1. Here we only consider the case where $y_T \neq \emptyset$ and $y_F \neq \emptyset$; the case where one of them is empty can be considered similarly as in Lemma 1):

For singletons:

$$\forall j : \forall q_j \in \Delta_{q_{Tj}} : \mathsf{S}(N, q_j) \tag{266}$$
$$\forall j : \forall q_j \in \Delta_{q_{Fj}} : \neg\mathsf{S}(N, q_j) \tag{267}$$
$$\forall (1, j) \in \psi : \forall p'_1 \in \Delta_{p'_1}, q_j \in \Delta_{q_{Tj}} : \mathsf{S}(p'_1, q_j) \tag{268}$$
$$\forall (1, j) \in \psi : \forall p'_1 \in \Delta_{p'_1}, q_j \in \Delta_{q_{Fj}} : \mathsf{S}(p'_1, q_j) \tag{269}$$
$$\forall (1, j) \in \overline{\psi} : \forall p'_1 \in \Delta_{p'_1}, q_j \in \Delta_{q_{Tj}} : \neg\mathsf{S}(p'_1, q_j) \tag{270}$$
$$\forall (1, j) \in \overline{\psi} : \forall p'_1 \in \Delta_{p'_1}, q_j \in \Delta_{q_{Fj}} : \neg\mathsf{S}(p'_1, q_j) \tag{271}$$
$$\forall (i \neq 1, j) \in \psi : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_{Tj}} : \mathsf{S}(p_i, q_j) \tag{272}$$
$$\forall (i \neq 1, j) \in \psi : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_{Fj}} : \mathsf{S}(p_i, q_j) \tag{273}$$
$$\forall (i \neq 1, j) \in \overline{\psi} : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_{Tj}} : \neg\mathsf{S}(p_i, q_j) \tag{274}$$
$$\forall (i \neq 1, j) \in \overline{\psi} : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_{Fj}} : \neg\mathsf{S}(p_i, q_j) \tag{275}$$

The two singletons on Δ_{y_F} and Δ_{y_T}:

$$\forall y \in \Delta_{y_T} : \mathsf{S}(N, y) \tag{276}$$
$$\forall y \in \Delta_{y_F} : \neg\mathsf{S}(N, y) \tag{277}$$

for (1):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{278}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{279}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{280}$$
$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{281}$$

for (2):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{282}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_T}, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{283}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{284}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_F} : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{285}$$

for (3):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_T} : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{286}$$
$$\forall x \in \Delta_{x'}, y_1 \Delta_{y_T}, y_2 \in \Delta_{y_F} : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{287}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_T} : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{288}$$
$$\forall x \in \Delta_{x'}, y_1 \in \Delta_{y_F}, y_2 \in \Delta_{y_F} : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{289}$$

for (4):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \mathsf{S}(x, y) \tag{290}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_{y_T} : \mathsf{S}(x_1, y) \vee \mathsf{S}(x_2, y) \tag{291}$$

for (5):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \neg\mathsf{S}(x, y) \tag{292}$$
$$\forall x \in \Delta_{x'}, y \in \Delta_{y_T} : \mathsf{S}(x, y) \tag{293}$$

for (6):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_T} : \neg\mathsf{S}(x, y) \tag{294}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_{y_F} : \neg\mathsf{S}(x_1, y) \vee \neg\mathsf{S}(x_2, y) \tag{295}$$

for (7):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_T} : \mathsf{S}(x, y) \tag{296}$$
$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \mathsf{S}(x, y) \tag{297}$$

for (8):

$$False \tag{298}$$

for (9):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_T} : \neg\mathsf{S}(x, y) \tag{299}$$
$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \neg\mathsf{S}(x, y) \tag{300}$$

Clauses number 266, 267, 276 and 277 are disconnected from the rest of the theory and can be reasoned about separately. It is trivial to lift these clauses. Now let's consider the other clauses.

