Open-Universe Weighted Model Counting

Citation for published version:
Belle, V 2017, Open-Universe Weighted Model Counting. in *Proceedings of The Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17).* AAAI Press, pp. 3701-3708, Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, United States, 4/02/17.
<http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/15008>

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proceedings of The Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Open-Universe Weighted Model Counting*

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

Abstract

Weighted model counting (WMC) has recently emerged as an effective and general approach to probabilistic inference, offering a computational framework for encoding a variety of formalisms, such as factor graphs and Bayesian networks. The advent of large-scale probabilistic knowledge bases has generated further interest in relational probabilistic representations, obtained by according weights to first-order formulas, whose semantics is given in terms of the ground theory, and solved by WMC. A fundamental limitation is that the domain of quantification, by construction and design, is assumed to be finite, which is at odds with areas such as vision and language understanding, where the existence of objects must be inferred from raw data. Dropping the finite-domain assumption has been known to improve the expressiveness of a first-order language for open-universe purposes, but these languages, so far, have eluded WMC approaches. In this paper, we revisit relational probabilistic models over an infinite domain, and establish a number of results that permit effective algorithms. We demonstrate this language on a number of examples, including a parameterized version of Pearl's Burglary-Earthquake-Alarm Bayesian network.

Introduction

Weighted model counting (WMC) has recently emerged as an effective and general approach to probabilistic inference, offering a computational framework for encoding a variety of formalisms, including factor graphs and Bayesian networks (Choi, Kisa, and Darwiche 2013; Chavira and Darwiche 2008). WMC generalizes #SAT, where we are to count the models of a propositional formula, in that models can be additionally accorded numeric weights, whose sum we are to compute. In particular, the encoding of graphical models to a propositional theory allows us to leverage context-specific dependencies, express hard constraints, and reason about logical equivalence. Exact solvers are based on knowledge compilation (Chavira and Darwiche 2008) or exhaustive DPLL search (Sang, Beame, and Kautz 2005); approximate ones use local search (Wei and Selman 2005) or sampling (Ermon et al. 2013; Chakraborty et al. 2014).

*This research was supported in part by the Research Foundation-Flanders (FWO-Vlaanderen).

The advent of large-scale probabilistic knowledge bases, such as Google's Knowledge Vault (Dong et al. 2014) and Microsoft's Probase (Wu et al. 2012), often containing billions of tuples and structured data extracted from the Web and other text corpora has generated enormous interest in relational probabilistic representations (Getoor and Taskar 2007). In Markov logic networks (Richardson and Domingos 2006), for example, the weighted formula:

$$1.2 \quad \forall x, y\ Smoker(x) \land Friends(x, y) \supset Smoker(y)$$

is indicative of a fairly involved Markov network obtained by grounding the formula wrt all possible values for $\{x, y\}$ and assigning a potential of 1.2 to the edges of the network. Such *template models* can be seen as modest relatives of general-purpose probabilistic logics (Bacchus 1990; Halpern 1990) in succinctly characterizing large finite graphical models, making them appropriate for reasoning and learning with big uncertain data. In practice, the semantics of template models is given in terms of the ground (propositional) theory, for which WMC and its extensions (Gogate and Domingos 2011; Van den Broeck 2013) suffice.

By construction, and indeed by design, a fundamental limitation of these template models is its *finite domain closure assumption*: the range of the quantifiers is assumed to be finite, typically the set of constants appearing in the logical theory. As argued in (Russell 2015), such a closed-world system is at odds with areas such as vision, language understanding, and Web mining, where the existence of objects must be inferred from raw data. Various syntactic and semantic devices have been proposed to address this, including probabilistic programming and infinite-state Bayesian networks (Milch et al. 2005a; 2005b). While the progress has been significant, we lack a complete picture of the expected properties from a logical perspective.

In this paper, we revisit *open-universe* (OU) template models, and develop an account of probabilistic inference by weighted model counting. Formally, our proposal will allow quantifiers to range over an *infinitary set of rigid designators* – constants that exist in all possible worlds – a common technical device from modal logic, also used elsewhere in OU applications (Srivastava et al. 2014). Such languages have eluded WMC approaches so far. As an extension, it is powerful, and can be used for features such as:

- *unknown atoms*: $\{\forall x\ (x \neq john \supset Smoker(x))\}$ says that infinitely many individuals other than John are smokers, while also leaving open whether John is a smoker;

- *unknown values*: $\{\forall x(Canary(x) \supset \neg Color(x, black))\}$ says the color of canaries is anything but black;
- *varying sets of objects*: $\{\forall x(Thing(x) \supset \phi(x))\}$ says that property ϕ is true of all things, but the set of (possibly infinite) things can vary depending on the world;
- *closed-world assumption:* $\forall x((x = john) \supset Smoker(x)) \wedge \forall x((x \neq john) \supset \neg Smoker(x))$ says that John is the only one in the universe who smokes.

In general, such a first-order logical language is undecidable. However, we show that when restricted to universally quantified clauses, as is usual in the literature on template models, a finite version of the ground knowledge base suffices for reasoning purposes.

When probabilities are further accorded to infinitely many atoms, the natural question is in which sense are conditional probabilities correct? Prior work has focused on various semantics with infinitely many random variables (Singla and Domingos 2007; Doshi et al. 2011; Kersting and Raedt 2000), based on topological orderings and locality properties. In this work, we contribute a definition of conditional probabilities that is coherent wrt first-order logical entailment: this has the benefit that no syntactical constraints on the encoding are necessary. We show its application in a number of examples, including a parametrized version Pearl's (1988) Alarm Bayesian network discussed in (Russell 2015).

Overall, the main contributions are: results on satisfiability and validity for first-order clauses over an infinite universe, a coherent account of weighted model counting in an OU setting, the leveraging of existing algorithms for the clausal representation, and finally, numerous examples that have been verified using existing WMC solvers.

Proof sketches for all the main claims are found in the appendix.

