Weighted Model Counting With Function Symbols*

Vaishak Belle
University of Edinburgh
United Kingdom
vaishak@ed.ac.uk

Abstract

Probabilistic relational languages lift the syntax of relational logic for the specification of large-scale probabilistic graphical models, often admitting concise descriptions for interacting random variables over classes, hierarchies and constraints. The emergence of weighted model counting as an effective and general approach to probabilistic inference has further allowed practitioners to reason about heterogeneous representations, such as Markov logic networks and ProbLog programs, by encoding them as a logical theory. However, much of this work has been limited to an essentially propositional setting: the logical model is understood in terms of ground formulas over a fixed and finite domain; no infinite domains, and certainly no function symbols (other than constants). On the one hand, this is not surprising, because such features are very problematic from a decidability viewpoint, but on the other, they turn out to be very attractive from the point of view of machine learning applications when there is uncertainty about the existence and identity of objects. In this paper, we reconsider the problem of probabilistic reasoning in a logical language with function symbols, and establish some key results that permit effective algorithms.

1 INTRODUCTION

Unifying logic and probability is the one of the long-standing goals of AI [Russell, 2015]. Early efforts attempted the design of general-purpose representation languages [Bacchus, 1990], but were found to be too expressive as they fell off the computational cliff. As a consequence, more modest proposals, such as *probabilistic relational languages*, were developed that lift the syntax of finite-domain relational logic for the specification of large-scale probabilistic graphical models. These often admit concise descriptions for interacting random variables over classes, hierarchies and constraints [Getoor and Taskar, 2007, Wu et al., 2012]. The emergence of *weighted model counting* (WMC) as an effective and general approach to probabilistic inference has further allowed practitioners to reason about heterogeneous representations, such as Markov logic networks and ProbLog programs [Richardson and Domingos, 2006, Fierens et al., 2011], by encoding them as a logical theory. Indeed, the decision versions of both model counting and Bayesian inference are #P-complete, and there are polynomial-time reductions from each problem to the other [Bacchus et al., 2009]. The last decade has seen significant progress in exact and approximate WMC technology [Chavira and Darwiche, 2008, Gogate and Domingos, 2011, Ermon et al., 2013], which enable deterministic reasoning and exploit context-specific independencies.

In so much as the representation language determines the *expressiveness* of the probabilistic framework, however, much of this work has been limited to an essentially propositional setting. (Notable exceptions will be discussed in the final section.) That is, the logical model is understood in terms of ground formulas over a fixed and finite domain for a relational vocabulary; no infinite domains, and certainly no function symbols (other than constants). But these latter features turn out to be very attractive from the point of view of statistical experiments and machine learning pipelines, especially when there is uncertainty about the existence and identity of objects. Consider, for example, a *target tracking* application where we observe a 171 cm tall female individual believed to be James Bond's fiancee [Srivastava et al., 2014]. This can be expressed as:

$$\exists x\,(\mathit{scanned}(x) \wedge \mathit{height}(x) = 171 \wedge \mathit{isFemale}(x) \wedge \mathit{fiancee}(\mathit{james}) = x) \quad (1)$$

which uses an existential for the unknown individual, and functions and relations to express her properties. Perhaps more significantly, functions can be used to express classes and hierarchies exactly as relations would

* The author is grateful to Gerhard Lakemeyer for discussions on the compactness property.

allow, but can additionally enable equational reasoning over complex terms. For example, suppose we are interested in *knowledge base completion* over incomplete data [Nickel et al., 2016]. Consider the sentence:

$$\begin{aligned} &advisor(advisor(mary)) \neq john \wedge \\ &fiancee(advisor(mary)) = beth \wedge \\ &advisor(mary) = adam \vee advisor(mary) = dave \end{aligned} \tag{2}$$

Although we do not know who *mary*'s advisor is, learning about *beth*'s betrothed, or otherwise evaluating who among $\{adam, dave\}$ are not *john*'s academic descendants may allow us to infer just that.

Unfortunately, functions are very problematic from a decidability viewpoint [Boerger et al., 1997], with many elementary cases involving function symbols leading to undecidable properties. The hope then is to identify reasonable fragments that suffice for practical purposes. Most of the well-studied fragments [Boerger et al., 1997] seem to have little relevance to the encodings considered for graphical models. To that end, there have been some attempts that allow function symbols in probabilistic programming languages. Distribution semantics [Sato, 1995] underlies languages like ProbLog [Fierens et al., 2011], which in principle support proofs (chains of inference) of infinite length. For example, the following instantiates the set of natural numbers:

$$nat(0) \wedge \forall x(nat(x) \supset nat(next(x)))$$

where $next(x)$ returns the successor of x understood as $x + 1$. How such programs can be well-defined in a probabilistic context has received considerable attention [Poole, 1997], leading to definitive results such as [Riguzzi, 2015]. However, the concern here is that for queries which require infinite support and which admit infinite explanations, probabilities are computed over worlds that instantiate infinitely many terms. Moreover, logic programs were never designed for equational reasoning over complex terms like in (2), where to find a model, possibly infinitely many values need to be attempted for $advisor(advisor(mary))$.

The language BLOG [Milch et al., 2005a] also recognizes the importance of function symbols, but avoids instantiating infinitely many terms by sampling values for function symbols. When it comes to termination, however, a number of structural conditions need to hold, including one that the quantifiers over formulas can only range over finite sets. It is also unclear how the methodology should cope in the presence of negated atoms involving functions, like in (2), where infinitely many substitutions and deterministic reasoning are needed to arrive at the right answer.

In this paper, we reconsider the problem of probabilistic reasoning in a logical language with function symbols. Akin to WMC for propositional logic, our goal is to investigate a general framework that supports an expressive fragment of a logical language, while avoiding issues like instantiations of infinitely many terms. In that regard, we obtain *very encouraging results*: we show that for first-order CNF formulas – as is standard in probabilistic relational languages [Van den Broeck et al., 2014] – mentioning functions in an *arbitrary* way, logical reasoning (validity, satisfiability) is not only computable, but obtainable by means of grounding the formulas wrt a small set of constants. The essential idea is to *exploit symmetries over unmentioned constants*. In fact, we allow *existential quantifiers*, *nested function terms*, and *quantifiers over infinite sets*. With regards to probabilistic reasoning, we show that the above construction yields a simple WMC-style definition for conditional probabilities; this definition is shown to enjoy a number of attractive properties tied to logical equivalence and additivity. Throughout the paper, we demonstrate how the methodology can be *encoded* as a propositional theory, making it straightforward to use existing exact or approximate WMC technology.

2 LANGUAGE WITH RELATIONS

To prepare for the logical language with function symbols, we begin with the well-understood relational counterpart, and discuss the definition of weighted model counting.

2.1 Syntax and Semantics

We let $\mathcal{L}_D^{\text{Rel}}$ be a be a first-order language with equality, relational symbols $\{P(x), \ldots, Q(x, y), \ldots, R(x, y, z), \ldots\}$, variables $\{x, y, z, \ldots\}$, connectives $\vee, \wedge, \neg, \forall$ and a finite set of constants D, serving as the domain of discourse for quantification. Usual abbreviations hold for connectives: we write $\alpha \supset \beta$ (material implication) to mean $\neg\alpha \vee \beta$, $\alpha \equiv \beta$ (equivalence) to mean $(\alpha \supset \beta) \wedge (\beta \supset \alpha)$ and $\exists x\alpha$ (existential quantification) to mean $\neg\forall x\neg\alpha$.