If either one of the clauses #5, #7, #8. or #9 are in the theory, then unit propagation either gives false or satisfies all clauses. The same is true when both #4 and #6 are in the theory. If neither #4 nor #6 are in the theory, then we can conjoin the individuals in Δ_{y_T} and Δ_{y_F} as well as those in $\Delta_{q_{jT}}$ and $\Delta_{q_{jF}}$ and write the theory as follows:
For singletons:

$$\forall (1,j) \in \psi : \forall p'_1 \in \Delta_{p'_1}, q_j \in \Delta_{q_j} : \mathsf{S}(p'_1, q_j) \tag{301}$$

$$\forall (1,j) \in \overline{\psi} : \forall p'_1 \in \Delta_{p'_1}, q_j \in \Delta_{q_j} : \neg\mathsf{S}(p'_1, q_j) \tag{302}$$

$$\forall (i \neq 1,j) \in \psi : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_j} : \mathsf{S}(p_i, q_j) \tag{303}$$

$$\forall (i \neq 1,j) \in \overline{\psi} : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_j} : \neg\mathsf{S}(p_i, q_j) \tag{304}$$

for (1):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{305}$$

for (2):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{306}$$

for (3):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_y : \neg\mathsf{S}(x, y_1) \vee \neg\mathsf{S}(x, y_2) \tag{307}$$

This theory is an instance of our initial theory, but with p_1 having a smaller domain size. So we can continue this process recursively on the remaining clauses.

Now let's consider the case where #4 is in the theory but #6 is not (the case where #6 is in the theory and #4 is not is similar). In this case, if #2 or #3 are in the theory, then unit propagation either gives false or satisfies all the clauses. If #4 and #1 are in the theory, then the theory is as follows:

For singletons:

$$\forall (1,j) \in \psi : \forall p'_1 \in \Delta_{p'_1}, q_j \in \Delta_{q_{Tj}} : \mathsf{S}(p'_1, q_j) \tag{308}$$

$$\forall (1,j) \in \psi : \forall p'_1 \in \Delta_{p'_1}, q_j \in \Delta_{q_{Fj}} : \mathsf{S}(p'_1, q_j) \tag{309}$$

$$\forall (1,j) \in \overline{\psi} : \forall p'_1 \in \Delta_{p'_1}, q_j \in \Delta_{q_{Tj}} : \neg\mathsf{S}(p'_1, q_j) \tag{310}$$

$$\forall (1,j) \in \overline{\psi} : \forall p'_1 \in \Delta_{p'_1}, q_j \in \Delta_{q_{Fj}} : \neg\mathsf{S}(p'_1, q_j) \tag{311}$$

$$\forall (i \neq 1,j) \in \psi : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_{Tj}} : \mathsf{S}(p_i, q_j) \tag{312}$$

$$\forall (i \neq 1,j) \in \psi : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_{Fj}} : \mathsf{S}(p_i, q_j) \tag{313}$$

$$\forall (i \neq 1,j) \in \overline{\psi} : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_{Tj}} : \neg\mathsf{S}(p_i, q_j) \tag{314}$$

$$\forall (i \neq 1,j) \in \overline{\psi} : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_{Fj}} : \neg\mathsf{S}(p_i, q_j) \tag{315}$$

$$\tag{316}$$

for (1):

$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{317}$$

for (4):

$$\forall x \in \Delta_{x'}, y \in \Delta_{y_F} : \mathsf{S}(x, y) \tag{318}$$

$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_{y_T} : \mathsf{S}(x_1, y) \vee \mathsf{S}(x_2, y) \tag{319}$$

Clause number 318 and the singleton clauses having $q_j \in \Delta_{q_{Fj}}$ are disconnected from the rest of the theory and can reasoned about separately. They can be trivially lifted. Once we remove these clauses, the theory will be as follows:

$$\forall (1,j) \in \psi : \forall p'_1 \in \Delta_{p'_1}, q_j \in \Delta_{q_{Tj}} : \mathsf{S}(p'_1, q_j) \tag{320}$$
$$\forall (1,j) \in \overline{\psi} : \forall p'_1 \in \Delta_{p'_1}, q_j \in \Delta_{q_{Tj}} : \neg\mathsf{S}(p'_1, q_j) \tag{321}$$
$$\forall (i \neq 1,j) \in \psi : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_{Tj}} : \mathsf{S}(p_i, q_j) \tag{322}$$
$$\forall (i \neq 1,j) \in \overline{\psi} : \forall p_i \in \Delta_{p_i}, q_j \in \Delta_{q_{Tj}} : \neg\mathsf{S}(p_i, q_j) \tag{323}$$
$$\forall x \in \Delta_{x'}, y_1, y_2 \in \Delta_{y_T} : \mathsf{S}(x, y_1) \vee \mathsf{S}(x, y_2) \tag{324}$$
$$\forall x_1, x_2 \in \Delta_{x'}, y \in \Delta_{y_T} : \mathsf{S}(x_1, y) \vee \mathsf{S}(x_2, y) \tag{325}$$

which is an instance of our initial theory, but where Δ_{q_j}s have smaller domain sizes. So we can continue this process recursively on the remaining clauses.

We showed that in all cases, after domain recursion we will have an instance of our initial theory again, but with smaller domain sizes. By keeping the WFOMC of sub-problems in a cache, the whole process will be domain-liftable: i.e. polynomial in the population sizes. □

9.7 Proof of Proposition 4

Proof. Let $T \in FO^2$ and T' be any of the theories resulting from exhaustively applying rules in $\mathcal{R}$ expect lifted case-analysis on T. If T initially contains a unary PRV with predicate S, either it is still unary in T' or lifted decomposition has replaced the LV with a constant. In the first case, we can follow a generic branch of lifted case-analysis on S, and in the second case, either T' is empty or all binary PRVs in T have become unary in T' due to applying the lifted decomposition and we can follow a generic branch of lifted case-analysis for any of these PRVs. The generic branch in both cases is in FO^2 and the same procedure can be followed until all theories become empty. If T initially contains only binary PRVs, lifted decomposition applies as the grounding of T is disconnected for each pair of individuals, and after lifted decomposition all PRVs have no LVs. Applying case analysis on all PRVs gives empty theories. Therefore, $T \in RU$. The theory $\forall x, y, z \in \Delta_p : \mathsf{F}(x, y) \vee \mathsf{F}(y, z) \vee \mathsf{F}(x, y, z)$ is an example of a *RU* theory that is not in FO^2, showing $RU \not\subset FO^2$. FO^2 and RU are special cases of S^2FO^2 and S^2RU respectively, where $\alpha = \emptyset$, showing $FO^2 \subset S^2FO^2$ and $RU \subset S^2RU$. However, Example 3 is both in S^2FO^2 and S^2RU but is not in FO^2 and not in *RU*, showing $S^2FO^2 \not\subset FO^2$ and $S^2RU \not\subset RU$. Since $FO^2 \subset RU$ and the class of added $\alpha(\mathsf{S})$ clauses are the same, $S^2FO^2 \subset S^2RU$. □

References

[1] Babak Ahmadi, Kristian Kersting, and Sriraam Natarajan. Lifted online training of relational models with stochastic gradient methods. In *ECML PKDD*, pages 585–600, 2012.

[2] W. W. Rouse Ball. Other questions on probability. *Mathematical Recreations and Essays*, page 45, 1960.

[3] Paul Beame, Guy Van den Broeck, Eric Gribkoff, and Dan Suciu. Symmetric weighted first-order model counting. In *PODS*, pages 313–328, 2015.

[4] Hung Hai Bui, Tuyen N Huynh, Artificial Intelligence Center, and Sebastian Riedel. Automorphism groups of graphical models and lifted variational inference. In *UAI*, page 132, 2013.

[5] Jaesik Choi, Rodrigo de Salvo Braz, and Hung H. Bui. Efficient methods for lifted inference with aggregate factors. In *AAAI*, 2011.

[6] Nilesh Dalvi and Dan Suciu. Efficient query evaluation on probabilistic databases. *The VLDB Journal*, 16(4):523–544, 2007.

[7] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. ProbLog: A probabilistic Prolog and its application in link discovery. In *IJCAI*, volume 7, 2007.