Preliminaries

Logical Background

Language: We let $\mathcal{L}$ be a first-order language with equality, relational symbols $\{P(x), Q(x, y), R(x, y, z), \ldots, P'(x), \ldots\}$ of every arity, variables $\{x, y, z, \ldots\}$, and a countably infinite set of *rigid designators* or *names*, say, the set of natural numbers $\mathbb{N}$, serving as the domain of discourse for quantification. But for presentation purposes, we often use identifiers such as Latin alphabets $(a, b, \ldots)$ and proper names $(john, jane, \ldots)$. Together with equality, names essentially realize an infinitary version of the unique-name assumption.[1]

The set of (ground) atoms is obtained as:[2]

$$\text{VARS} = \{P(a_1, \ldots, a_k) \mid P \text{ is a predicate, } a_i \text{ a name}\}.$$

[1]In general, this does not rule out capturing uncertainty about the identity of objects; see (Giacomo, Lespérance, and Levesque 2011; Srivastava et al. 2014), for example.

[2]Because equality is treated separately, atoms and clauses do not include equalities.

Semantics: A $\mathcal{L}$-model M is a $\{0, 1\}$ assignment to the elements of VARS. Using $\models$ to denote satisfaction, the semantics for $\phi \in \mathcal{L}$ is defined as usual inductively, but with equality as identity: $M \models (a = b)$ iff a and b are the same names, and quantification understood substitutionally over all names in $\mathbb{N}$: $M \models \forall x \phi(x)$ iff $M \models \phi(a)$ for all $a \in \mathbb{N}$. We say that ϕ is *valid* iff for every $\mathcal{L}$-model M, $M \models \phi$.

We will refer to atoms also as propositions, and a ground formula also as a propositional formula. We will use p, q, e to refer to atoms, α, ϕ to refer to ground formulas, and Δ to refer to closed formulas with quantifiers.

Remark: It is worth noting that the *compactness* property that holds for classical first-order logic *does not hold* in general when the domain is fixed and infinite (Levesque 1998):

$$\{\exists x P(x), \neg P(1), \neg P(2), \ldots\}$$

is an unsatisfiable theory whose every finite subset is indeed satisfiable.

Propositional Fragments: While $\mathcal{L}$ and its models are infinitary, satisfiability checkers and model counters work with a finite language, built from a finite set of symbols. To enable this, if ϕ is a propositional formula, we use VARS(ϕ) to refer to the propositions in ϕ, and $\mathcal{L}(\phi)$ to refer to the finite propositional language built from VARS(ϕ) and connectives $\{\neg, \vee, \wedge\}$. The understanding here is that a $\mathcal{L}(\phi)$-model M is a $\{0, 1\}$ assignment to the elements of VARS(ϕ).

Example 1 : If $\phi = P(a) \vee P(b)$, then $\text{VARS}(\phi) = \{P(a), P(b)\}$, $\mathcal{L}(\phi) = \{P(a), P(b), \neg P(a), \ldots\}$, and there are 4 $\mathcal{L}(\phi)$-models, 3 of which satisfy ϕ.

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 $\mathcal{L}(\phi)$-model, and w is a *weight function.* Usually, the weight function is *factorized*. For example, in classical WMC (Chavira and Darwiche 2008), suppose $v, \overline{v} : \text{VARS}(\phi) \to \mathbb{R}^+$ map atoms to positive reals, then:

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

where $p \in M$ are those atoms that are true at M, and $p \notin M$ are those that are false. A relational extension of WMC (Van den Broeck 2013), dubbed WFOMC, assumes $v, \overline{v}$ map predicate symbols from ϕ to $\mathbb{R}^+$. Then:

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

where PRED maps an atom to its predicate.

WMC has emerged as a basic computational framework for encoding many formalisms. For example, Markov logic networks (Richardson and Domingos 2006), typically defined as a set of weighted (finite domain) first-order formulas, *e.g.,* $\{(w, \alpha(x, \ldots, y))\}$ can be cast a WFOMC problem (Van den Broeck 2013) by introducing a new predicate

$P(x,\ldots,y)$, letting the theory be the sentence $P(x,\ldots,y) \equiv \alpha(x,\ldots,y)$, and defining $v(P) = e^w$ and $\overline{v}(P) = 1$, and letting $v(Q) = \overline{v}(Q) = 1$ for all predicates Q in α.

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:

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

Logical Entailment

Before addressing model counting, we first turn to the basic question of logical entailment.

Representation

Unfortunately, reasoning in the full language is undecidable (Levesque 1998), but fortunately, when the logical encoding (of a probabilistic formalism) is given as a set of clauses, as is usual in template models, we will be able to establish some results on logical entailment. We begin as follows:

Definition 2: An *acceptable* equality is of the form $x = a$, where x is any variable and a any name. Let e range over formulas built from acceptable equalities and connectives $\{\neg, \vee, \wedge\}$. Let c range over quantifier-free disjunctions of (possibly non-ground) atoms. Let $\forall\phi$ mean the universal closure of ϕ. A formula of the form $\forall(e \supset c)$ is called a $\forall$-clause. A knowledge base (KB) Δ is *acceptable* if it is a finite non-empty set of $\forall$-clauses. The *rank* of Δ is the maximum number of variables mentioned in any $\forall$-clause in Δ.

To that end, an encoding is taken to be of the form (Δ, w) where Δ is an acceptable KB, and w is a weight function.

Example 3: For example, $\forall((x \neq \mathit{john} \wedge x \neq \mathit{jane}) \supset \mathit{Smoker}(x) \vee \mathit{Alcoholic}(x))$ is a $\forall$-clause with rank 1 and says that everybody other than John and Jane are either smokers or alcoholics. (The first section mentioned a few examples of first-order formulas that are also definable as $\forall$-clauses.)

Note that $\forall$-clauses are logically equivalent to a possibly infinite set of ground clauses, *e.g.,* $\forall(P(x) \vee Q(x))$ is equivalent to $\{P(1) \vee Q(1), P(2) \vee Q(2), \ldots\}$.