The set of (ground) atoms is obtained as:[1]

$$\text{ATOMS} = \{P(a_1, \ldots, a_k) \mid P \text{ is a predicate}, a_i \in D\}.$$

A $\mathcal{L}_D^{\text{Rel}}$-model M is a $\{0, 1\}$ assignment to the elements of ATOMS. Using $\models$ to denote satisfaction, the semantics for $\phi \in \mathcal{L}_D^{\text{Rel}}$ is defined as usual inductively, but with equality as identity: $M \models (a = b)$ iff a and b are the same constants, and quantification is understood substitutionally over all elements of D: $M \models \forall x\phi(x)$ iff $M \models \phi(a)$ for

[1] Because equality is treated separately, atoms do not include equalities.

all $a \in D$. We say ϕ is *satisfiable* iff there is a $\mathcal{L}_D^{\text{Rel}}$-model M such that $M \models \phi$. We say that ϕ is *valid*, written $\models \phi$, iff for every $\mathcal{L}_D^{\text{Rel}}$-model M, $M \models \phi$.

It is not hard to see that $\mathcal{L}_D^{\text{Rel}}$ is essentially propositional. That is, let $\mathcal{L}^{\text{Prop}}$ be a propositional language over propositions $\{p, q, \ldots\}$ and connectives $\{\neg, \vee, \wedge\}$. Understanding a $\mathcal{L}^{\text{Prop}}$-model to mean $\{0, 1\}$ assignments to $\mathcal{L}^{\text{Prop}}$-formulas, we have:

Proposition 1 *Suppose $*$ is a bijection from* ATOMS *to the propositions in $\mathcal{L}^{Prop}$. For any $\alpha \in \mathcal{L}_D^{Rel}$, let α^* be α with every atom p replaced by the proposition p^*. A quantifier and equality-free formula $\phi \in \mathcal{L}_D^{Rel}$ is satisfiable iff $\phi^* \in \mathcal{L}^{Prop}$ is satisfiable.*

2.2 Weighted Model Counting

In a nutshell, WMC extends #SAT in summing the weights of the models of a propositional formula ϕ:

$$\text{WMC}(\phi, w) = \sum_{M \models \phi} w(M)$$

where M is an $\mathcal{L}_D^{\text{Rel}}$-model, and w is a *weight function.* The weight function is usually *factorized*. For example [Van den Broeck et al., 2014], suppose $v, \overline{v}$ map predicate symbols from ϕ to $\mathbb{R}^+$, $p \in M$ are those atoms that are true at M, and $p \notin M$ are those that are false. Then:[2]

$$w(M) \doteq \prod_{p \in M} v(\text{PRED}(p)) \times \prod_{p \notin M} \overline{v}(\text{PRED}(p))$$

where PRED maps an atom to its predicate.

WMC has emerged as a basic computational framework for encoding many formalisms, such as Markov logic networks [Van den Broeck et al., 2014] and ProbLog programs [Fierens et al., 2011].

Finally, given a query Q and evidence E, both of which are propositional formulas, we define the probability of Q given E wrt (ϕ, w) as:

$$\Pr_{(\phi,w)}(Q \mid E) = \text{WMC}(\phi \wedge Q \wedge E, w)/\text{WMC}(\phi \wedge E, w).$$

3 LANGUAGE WITH FUNCTIONS

We now define the logical language in a general way.

3.1 Syntax and Semantics

Let $\mathcal{L}_{\mathbb{N}}^{\text{FUN}}$ be a first-order language with function symbols $\{f(x), \ldots, g(x, y), \ldots, h(x, y, z), \ldots\}$, variables, equality, logical connectives $\{\vee, \neg, \wedge, \forall\}$ and a countably infinite set of constants (say) $\mathbb{N}$. For presentation purposes, we often use identifiers such as Latin alphabets $(a, b, \ldots)$ and proper names (*john*, *jane*, ...) to implicitly mean elements of $\mathbb{N}$. We understand $\{\supset, \equiv, \exists\}$ as usual.

The *terms* of the language form the least set containing the variables, constants, and function symbols applied to terms (e.g., $f(g(x), x, a, b)$.) A *primitive term* is a function symbol applied to constants only. A *primitive atom* is of the form $t = a$, where t is a primitive term and $a \in \mathbb{N}$. In other words, the set of primitive atoms is obtained as:

$$\text{ATOMS} = \{f(a_1, \ldots, a_k) = a_0 \mid f \text{ is a function}, a_i \in \mathbb{N}\}.$$

In general, atoms may be positive or negative, that is, they are of the form $t = t'$ or $\neg(t = t')$, and *formulas* are built from atoms and connectives as usual. We write $t \neq t'$ to mean $\neg(t = t')$. A *clause* is a disjunction of atoms. Ground formulas and terms are those not mentioning any variables.

A $\mathcal{L}_{\mathbb{N}}^{\text{FUN}}$-model M maps primitive terms to $\mathbb{N}$. We extend the notion of *co-referring* constants to arbitrary ground terms as follows. For any ground term t and M, we define $|t|_M$ as:

- if $t \in \mathbb{N}$, $|t|_M = t$;
- $|f(t_1, \ldots, t_k)|_M = M[f(a_1, \ldots, a_k)]$ where $a_i = |t_i|_M$.

The semantics for $\phi \in \mathcal{L}_{\mathbb{N}}^{\text{FUN}}$ is defined as usual inductively, but with equality as identity over co-reference: $M \models (t = t')$ iff $|t|_M$ and $|t'|_M$ are the same constants, and quantification is understood substitutionally over all constants: $M \models \forall x \phi(x)$ iff $M \models \phi(a)$ for all $a \in \mathbb{N}$. Satisfaction and validity is defined as usual.

3.2 A Propositional Fragment

Our intuitions from $\mathcal{L}_D^{\text{Rel}}$ and $\mathcal{L}^{\text{Prop}}$ do not carry over to $\mathcal{L}_{\mathbb{N}}^{\text{FUN}}$ in an obvious way as it is defined over infinitely many terms and an infinite domain. In fact, even disallowing nesting of function symbols does not make it any easier. To see this, call a formula *flat* iff the argument to a function is either a constant or a variable. A ground flat clause is then of the form $f(a_1, \ldots, a_k) = a_0 \vee \ldots \vee g(b_1, \ldots, b_k) \neq b_0$. So, consider the formula $f(1) \neq 1$. Clearly, this has infinitely many models: $\{f(1) = c \mid c \in \mathbb{N} - \{1\}\}$, and moreover, WMC would also be ill-defined in general.

However, if we define the fragment $\mathcal{L}_D^{\text{FUN}} \subset \mathcal{L}_{\mathbb{N}}^{\text{FUN}}$ as the set of ground flat formulas over a finite set of constants $D \subsetneq \mathbb{N}$, then we note that ATOMS is finite, and every formula of $\mathcal{L}_D^{\text{FUN}}$ has finitely many models.

[2] Alternatively, a weight function could map atoms themselves to positive reals [Chavira and Darwiche, 2008]. The nature of the weight function is irrelevant for most of the technical development in this paper.