[8] Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. Statistical relational artificial intelligence: Logic, probability, and computation. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 10(2):1–189, 2016.

[9] Rodrigo de Salvo Braz, Eyal Amir, and Dan Roth. Lifted first-order probabilistic inference. In *IJCAI*, pages 1319–1325, 2005.

[10] Lise Getoor and Ben Taskar. *Introduction to statistical relational learning*. MIT press, 2007.

[11] Vibhav Gogate and Pedro Domingos. Probabilistic theorem proving. In *UAI*, pages 256–265, 2011.

[12] Manfred Jaeger. Relational Bayesian networks. In *UAI*. Morgan Kaufmann Publishers Inc., 1997.

[13] Yacine Jernite, Alexander M Rush, and David Sontag. A fast variational approach for learning Markov random field language models. In *ICML*, 2015.

[14] Abhay Jha, Vibhav Gogate, Alexandra Meliou, and Dan Suciu. Lifted inference seen from the other side: The tractable features. In *NIPS*, pages 973–981, 2010.

[15] Seyed Mehran Kazemi and David Poole. Knowledge compilation for lifted probabilistic inference: Compiling to a low-level language. In *KR*, 2016.

[16] Seyed Mehran Kazemi and David Poole. Why is compiling lifted inference into a low-level language so effective? *arXiv preprint arXiv:1606.04512*, 2016.

[17] Kristian Kersting, Babak Ahmadi, and Sriraam Natarajan. Counting belief propagation. In *UAI*, pages 277–284, 2009.

[18] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, 2009.

[19] Timothy Kopp, Parag Singla, and Henry Kautz. Lifted symmetry detection and breaking for MAP inference. In *NIPS*, pages 1315–1323, 2015.

[20] Brian Milch, Luke S. Zettlemoyer, Kristian Kersting, Michael Haimes, and Leslie Pack Kaelbling. Lifted probabilistic inference with counting formulae. In *AAAI*, pages 1062–1068, 2008.

[21] Mathias Niepert. Markov chains on orbits of permutation groups. In *UAI*, 2012.

[22] David Poole, Fahiem Bacchus, and Jacek Kisynski. Towards completely lifted search-based probabilistic inference. *arXiv:1107.4035 [cs.AI]*, 2011.

[23] David Poole. First-order probabilistic inference. In *IJCAI*, pages 985–991, 2003.

[24] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.

[25] Parag Singla and Pedro M Domingos. Lifted first-order belief propagation. In *AAAI*, volume 8, pages 1094–1099, 2008.

[26] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. Probabilistic databases. *Synthesis Lectures on Data Management*, 3(2):1–180, 2011.

[27] Nima Taghipour, Daan Fierens, Guy Van den Broeck, Jesse Davis, and Hendrik Blockeel. Completeness results for lifted variable elimination. In *AISTATS*, pages 572–580, 2013.

[28] Guy Van den Broeck, Nima Taghipour, Wannes Meert, Jesse Davis, and Luc De Raedt. Lifted probabilistic inference by first-order knowledge compilation. In *IJCAI*, pages 2178–2185, 2011.

[29] Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Lifted relax, compensate and then recover: From approximate to exact lifted probabilistic inference. In *UAI*, 2012.

[30] Guy Van den Broeck, Wannes Meert, and Adnan Darwiche. Skolemization for weighted first-order model counting. In *KR*, 2014.

[31] Guy Van den Broeck. On the completeness of first-order knowledge compilation for lifted probabilistic inference. In *NIPS*, pages 1386–1394, 2011.

[32] Jan Van Haaren, Guy Van den Broeck, Wannes Meert, and Jesse Davis. Lifted generative learning of Markov logic networks. *Machine Learning*, pages 1–29, 2015.

[33] Deepak Venugopal and Vibhav Gogate. Evidence-based clustering for scalable inference in Markov logic. In *ECML PKDD*, pages 258–273, 2014.

[34] Deepak Venugopal and Vibhav G Gogate. Scaling-up importance sampling for Markov logic networks. In *NIPS*, pages 2978–2986, 2014.