Grounding: A ground theory is obtained from Δ by substituting variables with names. Suppose θ denotes a substitution. For any set of names $C \subseteq \mathbb{N}$, we write $\theta \in C$ to mean substitutions are only allowed wrt the names in C. Finally, we define the following terms:

- $\mathrm{GND}(\Delta) = \{c\theta \mid \forall(e \supset c) \in \Delta \text{ and } \models e\theta\}$;
- For $k \geq 0$, $\mathrm{GND}(\Delta, k) = \{c\theta \mid \forall(e \supset c) \in \Delta, \models e\theta, \theta \in K\}$, where K is the set of names mentioned in Δ plus k (arbitrary) new ones;
- $\mathrm{DC}(\Delta) = \mathrm{GND}(\Delta, 0)$;
- $\mathrm{OU}(\Delta) = \mathrm{GND}(\Delta, k)$ where k is the rank of Δ.

Example 4: Recall that because equality is understood as identity for names, formula $\mathit{john} \neq \mathit{jane}$ is valid, and formulas $\mathit{john} = \mathit{jane}$ and $\mathit{john} \neq \mathit{john}$ are unsatisfiable. So, given $\Delta = \forall(x \neq \mathit{john} \supset \mathit{Smoker}(x))$, we have $\mathrm{GND}(\phi) = \{\mathit{Smoker}(\mathit{jane}), \mathit{Smoker}(\mathit{bob}), \ldots\}$.

Example 5: Suppose $\Delta = \{\forall(P(x)), Q(a)\}$. Then $\mathrm{DC}(\Delta) = \mathrm{GND}(\Delta, 0) = \{P(a), Q(a)\}$, and $\mathrm{OU}(\Delta) = \mathrm{GND}(\phi, 1) = \{P(a), P(b), Q(a)\}$. (Here, b is picked arbitrarily.)

Proposition 6: *For any acceptable KB* Δ, $\mathrm{OU}(\Delta)$ *is finite.*

The understanding here is that $\mathrm{DC}(\Delta)$ represents the *domain closure assumption* built into template models. The domains in Markov logic networks and WFOMC, for example, are defined in terms of the Herbrand universe: basically, the constants mentioned in a (function-free) KB (Richardson and Domingos 2006; Van den Broeck 2013). Going further, $\mathrm{OU}(\Delta)$ will play an important role in realizing logical properties in an OU setting. It should be clear that it represents a relatively small blowup to $\mathrm{DC}(\Delta)$ by including a few extra constants (= Δ's rank). In the worst-case, we have:

Proposition 7: *Suppose* Δ *mentions c names, has n* $\forall$-*clauses, each mentioning at most m predicates, and with rank k. Then, the total number of atoms mentioned in* $\mathrm{DC}(\Delta)$ *is* $O(n \cdot m \cdot c^k)$*, and the total in* $\mathrm{OU}(\Delta)$ *is* $O(n \cdot m \cdot (c + k)^k)$.

Results

The first hurdle is addressing the compactness property, or the lack thereof. Such a property is useful to draw conclusions about a possibly infinite theory in terms of its finite subsets. As mentioned, $\mathcal{L}$ does not enjoy this property, but fortunately, acceptable KBs do. For our first result, we have:

Theorem 8: *Suppose* Δ *is acceptable. Then* $S = \mathrm{GND}(\Delta)$ *is satisfiable iff every finite subset of* S *is satisfiable.*

This is because $\mathrm{GND}(\Delta)$ is a (possibly infinite) propositional theory, and we can lift compactness from propositional logic.

Next, for our main result, we establish that entailment can eschew the need to appeal to an infinitary theory:

Theorem 9: *Suppose* Δ *is as above, and* α *is any propositional formula. Then*

$$\models \Delta \supset \alpha \quad \textit{iff} \quad \mathrm{OU}(\Delta \wedge \neg\alpha) \textit{ is unsatisfiable.}$$

The proof is long, but can be approached using 2 lemmas:

Lemma 10: *Suppose* Δ*'s rank is k. If* $\mathrm{GND}(\Delta, k) \wedge \alpha$ *is satisfiable, then so is* $\mathrm{GND}(\Delta, j) \wedge \alpha$ *for all* $j \geq k$.

Lemma 11: $\Delta \models \alpha$ *iff* $\mathrm{OU}(\Delta) \models \alpha$.

The first argues that grounding Δ wrt additional names does not affect satisfiability. The crux of the argument lies in showing that names not mentioned in $\Delta \wedge \alpha$ behave identically (in a suitable formal sense). The second argues that for entailment $\mathrm{OU}(\Delta)$ suffices, which is then used for a refutation-based claim in Theorem 9.

Putting Theorem 9 and Proposition 6 together, we have:

Corollary 12: *Suppose* Δ *and* α *are as above, then* $\Delta \models \alpha$ *can be verified in classical (finitary) propositional logic.*

Example 13: Let Δ be the union of:

- $\forall(\mathit{Smoker}(x) \wedge \mathit{Friends}(x, y) \supset \mathit{Smoker}(y))$
- $\mathit{Smoker}(\mathit{john})$
- $\forall y(\mathit{Friends}(\mathit{john}, y))$

Let the query be $\alpha = Smoker(jane)$. Since everybody is friends with John, a smoker, we have $\models \Delta \supset \alpha$.

To see Theorem 9 in action, consider $\mathrm{OU}(\Delta \wedge \neg\alpha) = \mathrm{GND}(\Delta \wedge \neg\alpha, 2)$. On applying all possible substitutions wrt the names mentioned and 2 new ones, $\mathrm{OU}(\Delta \wedge \neg\alpha)$ would include: $(Smoker(john) \wedge Friends(john, jane)) \supset Smoker(jane)$, $Smoker(john)$, $Friends(john, jane)$ but also $\neg\alpha = \neg Smoker(jane)$. Indeed $\mathrm{OU}(\Delta \wedge \neg\alpha)$ is unsatisfiable.