We can also intuit a definition for WMC. Given a ground flat formula ϕ, a domain D, and a weight functions $v, \overline{v}$ that map function symbols to positive reals, we have:

$$\text{WMC}(\phi, D, w) = \sum_{M \models \phi} \prod_{p \in M} v(\text{FUN}(p)) \times \prod_{p \notin M} \overline{v}(\text{FUN}(p))$$

where FUN maps a primitive atom to its function symbol, and as usual $p \in M$ are those primitive atoms that are true at M, and $p \notin M$ are those that are false.[3]

Example 2 Consider $f(a) \neq a$ over domain $\{a, b\}$. We have primitive atoms $\{f(a) = b, f(b) = b, f(a) = a, f(b) = a\}$, and models $M_1 = \{f(a) = b, f(b) = a\}$ and $M_2 = \{f(a) = b, f(b) = b\}$. Then the atoms in $\{f(a) = b, f(b) = a, f(a) \neq a, f(b) \neq b\}$ are satisfied at M_1 and those in $\{f(a) = b, f(b) = b, f(a) \neq a, f(b) \neq a\}$ at M_2. Suppose v maps f to 2 and $\overline{v}$ maps it to 1. Then, $w(M_1) = w(M_2) = 2 \times 2 \times 1 \times 1 = 4$. So WMC is 8.

3.3 More Expressiveness

We are caught between two extremes. The language $\mathcal{L}^{\text{FUN}}_{\mathbb{N}}$ is undecidable, but on the other hand, $\mathcal{L}^{\text{FUN}}_{D}$ is clearly too limited as we have gained nothing over $\mathcal{L}^{\text{Prop}}$. We will now attempt to motivate a fragment with first-order features in between these extremes. Consider that the fragment will determine the expressiveness of our knowledge bases.

Perhaps the simplest way to go beyond $\mathcal{L}^{\text{FUN}}_{D}$ is to limit our attention to ground flat formulas from $\mathcal{L}^{\text{FUN}}_{\mathbb{N}}$. As mentioned earlier, this fragment has infinitely many atoms, and formulas such as $f(1) \neq 1$ has infinitely many models. Thus, it already represents a non-trivial and challenging extension to $\mathcal{L}^{\text{Prop}}$.

What the fragment does not yet allow is existential quantifiers. One of our primary motivations was to allow for identity uncertainty in an unrestricted manner: that is, admitting sentences such as $\exists x f(x) \neq 1$ in our KBs. Of course, we can use Skolemization: introduce a new nullary function g and concern ourselves instead with the sentence $f(g) \neq 1$. Moreover, this formula can be brought to a flat form: $\forall z(g = z \supset f(z) \neq 1)$. The very same recipe can also be applied to nesting of function symbols: consider $\forall x(next(x) \neq x \supset next(next(x)) \neq x)$, which says that if the successor of x is not x, then neither is the successor of the successor of x. This is equivalent to $\forall x, z(next(x) \neq x \supset (next(x) = z \supset next(z) \neq x))$.

These reformulations notwithstanding, it should be clear that since quantifiers range over $\mathbb{N}$, a formula such as $\forall z(g = z \supset f(z) \neq 1)$ is logically equivalent to infinitely many clauses: $(g = 1 \supset f(1) \neq 1), (g = 2 \supset f(2) \neq 1)$, and so on. We will identify a way to effectively reason over such knowledge bases. Concretely, let a *basic* clause be a disjunction of atoms of the form $f(x_1, \ldots, x_k) \circ x_0$, where $\circ \in \{=, \neq\}$ and x_i are either constants or variables. (So, a ground basic clause is a disjunction of positive or negative primitive atoms.) Then:

An *acceptable knowledge base* is a satisfiable, *compact* and finite set of *universally quantified basic clauses*.

This is a very expressive fragment that handles existential quantifiers, nested functions, and infinite domains. (We need to assume that knowledge bases are *compact,* a property we formally define in a short while.) To handle acceptable KBs, however, we will first need to solve the problem of reasoning about ground flat formulas from $\mathcal{L}^{\text{FUN}}_{\mathbb{N}}$.

4 SATISFIABILITY AND VALIDITY

Before we discuss model counting, we will need to consider the problem of logical reasoning for which we first turn to ground flat formulas from $\mathcal{L}^{\text{FUN}}_{\mathbb{N}}$ before considering quantified formulas.

4.1 Ground Flat Formulas

Since formulas such as $f(1) \neq 1$ has infinitely many models over infinitely many literals, we cannot readily apply existing propositional techniques. Our key result here is to show that logical reasoning is possible by considering finitely many values for terms.

To achieve the result, a crucial observation is this: when considering assignments to primitive terms in a ground flat clause, constants that are outside the KB behave identically, at least as far as satisfiability is concerned. For example, consider $f \neq 1$. Then, $(f = 2) \wedge (f \neq 1)$ is satisfiable, but so is $(f = 3) \wedge (f \neq 1)$.

More precisely, given a ground flat formula ϕ in CNF (that is, conjunctions of clauses), define:

- $\text{TERMS}(\phi)$ as the set of primitive terms in ϕ;
- $\text{CONS}(\phi)$ is the set of constants in ϕ;
- $\text{XOR}(t, C) = \bigvee_{a \in C}(t = a)$, for any $C \subset \mathbb{N}$;
- $\text{XOR}(\phi) = \phi \wedge \bigwedge_{t \in \text{TERMS}(\phi)} \text{XOR}(t, D)$, where $D =$

[3] Our definition of WMC is reasonable when considering the simulation of a predicate $P(x_1, \ldots, x_k)$ by a function $f(x_1, \ldots, x_k)$ as follows: $P(x_1, \ldots, x_k) \doteq f(x_1, \ldots, x_k) = 1$ with the additional constraint: $f(x_1, \ldots, x_k) = 1 \vee f(x_1, \ldots, x_k) = 0$. It is then not hard to see that our definition of WMC for functions coincides with the relational version. While variant formulations are conceivable, much of the technical contributions in this paper do not hinge on the definition.

$\text{CONS}(\phi) \cup \{b\}$ and b is any constant not mentioned in $\text{CONS}(\phi)$.

So, $\text{XOR}(t, D)$ is a disjunction over possible assignments to t, which includes every constant in ϕ plus an arbitrary new one, and $\text{XOR}(\phi)$ conjoins this constraint for all terms to ϕ.

Example 3 Let $\phi = (f \neq 1 \vee g(1) = 2) \wedge g(1) \neq 1$. Then for $t \in \{f, g(1)\}$, $\text{XOR}(t, D) = (t = 1 \vee t = 2 \vee t = 3)$, where $D = \{1, 2, 3\}$ and 3 is some arbitrary new constant. So $\text{XOR}(\phi) = \phi \wedge \text{XOR}(f, D) \wedge \text{XOR}(g(1), D)$.

It is worth remarking that repeated applications of XOR do not change the meaning of the formula:

Proposition 4 $\text{XOR}(\phi) \equiv \text{XOR}(\text{XOR}(\phi))$.