In the above example, when grounding, the new names were not needed. To see why names from outside the KB and the query may be essential, consider:

Example 14: Let α be as above, but Δ be the union of:

- $\forall(x \neq jane \supset Smoker(x))$
- $\forall(Friends(x, y))$
- $\forall(Smoker(x) \wedge Friends(x, y) \supset Smoker(y))$

Here $\models \Delta \supset \alpha$. Argument: because someone other than Jane, say John, is declared to be a smoker and since he is Jane's friend, she must also be a smoker. However, $\mathrm{DC}(\Delta \wedge \neg\alpha)$ is satisfiable: only using the names mentioned in $\Delta \wedge \neg\alpha$ is insufficient. In contrast, $\mathrm{OU}(\Delta \wedge \neg\alpha)$ is indeed unsatisfiable.

Weighted Model Counting

We now develop an OU account of WMC. Prior work has considered various semantics with high-dimension probability spaces, appealing to topological orderings and such. We contribute a definition to this literature that is logically motivated, based on the below (informally stated) properties:

1. *When the query is believed*, it has probability 1.
2. *When the query contradicts what is believed*, it has probability 0.
3. *When the query mentions unknown individuals*, these individuals are interchangeable with other unknowns.

Our definition satisfies these features in a first-order setting, and will further leverage Theorem 9 in only using $\mathrm{OU}(\Delta)$. The benefit is that no assumptions are needed about the encoding (Δ, w) because only a finite weighted propositional theory is instantiated wrt a given query.[3] Formally:

Definition 15: Suppose Δ is an acceptable KB, w a weight function, and $\{Q, E\}$ propositional formulas.[4] Then, the WMC of (Δ, w) in an OU setting is defined as:

$$\mathrm{WMCOU}(\Delta, w) = \mathrm{WMC}(\mathrm{OU}(\Delta), w).$$

The probability of Q given E wrt (Δ, w) is defined as:

$$\Pr(Q \mid E, \Delta, w) = \mathrm{WMCOU}(\Delta \wedge Q \wedge E, w) / \mathrm{WMCOU}(\Delta \wedge E, w).$$

We often write $\Pr(Q \mid \Delta, w)$ to mean $E = true$. Algorithm 1 provides a pseudocode for Pr in terms of WMC.

Its correctness is stated as follows:

Theorem 16 : *Suppose Δ is any acceptable KB, α any propositional formula, w any predicate-level weight function. Then*

(a) $\Pr(\alpha \mid \Delta, w) = 1$ *if* $\models \Delta \supset \alpha$*;*

(b) $\Pr(\alpha \mid \Delta, w) = 0$ *if* $\models \Delta \supset \neg\alpha$*;*

(c) $\Pr(\alpha \mid \Delta, w) = \Pr(\alpha^\bullet \mid \Delta, w)$ *for any bijection $\bullet$ from $\mathbb{N}$ to $\mathbb{N}$ such that it maps names from Δ to themselves and otherwise arbitrary.*

If Δ was propositional, classical WMC would satisfy (a) and (b), *e.g.,* if $\phi \models \alpha$ then $\phi \equiv \phi \wedge \alpha$ and therefore, ϕ and $\phi \wedge \alpha$ have the same model counts. Our definition not only lifts this property to first-order entailment but also enjoys (c) which is relevant with infinitely many unknowns.[5]

Example 17: To see this in action, let Δ be the union of:

- $\forall(Smoker(x) \vee Alcoholic(x))$
- $\forall(x \neq john \supset \neg Smoker(x))$

Suppose $v, \bar{v}$ map all predicates to 1. Consider $\alpha_1 = Alcoholic(jane)$ vs $\alpha_2 = Smoker(jane)$. Note that $\Delta \models \alpha_1$ but $\Delta \models \neg\alpha_2$. Observe $\mathrm{OU}(\Delta \wedge \alpha_1)$ is the conjunction of:

- $(Smoker(john) \vee Alcoholic(john)) \wedge (Smoker(jane) \vee Alcoholic(jane)) \wedge (Smoker(bob) \vee Alcoholic(bob))$
- $\neg Smoker(jane) \wedge \neg Smoker(bob) \wedge Alcoholic(jane)$

where *bob* is chosen arbitrarily. Clearly models can vary in how they interpret $Smoker(john) \vee Alcoholic(john)$, and so we obtain $\mathrm{WMCOU}(\Delta \wedge \alpha_1) = 3$, and also, $\Pr(\alpha_1 \mid \Delta, w) = 1$. However, $\Delta \wedge \alpha_2$ is inconsistent, and indeed $\mathrm{WMCOU}(\Delta \wedge \alpha_2) = 0$ leading to $\Pr(\alpha_2 \mid \Delta, w) = 0$.

Finally, let $\beta = Alcoholic(bob)$. We note $\Pr(\beta \mid \Delta, w) = 1$. Let $\bullet$ map *john* to itself, otherwise arbitrary: say, it maps *bob* to *jane*. So $\beta^\bullet = Alcoholic(jane)$. Indeed $\Pr(\beta^\bullet \mid \Delta, w) = 1$.

Example 18: We now consider a parameterized version of Pearl's (1988) Alarm Bayesian network. In the classic version, a burglary or an earthquake can trigger your house alarm, and hearing that your neighbors would probably call you. Russell (2015) sketches a richer variant, which we call *AlarmX*: imagine a large set of regions, and in each region, there are any number of houses. In region r, the likelihood of an earthquake and the likelihood of a burglary in house h are Bernoulli random variables. Roughly, imagine rules like:

.003 $Burglary(r, h)$.
.002 $Earthquake(r)$.
.8 $Alarm(r, h) \Leftarrow Burglary(r, h), \neg Earthquake(r)$.

Consider the following set-up. Let Δ represent the above parameterized Bayesian network as a set of $\forall$-clauses, using the usual semantics for encoding Bayesian networks

[3] While the definition is coherent wrt entailment over an infinite number of atoms, it is conceivable that other notions are possible, such as those based on infinitely many random variables (Singla and Domingos 2007).

[4] For technical reasons, we assume here that Q and E mention the same set of names. This is without loss of generality: let $\gamma = \{p \vee \neg p \mid \text{atom } p \text{ is mentioned in } Q \wedge E\}$, replace Q with $Q' = Q \wedge \gamma$, and E with $E' = E \wedge \gamma$. Of course, $Q \equiv Q'$ and $E \equiv E'$, and clearly Q' and E' mention the same set of names.

[5] Readers may observe that we stipulate a predicate-level weight function. As it turns out, the weight function is irrelevant for (a) and (b). However, (c) would not hold, if, for example, given $\Delta = \{\forall(Smoker(x))\}$, w maps $Smoker(bob)$ to a higher weight than $Smoker(rob)$ even though both of them are unknown. Thus, factorization at the level of predicates is one way to enable that unknown instances of a predicate are considered equally likely.

Algorithm 1 $\Pr(Q \mid E, \Delta, w)$

$\alpha_1 = \Delta \wedge Q \wedge E, \alpha_2 = \Delta \wedge E$
$C = names(\alpha_1)$, $d = rank(\Delta)$
$C' = C \cup \{$ d new names $\}$
$\alpha'_i =$ ground α_i wrt C'
return $\text{WMC}(\alpha'_1)/\text{WMC}(\alpha'_2)$

as logical theories (Chavira and Darwiche 2008). However, the domain of the quantifiers will be unrestricted in Δ: we will not stipulate the list of regions, etc. Suppose we now provide the evidence $E = Alarm(region1, house2)$. Intuitively, we would expect the following properties in an infinite instantiation: (a) the probability of $Q1 = Burglary(region1, house2)$ is $\geq$ the probability of $Q2 = Burglary(a, b)$ for every $\langle a, b\rangle \neq \langle region1, house2\rangle$; (b) the probability of $Q3 = Earthquake(region1)$ is $\geq$ the probability of $Q4 = Earthquake(c)$ for every $c \neq region1$. As it turns out, our definition confirms this intuition:

Theorem 19: *Let Δ, E and Qs be as above, with $a = c = region2$ and $b = house3$. Let w be a predicate-level (strictly positive) weight function. Then* $\Pr(Q1 \mid E, \Delta, w) \geq \Pr(Q2 \mid E, \Delta, w)$*, and* $\Pr(Q3 \mid E, \Delta, w) \geq \Pr(Q4 \mid E, \Delta, w)$.

Lifted Inference

The WMC formulation works with the ground propositional theory. But just as first-order resolution is equivalent to a large number of propositional resolution steps, there are a number of extensions to WMC that leverage the first-order structure of template models by lumping together classes of objects (Poole 2003; de Salvo Braz, Amir, and Roth 2005; Gogate and Domingos 2011; Beame et al. 2015), referred to as *lifted inference*. A crisp formal claim is this (Van den Broeck 2013): an algorithm is said to be *domain-lifted* when it runs in time polynomial in the size of the domain.

Space precludes a discussion on these algorithms. As an example, observe that on grounding $\alpha = \forall (Smoker(x) \vee Alcoholic(x))$ wrt the domain $\{a\}$ the model count is 3, on grounding α wrt $\{a, b\}$ the model count is 3^2, and so on. Lifted systems recognize that the model count of α wrt the domain C is $3^{|C|}$, without the need for explicit grounding.

Unfortunately, different from standard template models, in the OU setting, quantification ranges over an infinite set, and equality is understood as identity over these infinitely many individuals. So, we will be interested in two questions:

- can these KBs be put in a form that is acceptable as input to lifted algorithms in a correctness preserving way, and
- can we inherit the polynomial time guarantees of domain-lifted algorithms where available?

We answer in the affirmative for both. The key construction will be to normalize acceptable KBs wrt the appropriate finite set of names required by the OU semantics.

Definition 20: Suppose Δ is acceptable, k its rank, and B the names mentioned in Δ. Let $\mathbb{N} \supset C \supseteq B$ be any finite set such that $|C| = |B| + k$. Then putting Δ in *equality-free normal form* (ENF) wrt C is achieved by recurring the transformations:

1. If $\forall (e \supset c) \in \Delta$, then c does not mention any names: that is, convert $\forall (e \supset c(a))$ to $\forall ((e \wedge z = a) \supset c(z))$ where z is a fresh variable not used in e.
2. If $\forall (e \supset c) \in \Delta$, then eliminate e by introducing sorts:
   (a) Transform $\forall (((x = a_1 \vee \ldots \vee x = a_n) \wedge e) \supset c)$ to $\forall [\forall x \in S\, (e \supset c)]$ where $S = \{a_1, \ldots, a_n\}$.
   (b) Transform $\forall (((x \neq a_1 \wedge \ldots \wedge x \neq a_n) \wedge e) \supset c)$ to $\forall [\forall x \in S\, (e \supset c)]$ where $S = C - \{a_1, \ldots, a_n\}$.

Proposition 21: *When Δ is converted to ENF, then (i) ENF(Δ) does not mention equality, and is a conjunction of formulas of the form $\forall x_1 \in S_1, \ldots, x_n \in S_n\ c$ where c is a clause not mentioning names; (ii) the sort S introduced in 2(a) is such that $S \subseteq B$; and (iii) the sort S introduced in 2(b) is such that $S = C - S'$, where $S' \subseteq B$.*

The sentence ENF(Δ) is equality-free, function-free and over finite sorts, which is precisely the fragment expected by lifted algorithms. For starters, it is correctness preserving:

Theorem 22: $\text{WMC}(\text{ENF}(\Delta), w) = \text{WMCOU}(\Delta, w)$ *for any predicate-level w.*

It is shown in (Van den Broeck 2013) that when first-order formulas mention only 2 variables, then there is a domain-lifted algorithm for their WFOMC task. We leverage this:

Corollary 23: *Suppose Δ, C and w are as above. Suppose* $\text{ENF}(\Delta) = \bigwedge_i \forall\, c_i$ *is such that c_i only uses at most 2 logical variables. Then* $\text{WMCOU}(\Delta, w) = \text{WMC}(\text{ENF}(\Delta), w)$ *is computable in time polynomial in $|C|$.*

Investigations on other classes of first-order formulas is an area of ongoing research (Beame et al. 2015), and owing to Theorem 22, we expect these to carry over.