Proof: Suppose $\text{CONS}(\phi) = \{a_1, \ldots, a_k\}$, and let $D = \text{CONS}(\phi) \cup \{b\}$. Then, for every $t \in \text{TERMS}(\phi)$, $\text{XOR}(\phi)$ includes $\text{XOR}(t, D)$. Observe that $\text{TERMS}(\text{XOR}(\phi)) = \text{TERMS}(\phi)$. So, for for every $t \in \text{TERMS}(\phi)$, $\text{XOR}(\text{XOR}(\phi))$ includes $\text{XOR}(t, D)$ (from $\text{XOR}(\phi)$) and $\text{XOR}(t, D')$, where $D' = D \cup \{c\}$ and c is some new constant. But then $\text{XOR}(t, D) \equiv \text{XOR}(t, D) \wedge \text{XOR}(t, D')$, since $\text{XOR}(t, D')$ is a weaker constraint. ∎

Putting it together, here is the main theorem:

Theorem 5 *Suppose ϕ, α are ground flat formulas. Then $\phi \models \alpha$ iff $\text{XOR}(\phi \wedge \neg\alpha)$ is unsatisfiable.*

Essentially, the forcing constraints for primitive terms added to a formula limit the number of possible assignments, making it essentially equivalent to a propositional theory, a point we return to later. But first, to establish the theorem, we proceed as follows:

Lemma 6 *Suppose ϕ is a ground flat formula, and $t \in \text{TERMS}(\phi)$. For any $a, b \notin \text{CONS}(\phi)$, $\phi \wedge (t = a)$ is satisfiable iff $\phi \wedge (t = b)$ is satisfiable.*

Proof: Suppose $M \models \phi \wedge (t = a)$ iff by definition, $M \models \phi$ and $M[t] = a$ iff $M \models \phi^t_a$, where ϕ^t_a is ϕ with every occurrence of t replaced by a. Now, ϕ^t_a does not mention t, and moreover, every occurrence of $t = c$ for some $c \in \text{CONS}(\phi)$ is replaced by $a = c$, and every occurrence of $t \neq c$ is replaced by $a \neq c$. But because $a \notin \text{CONS}(\phi)$ and equality is understood as identity, $a = c$ is false everywhere, and $a \neq c$ is true everywhere. By extension, because $b \notin \text{CONS}(\phi)$, ϕ^t_a is logically equivalent to ϕ^t_b. Since ϕ^t_b does not actually mention t, construct a M' such that $M'[t] = b$ and for all terms $t' \neq t$, $M'[t'] = M[t']$. Clearly, $M' \models \phi \wedge (t = b)$. ∎

Lemma 7 *Suppose ϕ, D and t are as above. Then ϕ is satisfiable iff $\phi \wedge \text{XOR}(t, D)$ is satisfiable.*

Proof: The if direction is obvious, since the satisfiability of $\phi \wedge \text{XOR}(t, D)$ immediately implies that of ϕ.

So suppose $M \models \phi$. By construction, there is some $a \in \mathbb{N}$ such that $M[t] = a$. So $\phi \wedge (t = a)$ is satisfiable.

Suppose $(t = a)$ is mentioned in $\text{XOR}(t, D)$, then $M \models \text{XOR}(t, D)$ and so we are done. Suppose not, that is, $a \notin D$. Then, $a \notin \text{CONS}(\phi)$ as well. Suppose $b \in (D - \text{CONS}(\phi))$. So we have that $\phi \wedge (t = a)$ is satisfiable iff by Lemma 6, $\phi \wedge (t = b)$ is satisfiable, which means that $\phi \wedge (t = a_1 \vee \ldots \vee t = a_k \vee t = b)$ must be as well. ∎

Corollary 8 *Suppose ϕ is above. Then ϕ is satisfiable iff $\text{XOR}(\phi)$ is satisfiable.*

We are now prepared for the proof of Theorem 5.

Proof: Here, $\phi \models \alpha$ iff $\phi \wedge \neg\alpha$ is unsatisfiable iff by Corollary 8, $\text{XOR}(\phi \wedge \neg\alpha)$ is unsatisfiable. ∎

The added constraints essentially limit the number of assignments to primitive terms, and in fact, we can think of each atom in $\text{XOR}(t, D)$ as a proposition, and thereby return to the familiar setting of $\mathcal{L}^{\text{Prop}}$. First, let us introduce the *logical exclusive-or* symbol $\oplus$ with the understanding that for any model M, $M \models p_1 \oplus \ldots \oplus p_k$ iff $M \models p_i$ for some i and $M \not\models p_j$ for all $j \neq i$. Observe that, in $\mathcal{L}^{\text{FUN}}_{\mathbb{N}}$, replacing the disjunctions in $\text{XOR}(t, D)$ with $\oplus$ does not affect logical equivalence, since the semantics enforces it anyway.

Proposition 9 *Suppose $\text{XOR}(\phi)$ is as above, and $\text{XOR}'(\phi)$ denote the same formula but with every $\text{XOR}(t, D) = (t = a_1 \vee \ldots \vee t = a_k)$ replaced by $(t = a_1 \oplus \ldots \oplus t = a_k)$. Then $\text{XOR}(\phi) \equiv \text{XOR}'(\phi)$.*

However, the point is that by making the exclusive-or explicit preserves satisfaction in the simpler $\mathcal{L}^{\text{Prop}}$.

Proposition 10 *Suppose $*$ is a bijection from the atoms in $\text{XOR}(\phi)$ to the propositions in $\mathcal{L}^{Prop}$. Then ϕ is satisfiable iff $(\text{XOR}'(\phi))^* \in \mathcal{L}^{Prop}$ is satisfiable.*

Example 11 Suppose $\phi = (f \neq 1)$, then $\text{XOR}(\phi) = \phi \wedge (f = 1 \vee f = 2)$. Then the $\mathcal{L}^{\text{FUN}}_{\mathbb{N}}$-model $M \models (f = 2) \wedge (f \neq 1)$ is the only one satisfying $\text{XOR}(\phi)$. Alternatively, suppose p and q are propositions from $\mathcal{L}^{\text{Prop}}$, and suppose $*$: $[f = 1, f = 2] \rightarrow [p, q]$ is a bijection, then $(\text{XOR}'(\phi))^* = \neg p \wedge (p \oplus q)$, and the $\mathcal{L}^{\text{Prop}}$-model $M = \{q, \neg p\}$ is the only one satisfying $(\text{XOR}'(\phi))^*$.

Corollary 12 *Suppose ϕ, α are ground flat formulas.*

Then there is a formula $\beta = (\text{XOR}'(\phi \wedge \neg\alpha))^* \in \mathcal{L}^{Prop}$ *such that* $\phi \models \alpha$ *iff* β *is unsatisfiable.*

4.2 Acceptable Knowledge Bases

Here, we will be interested in a version of Corollary 12 but under the generalization that ϕ is an acceptable KB. Unfortunately, in general, acceptable KBs are logically equivalent to a possibly infinite set of ground clauses, so we cannot simply apply Corollary 12. Our key result will be to show that grounding the KB wrt a finite set of constants is sufficient for checking satisfiability.

First, some notation. Given a universally quantified basic clause such as $\forall x, y(f(x) = y \vee g(y) = x)$, let its *rank* be the maximum number of variables mentioned (here 2). A ground clause, then, has 0 rank. The *rank* of an acceptable KB is the maximum of the ranks of its clauses.

Given an acceptable KB ϕ, a ground theory is obtained by substituting variables with constants. Suppose θ denotes a substitution. For any set of names $D \subseteq \mathbb{N}$, we write $\theta \in D$ to mean substitutions are only allowed wrt the constants in D. Then, define:

- $\text{GND}(\phi) = \{c\theta \mid (\forall x, \ldots, y\ c) \in \phi, \theta \in \mathbb{N}\}$;
- $\text{GND}(\phi, k) = \{c\theta \mid (\forall x, \ldots, y\ c) \in \phi, \theta \in D\}$, where $k \geq 0$ and D is the set of constants mentioned in ϕ plus k (arbitrary) new ones;
- $\text{RGND}(\phi) = \text{GND}(\phi, k)$ where k is the rank of ϕ;
- $\text{XOR}(\phi) = \psi \wedge \bigwedge_{t \in \text{TERMS}(\psi)} \text{XOR}(t, D)$ where $\psi = \text{RGND}(\phi)$, and $D = \text{CONS}(\psi) \cup \{b\}$ and b is any constant not mentioned in $\text{CONS}(\psi)$.

Note that $\text{XOR}(\phi)$ for acceptable KBs and ground flat formulas is compatible: if ϕ is a ground flat formula, then its rank is 0, and so, in fact, $\text{RGND}(\phi) = \phi$.

Example 13 Let ϕ denote $\{\forall x(\mathit{father}(x) \neq \mathit{adam} \supset \mathit{mother}(x) \neq \mathit{beth})\}$. Abbreviating symbols to their first letters, we have:

- $\text{GND}(\phi, 0) = \{f(a) \neq a \supset m(a) \neq b, f(b) \neq a \supset m(b) \neq b\}$;
- $\text{RGND}(\phi) = \text{GND}(\phi, 0) \cup \{f(c) \neq a \supset m(c) \neq b\}$, because ϕ's rank is 1 and where c is arbitrarily chosen from $\mathbb{N} - \{a, b\}$;
- $\text{XOR}(\phi) = \text{RGND}(\phi) \cup \{\text{XOR}(t, D) : t \in T\}$, where $T = \{f(a), m(a), f(b), m(b), f(c), m(c)\}$ and $D = \{a, b, c, d\}$.
- $\text{GND}(\phi) = \text{RGND}(\phi) \cup \{f(d) \neq a \supset m(d) \neq b, f(e) \neq a \supset m(e) \neq b, \ldots\}$.

We are now in a position to formally discuss the *compact* property we expect of our knowledge bases.

Definition 14 Suppose ϕ is a conjunction of universally quantified clauses, of rank k. Suppose $\text{GND}(\phi)$ is satisfiable iff $\text{GND}(\phi, j)$ is satisfiable for all $j \geq k$. Then we say ϕ is *compact.*

Recall that the compactness theorem [Enderton, 1972] says that a possibly infinite set of sentences S is satisfiable iff every finite subset of S is satisfiable. This property does not hold $\mathcal{L}^{\text{FUN}}_{\mathbb{N}}$, not even when restricted to universally quantified basic clauses. Therefore, we explicitly require that our knowledge bases are compact.[4]

The main theorem for the section is:

Theorem 15 *Suppose* ϕ *is an acceptable KB, and* α *is any ground flat formula. Then* $\phi \models \alpha$ *iff* $\text{XOR}(\phi \wedge \neg\alpha)$ *is unsatisfiable.*

Proof: Let us suppose ϕ's rank is k. We first begin with some observations. Note that $\phi \models \alpha$ iff $\phi \wedge \neg\alpha$ is unsatisfiable iff $\text{GND}(\phi) \wedge \neg\alpha$ is unsatisfiable. Also note that $\psi = \phi \wedge \neg\alpha$ is also an acceptable KB, since $\neg\alpha$ can be converted to a ground flat clause. Finally, by Corollary 8, $\text{RGND}(\psi)$ is satisfiable iff $\text{XOR}(\psi)$ is, and so in the arguments below, we only make use of $\text{RGND}(\psi)$.

The if direction is immediate, because if $\text{RGND}(\phi \wedge \neg\alpha)$ is unsatisfiable, then so is $\text{GND}(\phi \wedge \neg\alpha) = \text{GND}(\phi) \wedge \neg\alpha$ since $\text{RGND}(\phi \wedge \neg\alpha) \subseteq \text{GND}(\phi \wedge \neg\alpha)$.

For the only-if direction, suppose $\phi \models \alpha$ but $\text{RGND}(\phi \wedge \neg\alpha)$ is satisfiable. It suffices to show that for the acceptable KB $\psi = \phi \wedge \neg\alpha$ of rank k if $\Delta = \text{RGND}(\psi) = \text{GND}(\psi, k)$, is satisfiable then so is $\Delta' = \text{GND}(\psi, j)$ for any $j \geq k$. Then, by the compact property assumption, this means $\text{GND}(\psi) = \text{GND}(\phi) \wedge \neg\alpha$ is also satisfiable, which contradicts $\phi \models \alpha$.

Without loss of generality, assume $\text{CONS}(\Delta') = \text{CONS}(\Delta)$ plus n new ones, where $n = j - k$. By assumption, there is a M such that $M \models \Delta$. Construct a M':

1. for all $t \in \text{TERMS}(\Delta)$, $M'[t] = M[t]$;

[4] Our view is that KBs not enjoying the compact property do not come up often in practice. To see an example of such a KB, imagine a language with 2 function symbols that simulate relations $R(x, y)$ and $P(x)$ respectively. Then, let ϕ be $\{\forall x, y, z\ (R(x, y) \wedge R(y, z) \supset R(x, z)), \forall x\ (\neg R(x, x)), \forall x\ (P(x) \supset \exists y\ (R(x, y) \wedge P(y))), \forall x\ (\neg P(x) \supset \exists y\ (R(x, y) \wedge \neg P(y))), P(1), \neg P(2)\}$. That is, R is irreflexive but transitive. After Skolemization and flattening, it can be seen that $\text{XOR}(\phi)$ (and hence, $\text{RGND}(\phi)$) is unsatisfiable, but strangely, $\text{GND}(\phi)$ is indeed satisfiable in a $\mathcal{L}^{\text{FUN}}_{\mathbb{N}}$-model where infinitely many instances of P are true and infinitely many instances of P are false.

2. for all $t \notin \text{TERMS}(\Delta')$, $M'[t] = M[t]$;

3. for all $t \in \text{TERMS}(\Delta' - \Delta)$, do as follows. Consider that t is mentioned in some $c\theta \in (\Delta' - \Delta)$ such that $\forall x, \ldots, y\ c \in \psi$, and θ is a substitution possibly using constants from ψ and Δ, but at least some, say $a_1, \ldots, a_l$, $l \leq k$, from $(\Delta' - \Delta)$. (Otherwise, $c\theta \notin (\Delta' - \Delta)$.) But by that account, θ does not mention l constants from $\text{CONS}(\Delta) - \text{CONS}(\psi)$, say $b_1, \ldots, b_l$. Let $*$ be a bijection from $\mathbb{N}$ to $\mathbb{N}$ such that maps a_i with b_i but otherwise maps constants to themselves.

   By construction, $(c\theta)^*$, that is, $c\theta^* \in \Delta$. In other words, for every $f(e_1, \ldots, e_z) = e_0$ appearing in $c\theta$, $(f(e_1, \ldots, e_z) = e_0)^*$, that is, $(f(e_1^*, \ldots, e_z^*) = e_0^*) \in c\theta^*$. Let $M'[f(e_1, \ldots, e_z)] = M[f(e_1^*, \ldots, e_z^*)]$.

We have completed the construction of M'. It is easy to show (by induction) that M' satisfies Δ because of construction step (1). Similarly, it is easy to show that M' satisfies $\Delta' - \Delta$ because of construction step (3). Thus, M' satisfies Δ' and we are done. ∎

Example 16 Suppose ϕ is the conjunction of $female(beth) = 1, \forall x(female(x) = 1 \vee female(x) = 0)$, and $\forall x(female(x) = 1 \supset \forall y(father(y) \neq x))$, that is, if x is a female than she is not a father to any y.