Preliminary Evaluations

The simplicity of the OU semantics, that of grounding KBs wrt a few extra constants, makes it straightforward to solve OU problems using existing (unmodified) WMC software, either by Algorithm 1 or by ENF(Δ) with a grounder. For a range of problems, we have verified that correct answers wrt the OU semantics are returned using the C2D WMC solver (Darwiche 2004), the WFOMC lifted WMC solver (Van den Broeck 2013), and the ProbLog probabilistic programming language (Fierens et al. 2011).[6]

In this section, we investigate two empirical questions (for problems in ENF). First, how expensive is handling the OU case? Second, how does a WMC approach to OU problems compare against a sampling-based OU approach, as embedded in BLOG,[7] for example (Milch et al. 2005a)?

To answer these questions, we chose 5 problems that emphasize an OU environment. The particulars of the problems are discussed below, and the empirical behavior is presented in Figure 1. Experiments were run on OS X, using a 1.3 GHz Intel i5 processor with 4GB RAM. ProbLog was used for computing probabilities (OU with ENF or otherwise). BLOG v0.2 was run with the default settings.

[6]Recent installments of ProbLog are based on a logical translation of probabilistic rules to weighted CNFs (Fierens et al. 2011).

[7]BLOG is a strong baseline as it is designed and optimized for OU applications, is well-studied and approachable for modeling relational template models owing to its first-order syntax.

semantics	*AlarmR*	*Smoker1*	*Smoker2*
OU	0.194	0.172	0.182
DC	0.192	0.170	0.175

language	*AlarmR*	*AlarmX*	*Grades*
WMCOU	0.194	0.2	0.36
BLOG	0.782	0.85	6.0

Figure 1: DC vs OU (top) and WMCOU vs BLOG (bottom). Time reported is in seconds, and is averaged over 10 runs.

Domain Closure (DC) vs OU

Let us remark that for OU problems, DC may not give the correct answers. Nonetheless, we considered Examples 13 and 14 (*Smoker1* and *Smoker2* in Figure 1 respectively). Recall that DC would return the correct answer for the first, but not the second. These are then compared to WMCOU.

Next, we reconsider the classical Alarm Bayesian network (*AlarmR* in Figure 1), now with two neighbors, where we are interested in the probability that there was a burglary given that both persons called. This is a DC setting. For the OU case, we leave the neighbor predicate open, that is, it is not known how many neighbors there are, but of these, two distinct people call after hearing the alarm.

The execution times reported in Figure 1 indicate that handling the OU case has only a marginal impact on performance, as is expected (and desired).[8]

WMCOU vs BLOG

We contrasted the two approaches on *AlarmR*, but also on *AlarmX*. In both cases, WMCOU is notably faster. A more glaring difference is observed when we consider the *Grades* problem from (Heckerman, Meek, and Koller 2004). Briefly, we are to predict the probability of a student's grade, given the courses she takes, the difficulty of these courses, the student's intelligence level, and so on. We enable an OU setting here by leaving the set of students open, and their existence is inferred from the evidence. The problem is known to have a non-trivial relational skeleton, and so it seems that a WMC-based approach fares well owing to a SAT solver's ability to handle complex logical dependencies.

Lineage and Conclusions

There are several recent instrumental papers focusing on OU models. We mention the ones that are most closely related, and refer readers to (Milch et al. 2005a) and references therein on historical developments. Extensions of template models to infinite domains are, of course, the closest in spirit: 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 independency 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. (Approaches not based on template models include, *e.g.,* (Welling, Porteous, and Bart 2007), that view infinite-state BNs as hierarchical Dirichlet processes.) Such template models can also be enabled by means of programming languages, which include both declarative approaches *e.g.,* (Kersting and Raedt 2000; Fierens et al. 2011) that often support derivations that are infinitely long, as well as procedural ones, *e.g.,* BLOG (Milch et al. 2005a) that also allows identity uncertainty in addition to open domains. There has also been recent work on handling existential quantifiers and exchangeable atoms for model counting (Van den Broeck, Meert, and Darwiche 2014; Niepert and Van den Broeck 2014) that are related to some of the ideas in the OU semantics.

From the representation side, our language and KB syntax is based on modal logic with rigid designators (Levesque and Lakemeyer 2001; Lakemeyer and Levesque 2002; Belle and Lakemeyer 2011). Our results on logical entailment, in particular, revisit and simplify the developments in (Belle and Lakemeyer 2011). Beyond this, there is a large body of work on handling unknowns for query entailment in knowledge representation and database theory, *e.g.,* see (Van der Meyden 1998; Libkin 2016; Giacomo, Lespérance, and Levesque 2011) and references therein. Finally, in the context of WMCOU considering new constants, studying the model count of a first-order formula as a function of the domain size is of long-standing interest in database theory (Dalvi 2006; Beame et al. 2015; Fagin 1976).

The OU semantics of this paper contributes to the above literature, and offers the following salient features:

1. no syntactic stipulations (*e.g.,* stratification constraints) on the weighted theory (Δ, w) outside of the usual clausal representation for Δ;
2. if a parameterized representation serves as a proxy for a large but finite domain (*e.g.,* family trees, language models), the semantics necessitates grounding only wrt the constants in the query at hand (plus KB's rank);
3. conditional probabilities are coherent wrt logical entailment over a full first-order language;
4. the ability to leverage existing WMC technology, including lifted inference that can (sometimes) lead to exponential improvements.

We think these 4 properties can put OU applications within the reach of weighted model counters. The simplicity of the propositional apparatus and the separation of the weight function from the encoding has led to the popularity of WMC, and in this vein, we hope that WMCOU can serve as an assembly language for OU formalisms.

As for future work, it would be worthwhile to better understand the mathematical relationships between WMCOU and other OU semantics.

[8]Execution times with C2D and WFOMC, along with scalability numbers for increasing domain sizes, are left to an extended report.

References

Bacchus, F. 1990. *Representing and Reasoning with Probabilistic Knowledge*. MIT Press.

Beame, P.; Van den Broeck, G.; Gribkoff, E.; and Suciu, D. 2015. Symmetric weighted first-order model counting. In *PODS*, 313–328. ACM.

Belle, V., and Lakemeyer, G. 2011. On progression and query evaluation in first-order knowledge bases with function symbols. In *Proc. IJCAI*, 255–260.

Chakraborty, S.; Fremont, D. J.; Meel, K. S.; Seshia, S. A.; and Vardi, M. Y. 2014. Distribution-aware sampling and weighted model counting for sat. *AAAI*.

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

Choi, A.; Kisa, D.; and Darwiche, A. 2013. Compiling probabilistic graphical models using sentential decision diagrams. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*. Springer. 121–132.

Dalvi, N. 2006. Query evaluation on a database given by a random graph. In *Database Theory–ICDT 2007*. Springer. 149–163.

Darwiche, A. 2004. New advances in compiling CNF to decomposable negation normal form. In *ECAI*, 328–332.

de Salvo Braz, R.; Amir, E.; and Roth, D. 2005. Lifted first-order probabilistic inference. In *Proc. IJCAI*, 1319–1325.

Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; and Zhang, W. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Knowledge discovery and data mining*.

Doshi, F.; Wingate, D.; Tenenbaum, J. B.; and Roy, N. 2011. Infinite dynamic bayesian networks. In *ICML*, 913–920.

Ermon, S.; Gomes, C. P.; Sabharwal, A.; and Selman, B. 2013. Embed and project: Discrete sampling with universal hashing. In *NIPS*, 2085–2093.

Fagin, R. 1976. Probabilities on finite models. *The Journal of Symbolic Logic* 41(01):50–58.

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*, 211–220.

Getoor, L., and Taskar, B., eds. 2007. *An Introduction to Statistical Relational Learning*. MIT Press.

Giacomo, G. D.; Lespérance, Y.; and Levesque, H. J. 2011. Efficient reasoning in proper knowledge bases with unknown individuals. In *IJCAI*, 827–832.

Gogate, V., and Domingos, P. 2011. Probabilistic theorem proving. In *Proc. UAI*, 256–265.

Halpern, J. 1990. An analysis of first-order logics of probability. *Artificial Intelligence* 46(3):311–350.

Heckerman, D.; Meek, C.; and Koller, D. 2004. Probabilistic models for relational data. Technical report, Technical Report MSR-TR-2004-30, Microsoft Research.

Jaeger, M. 1998. Reasoning about infinite random structures with relational bayesian networks. In *KR*, 570–581.

Kersting, K., and Raedt, L. D. 2000. Bayesian logic programs. In *ILP Work-in-progress reports*.

Lakemeyer, G., and Levesque, H. J. 2002. Evaluation-based reasoning with disjunctive information in first-order knowledge bases. In *Proc. KR*, 73–81.

Laskey, K. B., and da Costa, P. C. G. 2005. Of starships and klingons: Bayesian logic for the 23rd century. In *UAI*, 346–353.

Levesque, H., and Lakemeyer, G. 2001. *The logic of knowledge bases*. The MIT Press.

Levesque, H. J. 1998. A completeness result for reasoning with incomplete first-order knowledge bases. In *Proc. KR*, 14–23.

Libkin, L. 2016. SQL's three-valued logic and certain answers. *ACM Trans. Database Syst.* 41(1):1:1–1:28.

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*, 1352–1359.

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*.

Niepert, M., and Van den Broeck, G. 2014. Tractability through exchangeability: A new perspective on efficient probabilistic inference. In *AAAI*, 2467–2475.

Pearl, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann.

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

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

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

Sang, T.; Beame, P.; and Kautz, H. A. 2005. Performing bayesian inference by weighted model counting. In *AAAI*, 475–482.

Singla, P., and Domingos, P. M. 2007. Markov logic in infinite domains. In *UAI*, 368–375.

Srivastava, S.; Russell, S. J.; Ruan, P.; and Cheng, X. 2014. First-order open-universe pomdps. In *UAI*, 742–751.

Van den Broeck, G.; Meert, W.; and Darwiche, A. 2014. Skolemization for weighted first-order model counting. In *KR*.

Van den Broeck, G. 2013. *Lifted Inference and Learning in Statistical Relational Models*. Ph.D. Dissertation, KU Leuven.

Van der Meyden, R. 1998. Logical approaches to incomplete information: A survey. In *Logics for databases and information systems*. Springer. 307–356.

Wei, W., and Selman, B. 2005. A new approach to model counting. In *Theory and Applications of Satisfiability Testing*, 96–97. Springer.

Welling, M.; Porteous, I.; and Bart, E. 2007. Infinite state bayes-nets for structured domains. In *NIPS*, 1601–1608.

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*, 481–492.

Appendix

Proof of Proposition 6

Proof: Since Δ is finite, and the number of quantifiers in a $\forall$-clause is finite, the set of names considered for the grounding is finite, leading to only finitely many substitutions for each $\forall$-clause in Δ. ∎

Proof of Proposition 7

Proof: For each $\forall$-clause, given a rank of k and c constants, there are c^k possible substitutions, each of which result in m atoms. In the OU case, we will consider $c + k$ constants, by definition. ∎

Proof of Theorem 8

Proof: By construction, $c\theta \in \mathrm{GND}(\Delta)$ is a ground clause, and so $\mathrm{GND}(\Delta)$ is essentially a (possibly infinite) propositional theory, over a (possibly infinite) vocabulary. Compactness from propositional logic then applies. ∎

Proof of Theorem 9

The proof will need two lemmas. Before presenting the lemmas, suppose $\bullet$ is a bijection from $\mathbb{N}$ to $\mathbb{N}$. Then for any formula β, we write $\beta^\bullet$ to mean that the names a appearing in β are replaced by $a^\bullet$.

We first prove Lemma 10.