Suppose α is $father(beth) \neq beth$. Suppose $\text{RGND}(\phi \wedge \neg\alpha)$ is wrt constants $\{beth, adam, dave\}$, since $(\phi \wedge \neg\alpha)$'s rank is 2. Observe that $\text{XOR}(\phi \wedge \neg\alpha)$ includes $father(beth) \neq beth$ from ϕ and $father(beth) = beth$ from $\neg\alpha$, which is inconsistent. So $\phi \models \alpha$.

Suppose $\beta = (father(adam) = beth)$. It is not hard to see that $\text{XOR}(\phi \wedge \neg\beta)$ is satisfiable, and so $\phi \not\models \beta$.

At this point, it should be clear that the familiar setting of $\mathcal{L}^{\text{Prop}}$ is sufficient here as well:

Proposition 17 *Suppose $*$ is a bijection from the atoms in $\text{XOR}(\phi \wedge \neg\alpha)$ to the propositions in $\mathcal{L}^{Prop}$, and $\text{XOR}'(\phi)$ is as before. Then $\phi \models \alpha$ iff $(\text{XOR}'(\phi \wedge \neg\alpha))^* \in \mathcal{L}^{Prop}$ is unsatisfiable.*

5 WEIGHTED MODEL COUNTING

We now propose an account of WMC for acceptable KBs, which will then lead to a notion of conditional probabilities. The proposal is motivated from a logical point of view, that is, the definition of probabilities obeys a number of non-trivial properties that respect logical entailment over possibly infinitely many atoms.[5] In essence, the definition satisfies the following properties in a *first-order setting*:

1. *If the query is believed*, it has probability 1.
2. *If the query contradicts what is believed*, it has probability 0.
3. *If the query mentions unknown individuals*, these individuals are interchangeable with other unknowns.
4. *If two queries are logically equivalent*, then they have the same probability.
5. Probabilities satisfy the *addition laws* over Boolean connectives [Fagin and Halpern, 1994].

The proposal, as one would surmise, leverages Theorem 15 as this has the advantage of grounding the KB wrt finitely many constants. Moreover, from a representational viewpoint, no structural assumptions are needed about the encoding. Recall that we provided a definition for $\text{WMC}(\phi, D, w)$ earlier, that is, where the domain was fixed to a finite set of constants D. If the D is implicit, as enabled by a set of possible assignments like $\text{XOR}(t, D)$, we simply write $\text{WMC}(\phi, w)$. Turning now to a countably infinite domain, such as $\mathbb{N}$, we define:

$$\text{WMC}(\phi, \mathbb{N}, w) = \text{WMC}(\text{XOR}(\phi), w)$$

That is, we compute the WMC of $\text{XOR}(\phi)$ (where the domain is implicit).

Suppose Q, E are ground flat formulas.[6] Then, the probability of Q given E wrt (ϕ, w) is defined as:

$$\Pr_{(\phi,\mathbb{N},w)}(Q \mid E) = \text{WMC}(\phi \wedge Q \wedge E, \mathbb{N}, w)/\text{WMC}(\phi \wedge E, \mathbb{N}, w).$$

We drop the subscript $(\phi, \mathbb{N}, w)$ when the context is clear, and often write $\Pr(Q)$ to mean $E = true$. Of course, by means of Proposition 17, probabilities can be computed using propositional WMC. For the sake of completeness, Algorithm 1 provides the pseudocode.

Overall, the definition is reasonable in terms on the following properties:

Theorem 18 *Suppose ϕ is any acceptable KB, α, β ground flat formulas, and w any weight function mapping function symbols to positive reals. Then*

[5] Other accounts of probabilities in the presence of infinitely many atoms are also conceivable [Singla and Domingos, 2007, Riguzzi, 2015].

[6] Observe that the WMC formulation is sensitive to the set of ground atoms over which the set of interpretations are considered. So, for purely technical reasons, we need to assume that all the primitive atoms in $\{Q, E\}$ are also mentioned in ϕ. This is without loss of generality: let $\gamma = \{(p \vee \neg p) \mid \text{atom } p \text{ is mentioned in } Q \wedge E\}$. Since $\phi \wedge \gamma \equiv \phi$, replace ϕ with $\phi \wedge \gamma$ when computing the model count.

Algorithm 1 $\Pr_{(\phi,\mathbb{N},w)}(Q \mid E)$ using propositional WMC

1: $\alpha_1 = \phi \wedge Q \wedge E, \alpha_2 = \phi \wedge E$
2: $\beta_i = \text{XOR}'(\alpha_i)$
3: ▷ Construct bijection $*$ to map atoms in $\text{XOR}(\alpha_i)$ to propositions $p_1, \ldots, p_n$
4: $\gamma_i = (\beta_i)^*$
5: ▷ Construct u to map a $\mathcal{L}^{\text{Prop}}$ model to the weight of the corresponding $\mathcal{L}^{\text{FUN}}_{\mathbb{N}}$ model, which is obtained by mapping propositions back to $\mathcal{L}^{\text{FUN}}_{\mathbb{N}}$-atoms using $*$
6: return $\text{WMC}(\gamma_1, u)/\text{WMC}(\gamma_2, u)$

1. $\Pr(\alpha) = 1$ *if* $\models \phi \supset \alpha$;
2. $\Pr(\alpha) = 0$ *if* $\models \phi \supset \neg\alpha$;
3. $\Pr(\alpha) = \Pr(\alpha^*)$ *for any bijection* $*$ *from* $\mathbb{N}$ *to* $\mathbb{N}$ *such that it maps names from* ϕ *to themselves and otherwise arbitrary;*
4. $\Pr(\alpha) = \Pr(\beta)$ *if* $\alpha \equiv \beta$;
5. $\Pr(\alpha \vee \beta) = \Pr(\alpha) + \Pr(\beta) - \Pr(\alpha \wedge \beta)$;
6. $\Pr(\alpha) = \Pr(\alpha \wedge \beta) + \Pr(\alpha \wedge \neg\beta)$.

Proof: For (1), since $\models \phi \supset \alpha$, $\phi \equiv \phi \wedge \alpha$. For (2), if $\phi \models \neg\alpha$, $\phi \wedge \alpha$ has no model. For (3), by construction, $(\phi \wedge \alpha)^* = \phi \wedge \alpha^*$. By induction on α, we can show that for any model M of $\phi \wedge \alpha$, we can construct a model M' of $\phi \wedge \alpha^*$, and vice versa. For (4), assume α and β mention the same set of atoms. (If not, augment the formulas appropriately in a equivalence-preserving manner.) Then, $M \models \phi \wedge \alpha$ iff by definition, $M \models \phi$ and $M \models \alpha$ iff by assumption, $M \models \phi$ and $M \models \beta$ iff $M \models \phi \wedge \beta$. So every model of $\text{XOR}(\phi \wedge \alpha)$ is a model of $\text{XOR}(\phi \wedge \beta)$ and vice versa. For (5), $M \models \phi \wedge (\alpha \vee \beta)$ iff $M \models \phi \wedge \alpha$ or $M \models \phi \wedge \beta$. It is not hard to see that the models of $\text{XOR}(\phi \wedge (\alpha \vee \beta))$ is the union of $\text{XOR}(\phi \wedge \alpha)$ and $\text{XOR}(\phi \wedge \beta)$. By applying the cardinality property for set union, we avoid double counting by subtracting the intersection, which is precisely the set of models for $\text{XOR}(\phi \wedge (\alpha \wedge \beta))$. The proof for (6) is analogous, but we recognize that the models for $\phi \wedge \alpha \wedge \beta$ and those for $\phi \wedge \alpha \wedge \neg\beta$ are disjoint. ∎

Example 19 Suppose ϕ is the conjunction of $\mathit{father}(\mathit{adam}) = \mathit{dave}$, and $\forall x(\mathit{father}(x) \neq x)$. Suppose the weight function is factorized in terms of $v, \overline{v}$ (that is, positive and negative instances) which map *father* to 1 and 1 respectively, for simplicity. Here are a few examples corresponding to items 1, 3 and 5 from Theorem 18:

- Suppose α is $\mathit{father}(\mathit{dave}) \neq \mathit{dave}$. Since ϕ's rank is 1, and letting *beth* be the new constant, it is easy to see that $\text{XOR}(\phi)$ includes formulas such as $\mathit{father}(\mathit{adam}) \neq \mathit{adam}$ and $\mathit{father}(\mathit{dave}) \neq \mathit{dave}$, and so $\text{XOR}(\phi)$ is identical to $\text{XOR}(\phi \wedge \alpha)$, and thus, $\Pr(\alpha) = 1$.

- Suppose α is $\mathit{father}(\mathit{john}) = \mathit{dave}$. Since *john* is not mentioned in ϕ, let $*$ be a bijection that maps *john* to *jane* but otherwise maps constants to themselves. So $\alpha^* = (\mathit{father}(\mathit{jane}) = \mathit{dave})$.

  Recall that, prior to computing the model counts, we need to ensure that the atoms in the query are also mentioned in ϕ. So, let $\phi' = \phi \wedge (\alpha \vee \neg\alpha)$. Then $\text{RGND}(\phi')$ involves grounding ϕ' wrt the constants in ϕ' plus one new one, say, *mary*. Next, $\text{XOR}(\phi')$ over a new constant *beth* essentially boils down to the ground terms $\mathit{father}(x), x \in \{\mathit{dave}, \mathit{john}, \mathit{mary}\}$ taking on possible values $y \in \{\mathit{dave}, \mathit{adam}, \mathit{john}, \mathit{beth}, \mathit{mary}\}$ under the constraint $x \neq y$. Note that ϕ' already assigns *dave* to $\mathit{father}(\mathit{adam})$, and so this ground term is not significant. In sum, each of the 3 ground terms can take on 4 possible values, and so the model count of $\text{XOR}(\phi) = 4^3$. In contrast, $\text{XOR}(\phi' \wedge \alpha)$ additionally assigns a value to the term $\mathit{father}(\mathit{john})$ and therefore its model count is 4^2, yielding $\Pr(\alpha) = 1/4$.

  Now, consider α^* and to account for the unmentioned atoms, let $\phi'' = \phi \wedge (\alpha^* \vee \neg\alpha^*)$. It is not hard to see that the model count of $\text{XOR}(\phi'') = 4^3$ and that of $\text{XOR}(\phi'' \wedge \alpha^*) = 4^2$ and so $\Pr(\alpha^*) = 1/4$.

- Suppose α is $\mathit{father}(\mathit{john}) = \mathit{dave}$ and β is $\mathit{father}(\mathit{john}) = \mathit{adam}$. To account for the unmentioned atoms, let $\phi' = \phi \wedge (\alpha \vee \neg\alpha) \wedge (\beta \vee \neg\beta)$. Analogous to how we proceeded before, $\text{XOR}(\phi')$ has a model count of 4^3, and $\text{XOR}(\phi' \wedge \alpha), \text{XOR}(\phi' \wedge \beta)$ a model count of 4^2. Since $\alpha \wedge \beta$ is unsatisfiable, $\text{XOR}(\phi' \wedge (\alpha \wedge \beta))$ has a model count of 0. So, $\Pr(\alpha) = 1/4$, $\Pr(\beta) = 1/4$, $\Pr(\alpha \wedge \beta) = 0$ and therefore, $\Pr(\alpha \vee \beta) = 1/2$.

To reiterate the intuition, these probabilities account for the unknowns: $\text{RGND}(\phi)$ accounts for unknown terms, $\text{XOR}(\phi)$ accounts for unknown values to these terms, which is then used to define Pr.

6 LINEAGE AND DISCUSSION

In the literature on extending graphical models to handle first-order features, several influential papers have appeared. We sketch the main directions that are most closely related to our own work, and refer interested readers to [Milch et al., 2005a, Russell, 2015] for a more comprehensive list.

While existential quantifiers are already discussed in finite-domain Markov logic networks [Richardson and Domingos, 2006], these are, in fact, interpreted as a disjunction over ground instances, which is possible because the domain is finite. To avoid such expansions, [Van den Broeck et al., 2014] suggest an elegant alternative. There is also extensive work in database theory on labeled unknowns [Farré et al., 2007], and in the verification community on functions in ground formulas against background axioms (e.g., theory of arithmetic) [Barrett et al., 2009]. In that vein, logical languages with arithmetic constraints have been shown to be useful for probabilistic reasoning in mixed discrete-continuous spaces [Sanner and Abbasnejad, 2012, Belle et al., 2015, Alberti et al., 2016, de Salvo Braz et al., 2016]. Using the relational structure to speed-up inference has also received attention [Gogate and Domingos, 2011, de Salvo Braz et al., 2005, Van den Broeck et al., 2014]. A synthesis of both threads is further investigated in [de Salvo Braz and O'Reilly, 2017].

The starting point for our work was a recent attempt to lift the finite domain assumption in probabilistic relational languages [Belle, 2017]. The thrust of our results follows the style of that work; however, function symbols necessitated a completely new set of technical devices. Recall that a key construction throughout this paper was a way to reason about possible assignments to ground terms in a finite manner; this is based on a reformulation and generalization of the development in [Belle and Lakemeyer, 2011]. Moreover, when limited to universally quantified clauses over a relational vocabulary, the compactness theorem from propositional logic is applicable [Belle, 2017, Theorem 8]. As we saw, compactness does not hold for us. Finally, from an expressiveness point of view, functions are more powerful. Recall that functions can simulate predicates easily. For a function $f(x)$ to be simulated by a predicate, we need an extra argument for the value: $f(x) = z$ can be represented by $P(x, z)$. The trouble is that functions are implicitly assumed to be assigned a value, but then we would need a constraint $\exists z\ (P(x, z))$ in the knowledge base, but this cannot be represented in [Belle, 2017], because no existential quantifiers are allowed. (It cannot be replaced by a disjunction because the range of z is $\mathbb{N}$.) Of course, with functions, we are able to also handle nesting of terms.

In general, the crux of our work is in handling infinite domains. In that regard, [Singla and Domingos, 2007] handle infinite domains in Markov logic networks using locality constraints over Gibbs measures, [Jaeger, 1998] proves decidability in infinite-domain relational BNs given independence constraints in query atoms, [Laskey and da Costa, 2005] provides a semantics for infinite-state BNs assuming stratification conditions, and [Milch et al., 2005b] provide a semantics for infinite-state BNs while assuming that only a finite number of ancestors affect a variable, which drives the formal foundations of BLOG. Readers may want to consult [Milch et al., 2005a, Singla and Domingos, 2007] for a better understanding on how other known methods in the literature, e.g. [Gilks et al., 1994], compare to those listed here. Finally, as mentioned earlier, most languages based on distribution semantics admit infinite instantiations [Sato, 1995].

In our view, all of these approaches are interesting, and it is difficult, from a formal angle, to suggest that one approach subsumes another. What we have attempted in this paper is a new algorithmic contribution that: (a) makes *no assumptions about the syntax* (outside of assuming it is in CNF, which is usual), (b) can embody full logical reasoning over *complex, nested terms* and *unbounded quantification*, and (c) can simply use any standard exact or approximate model counter.