Proof: Let $\Gamma = \mathrm{GND}(\Delta, k) \wedge \alpha$ and $\Gamma' = \mathrm{GND}(\Delta, j) \wedge \alpha$. Without loss of generality, we assume that the names mentioned in $\mathrm{GND}(\Delta, j)$ is the union of the names mentioned in $\mathrm{GND}(\Delta, k)$ and $j-k$ new ones. Notation: let $names(\beta)$ be the set of names mentioned in β.

Suppose, by assumption, $M \models \Gamma$. Construct M':

1. for every atom p mentioned in Γ, let $M'[p] = M[p]$;
2. for every atom p not mentioned in Γ', let $M'[p] = M[p]$;
3. for every atom p mentioned in $\Gamma' - \Gamma$, or more precisely, for $c\theta \in \Gamma' - \Gamma$, do as follows.

   Note that $c\theta$ mentions at most k names not appearing in Δ. By construction, since $c\theta \in \Gamma' - \Gamma$, $c\theta$ mentions names in $names(\Gamma') - names(\Gamma)$, say l of them. But because of this, $c\theta$ does not mention at least l names from $names(\Gamma) - names(\Delta)$. Then let $\bullet$ be a bijection from $\mathbb{N}$ to $\mathbb{N}$ that swaps every name from $(names(\Gamma') - names(\Gamma)) \cap names(c\theta)$ with l names from $names(\Gamma) - names(\Delta)$ but not appearing in $names(c\theta)$, and maps the rest to themselves.

   By construction, $c\theta \in \Gamma' - \Gamma$ because there is a $\forall$-clause $\forall(e \supset c) \in \Delta$ and $\models e\theta$. A simple induction shows that $\models e\theta$ iff $\models (e\theta)^\bullet$ iff (because e does not mention names from $\Gamma' - \Gamma$) $\models e\theta^\bullet$. But by construction, then, $c\theta^\bullet \in \Gamma$. (Basically, after the swap, the names mentioned in $c\theta^\bullet$ must be from $names(\Gamma)$.) In other words, for every atom p mentioned in $c\theta$, there is an atom $p^\bullet$ mentioned in $c\theta^\bullet$. To conclude, let $M'[p] = M[p^\bullet]$.

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

Next, we prove Lemma 11.

Proof: Suppose $\Delta \models \alpha$ but $\mathrm{GND}(\Delta, k) \wedge \neg\alpha$ is satisfiable. By Lemma 10, $\mathrm{GND}(\Delta, j) \wedge \neg\alpha$ is satisfiable for every $j \geq k$ and so, by Theorem 8, $\mathrm{GND}(\Delta) \wedge \neg\alpha$ is satisfiable, that is, $\Delta \wedge \neg\alpha$ is satisfiable. Contradiction.

Suppose $\mathrm{GND}(\Delta, k) \models \alpha$. Since $\mathrm{GND}(\Delta, k) \subseteq \mathrm{GND}(\Delta)$, $\mathrm{GND}(\Delta) \models \alpha$ and so $\Delta \models \alpha$. ∎

The proof for Theorem 9 is then as follows.

Proof: Let $\gamma = \{p \vee \neg p \mid \text{atom } p \text{ is mentioned in } \alpha\}$. Let $\Delta' = \Delta \wedge \gamma$. Since $\Delta' \equiv \Delta$, we have $\Delta \models \alpha$ iff $\Delta' \models \alpha$. From Lemma 11, $\Delta' \models \alpha$ iff $\mathrm{OU}(\Delta') \models \alpha$ iff $\mathrm{OU}(\Delta') \wedge \neg\alpha$ is unsatisfiable. It is easy to verify that $\mathrm{OU}(\Delta') \wedge \neg\alpha \equiv \mathrm{OU}(\Delta' \wedge \neg\alpha) \equiv \mathrm{OU}(\Delta \wedge \neg\alpha)$. Therefore, $\Delta \models \alpha$ iff $\mathrm{OU}(\Delta \wedge \neg\alpha)$ is unsatisfiable. ∎

Proof of Theorem 16

Proof: Here, (a) holds because if $\models \Delta \supset \alpha$ then $\Delta \equiv \Delta \wedge \alpha$. Next, (b) is by way of Theorem 9 because if $\Delta \wedge \alpha$ is unsatisfiable, then it has no models. Finally, suppose $\beta^\bullet$ means β with names a replaced by $a^\bullet$. For (c), by construction, $(\Delta \wedge \alpha)^\bullet = \Delta \wedge \alpha^\bullet$. By induction on α, we can show that for any model M of $\Delta \wedge \alpha$, we can construct a model M' of $\Delta \wedge \alpha^\bullet$, and vice versa. ∎

Proof of Proposition 21

Proof: Here, (i) is from the two transformation rules; (ii) is because S is obtained from a e of a $\forall$-clause $\forall(e \supset c) \in \Delta$ and so $S \subseteq B$ by construction; and (iii) is because S' is (also) obtained from a e of $\forall(e \supset c) \in \Delta$ and so $S' \subseteq B$ by construction. ∎

Proof of Theorem 22

Proof: The understanding is that $\mathrm{ENF}(\Delta)$ is wrt C. Let D be the set of names mentioned in $\mathrm{OU}(\Delta)$. By construction, $D \cap C = B$, where B is the set of names in Δ. Let $\bullet$ be a bijection from $\mathbb{N}$ to $\mathbb{N}$ that maps names from B to itself, and names from $D - C$ are mapped to those in $C - D$ (say, the smallest in $D - C$ is mapped to the smallest in $C - D$, and so on). An induction argument can be used to show that $\mathrm{WMC}(\mathrm{OU}(\Delta), w) = \mathrm{WMC}(\mathrm{OU}(\Delta)^\bullet, w)$. (Recall that predicate-level weight functions will not rank unknown atoms over other unknown instances, as already observed for Theorem 16.) But then $\mathrm{OU}(\Delta)^\bullet$ is equivalent to $\mathrm{ENF}(\Delta)$. ∎