While we investigated general results, it is conceivable that more restricted fragments are sufficient for the application at hand. Perhaps ground flat formulas in $\mathcal{L}_{\mathbb{N}}^{\text{FUN}}$ (discussed in Section 4.1) represents the most useful but non-trivial extension to the propositional fragment from Section 3.2, the latter being virtually identical to existing probabilistic relational languages. Roughly, if we think of the propositional fragment as an instantiation of a first-order language where both the *arguments* and *values* come from finite sets, Section 4.1 can be thought of as an instantiation where only the *arguments* come from finite sets. Thus, while standard probabilistic relational languages are restricted to finite-outcome discrete distributions, the fragment from Section 4.1 could be used for a richer class of distributions.

For the future, it would be worthwhile to better relate the semantics of other infinite domain approaches and our proposal. Moreover, in this paper, we exploited symmetries over unmentioned constants, and it would be worth exploring how symmetries over instances can be additionally exploited [Van den Broeck et al., 2014, Belle, 2017].

In the last decade, weighted model counting has enabled efficient probabilistic reasoning in a number of challenging domains and over heterogeneous representations. Like classical model counting, our approach supports a general grammar where logical connectives $\{\vee, \wedge, \neg\}$ can be used freely. In this spirit, the methodology in this paper could serve as a *general framework* onto which one would encode *expressive first-order* probabilistic logical and probabilistic programming languages involving functions and quantifiers.

References

[Alberti et al., 2016] Alberti, F., Ghilardi, S., and Pagani, E. (2016). Counting constraints in flat array fragments. In *IJCAR*.

[Bacchus, 1990] Bacchus, F. (1990). *Representing and Reasoning with Probabilistic Knowledge*. MIT Press.

[Bacchus et al., 2009] Bacchus, F., Dalmao, S., and Pitassi, T. (2009). Solving #SAT and Bayesian inference with backtracking search. *JAIR*, 34:391–442.

[Barrett et al., 2009] Barrett, C., Sebastiani, R., Seshia, S. A., and Tinelli, C. (2009). Satisfiability modulo theories. In *Handbook of Satisfiability*, pages 825–885. IOS Press.

[Belle, 2017] Belle, V. (2017). Open-universe weighted model counting. In *AAAI*.

[Belle and Lakemeyer, 2011] Belle, V. and Lakemeyer, G. (2011). On progression and query evaluation in first-order knowledge bases with function symbols. In *Proc. IJCAI*.

[Belle et al., 2015] Belle, V., Passerini, A., and Van den Broeck, G. (2015). Probabilistic inference in hybrid domains by weighted model integration. In *IJCAI*.

[Boerger et al., 1997] Boerger, E., Grädel, E., and Gurevich, Y. (1997). *The classical decision problem*. Springer Verlag.

[Chavira and Darwiche, 2008] Chavira, M. and Darwiche, A. (2008). On probabilistic inference by weighted model counting. *Artif. Intell.*, 172(6-7):772–799.

[de Salvo Braz et al., 2005] de Salvo Braz, R., Amir, E., and Roth, D. (2005). Lifted first-order probabilistic inference. In *Proc. IJCAI*, pages 1319–1325.

[de Salvo Braz and O'Reilly, 2017] de Salvo Braz, R. and O'Reilly, C. (2017). Exact inference for relational graphical models with interpreted functions: Lifted probabilistic inference modulo theories. In *Proc. UAI*.

[de Salvo Braz et al., 2016] de Salvo Braz, R., O'Reilly, C., Gogate, V., and Dechter, R. (2016). Probabilistic inference modulo theories. In *Proc. IJCAI*, pages 3591–3599.

[Enderton, 1972] Enderton, H. (1972). *A mathematical introduction to logic*. Academic press New York.

[Ermon et al., 2013] Ermon, S., Gomes, C. P., Sabharwal, A., and Selman, B. (2013). Embed and project: Discrete sampling with universal hashing. In *NIPS*.

[Fagin and Halpern, 1994] Fagin, R. and Halpern, J. Y. (1994). Reasoning about knowledge and probability. *J. ACM*, 41(2):340–367.

[Farré et al., 2007] Farré, C., Nutt, W., Teniente, E., and Urpí, T. (2007). Containment of conjunctive queries over databases with null values. In *ICDT*.

[Fierens et al., 2011] Fierens, D., den Broeck, G. V., Thon, I., Gutmann, B., and Raedt, L. D. (2011). Inference in probabilistic logic programs using weighted CNF's. In *UAI*.

[Getoor and Taskar, 2007] Getoor, L. and Taskar, B., editors (2007). *An Introduction to Statistical Relational Learning*. MIT Press.

[Gilks et al., 1994] Gilks, W. R., Thomas, A., and Spiegelhalter, D. J. (1994). A language and program for complex Bayesian modelling. *The Statistician*, pages 169–177.

[Gogate and Domingos, 2011] Gogate, V. and Domingos, P. (2011). Probabilistic theorem proving. In *UAI*.

[Jaeger, 1998] Jaeger, M. (1998). Reasoning about infinite random structures with relational Bayesian networks. In *KR*.

[Laskey and da Costa, 2005] Laskey, K. B. and da Costa, P. C. G. (2005). Of starships and klingons: Bayesian logic for the 23rd century. In *UAI*.

[Milch et al., 2005a] Milch, B., Marthi, B., Russell, S. J., Sontag, D., Ong, D. L., and Kolobov, A. (2005a). BLOG: Probabilistic models with unknown objects. In *Proc. IJCAI*.

[Milch et al., 2005b] Milch, B., Marthi, B., Sontag, D., Russell, S. J., Ong, D. L., and Kolobov, A. (2005b). Approximate inference for infinite contingent Bayesian networks. In *AISTATS*.

[Nickel et al., 2016] Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

[Poole, 1997] Poole, D. (1997). The independent choice logic for modelling multiple agents under uncertainty. *Artificial intelligence*, 94(1):7–56.

[Richardson and Domingos, 2006] Richardson, M. and Domingos, P. (2006). Markov logic networks. *Machine learning*, 62(1):107–136.

[Riguzzi, 2015] Riguzzi, F. (2015). The distribution semantics is well-defined for all normal programs. In *PLP@ ICLP*.

[Russell, 2015] Russell, S. J. (2015). Unifying logic and probability. *Commun. ACM*, 58(7):88–97.

[Sanner and Abbasnejad, 2012] Sanner, S. and Abbasnejad, E. (2012). Symbolic variable elimination for discrete and continuous graphical models. In *AAAI*.

[Sato, 1995] Sato, T. (1995). A statistical learning method for logic programs with distribution semantics. In *ICLP*, pages 715–729.

[Singla and Domingos, 2007] Singla, P. and Domingos, P. M. (2007). Markov logic in infinite domains. In *UAI*.

[Srivastava et al., 2014] Srivastava, S., Russell, S. J., Ruan, P., and Cheng, X. (2014). First-order open-universe POMDPs. In *UAI*.

[Van den Broeck et al., 2014] Van den Broeck, G., Meert, W., and Darwiche, A. (2014). Skolemization for weighted first-order model counting. In *KR*.

[Wu et al., 2012] Wu, W., Li, H., Wang, H., and Zhu, K. Q. (2012). Probase: A probabilistic taxonomy for text understanding. In *Proc. Int. Conf. on Management of Data*, pages 481–492.