DL-Lite Bayesian Networks: A Tractable Probabilistic Graphical Model

Denis D. Mauá and Fabio G. Cozman

Universidade de São Paulo,
São Paulo, Brazil
{denis.maua,fgcozman}@usp.br

Abstract. The construction of probabilistic models that can represent large systems requires the ability to describe repetitive and hierarchical structures. To do so, one can resort to constructs from description logics. In this paper we present a class of relational Bayesian networks based on the popular description logic DL-Lite. Our main result is that, for this modeling language, marginal inference and most probable explanation require polynomial effort. We show this by reductions to edge covering problems, and derive a result of independent interest; namely, that counting edge covers in a particular class of graphs requires polynomial effort.

1 Introduction

The search for an expressive *and* tractable formalism that can represent uncertainty *and* repetitive structures or hierarchical terminologies, is not an easy one. Most probabilistic models are propositional [14, 24], while combinations of logic and probabilities are typically quite flexible but intractable [3, 18]. However, there are proposals that try to balance expressivity and complexity by mixing logical constructs with graphs and independence relations [33, 17, 32]; for instance, *probabilistic relational models* [16] and *relational Bayesian networks* [21]. A few variants of these latter models even allow for polynomial time inferences by significantly restricting the syntax [15, 29].

In this paper we investigate the computational complexity of a modeling language that combines features of relational Bayesian networks with constructs of the popular description logic DL-Lite [7, 1]. In essence, we consider Bayesian networks which can be concisely specified using terminological assertions expressed in DL-Lite and marginal probability assertions on basic concepts and roles. For instance, we obtain a Bayesian network through the assertions

$$\mathsf{Employee} \equiv \mathsf{Person} \sqcap \exists \mathsf{salary}, \qquad \mathbb{P}(\mathsf{Person}) = 1/3, \qquad \mathbb{P}(\mathsf{salary}) = 1/4,$$

which encodes knowledge that an employee is defined as a person who receives a salary, an object is a person with probability $1/3$ and two objects are connected through the relation salary with probability $1/4$.

Our main contribution here is to show that marginal inferences and most probable explanations can be generated in polynomial time in our modeling language. So, we identify an island of tractability with non-trivial expressivity, offering a language that can be easily meshed with ontologies and relational schema.

The paper is organized as follows. We with some necessary background in Section 2. We then present DL-Lite Bayesian networks (Section 3) and their complexity with respect to marginal inferences and most probable explanations (Sections 4 and 5). Some of our results depend on a polynomial algorithm for counting edge covers, a result of independent interest that is briefly presented in Section 6. The connections with related work is discussed in Section 7. Section 8 comments on possible extensions and concludes the paper.

2 Bayesian networks, and DL-Lite

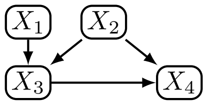
A Bayesian network consists of an acyclic directed graph whose nodes are random variables $X_1, \dots, X_n$, and a collection of conditional probability distributions, one distribution for each random variable given its parents. In this work, we consider only Boolean variables: we assume that each variable X_i takes on values 1 ("true") and 0 ("false"). The product of all conditional probability distributions determines a joint probability distribution over all variables, such that $\mathbb{P}(X_1 = x_1, \dots, X_n = x_n) = \prod_{i=1}^{n} \mathbb{P}(X_i = x_i | \mathrm{pa}(X_i) = \pi_i)$, where $\mathrm{pa}(X_i)$ denotes the parents of X_i and π_i is the projection of $\{x_1, \dots, x_n\}$ onto $\mathrm{pa}(X_i)$. A Bayesian network is *extensively specified* when its probability distributions are specified through tables of rational numbers.

A *marginal inference* is the computation of the probability of a number of assignments $\{X_i = x_i\}$ (*query*) given other assignments (*evidence*). This is a #P-complete problem [36], and NP-hard even to approximate [11].[1] Other common inference is *most probable explanation* (MPE), where one seeks an assignment to all variables that maximizes their joint probability given some evidence. Polynomial-time inference in extensively specified networks seems to require, under widely accepted assumptions about complexity classes, a bound on graph treewidth [26, 25], hence the interest in networks with restricted expressivity [9, 13, 15, 19, 31, 35].

To study the inferential complexity of various classes of Bayesian networks beyond the treewidth barrier, we have proposed a convenient framework in which to specify networks with binary variables [10]. In this framework, a directed acyclic graph is given where each node is a random variable; each root variable X is associated with a marginal probability $\mathbb{P}(X = 1) = \alpha$, and each non-root variable Y is associated with a formula $Y \Leftrightarrow \phi$, where ϕ is a well-formed formula on the parents of Y; the latter is equivalent to specifying that $\mathbb{P}(Y = 1|\phi) = 1$

[1] Recall that #P is the class of integer-valued functions computed by counting Turing machines in polynomial time; a counting Turing machine is a standard nondeterministic Turing machine that prints in binary notation, on a separate tape, the number of accepting computations induced by the input.

if ϕ is true and zero otherwise. By restricting the language from which ϕ can be selected, one obtains a class of Bayesian networks (a language is simply a set of well-formed formulas). For instance, if the language consists of all propositional sentences, we can represent any joint probability distribution (perhaps introducing fresh variables). Or we may employ a sub-Boolean language with only conjunction and disjunction, as in the next network:



$$\mathbb{P}(X_1) = 1/4, \qquad \mathbb{P}(X_2) = 1/2,$$
$$X_3 \Leftrightarrow X_1 \wedge X_2, \qquad X_4 \Leftrightarrow X_2 \vee X_3.$$

A *relational Bayesian network* consists of a directed acyclic graph whose nodes are relations $\mathsf{r}_1, \ldots, \mathsf{r}_n$ [21, 22], plus a set of real-valued functions soon to be explained. To interpret a relational Bayesian network, first take a set of *individuals* $\mathcal{D}$, called a *domain*. A *grounding* of k-ary relation r is denoted by $\mathsf{r}(\mathsf{a}_1, \ldots, \mathsf{a}_k)$, where $\mathsf{a}_1, \ldots, \mathsf{a}_k \in \mathcal{D}$. Given a relational Bayesian network and a domain, one can build a directed acyclic graph where each possible grounding is a node, and where an arc is added between two groundings if there is an arc between their corresponding relations in the network. The real-valued functions we have mentioned specify the probability of each grounding given its parents' grounding. In Jaeger's original proposal [21], these real-valued functions are restricted to a few basic forms. Here we focus on the restricted syntax proposed in [10]: for each root relation r we have an assessment $\mathbb{P}(\mathsf{r}) = \alpha$, where α is a rational in $[0, 1]$. And for each non-root k-ary relation s we have an equivalence $\mathsf{s}(\mathfrak{x}_1, \ldots, \mathfrak{x}_k) \Leftrightarrow \phi(\mathfrak{x}_1, \ldots, \mathfrak{x}_k)$, where each $\mathfrak{x}_i$ denotes a logical variable and ϕ is a well-formed formula in a first-order language. Our strategy in this paper is to restrict ϕ to constructs from the DL-Lite description logic.

DL-Lite is particularly interesting because it captures a great deal of features found in conceptual modeling by ER or UML diagrams, and yet common inference services have polynomial complexity [7]. A whole family of variants of DL-Lite has been developed [1], and in fact this family is the basis of one of the OWL QL profile (http://www.w3.org/TR/owl2-profiles/). To recap, DL-Lite is a description logic that deals with concepts, roles, and individuals; we treat those as unary relations, binary relations, and constants. Some of the concepts are marked as *primitive* ones. Given a primitive concept s, both s and $\neg\mathsf{s}$ are formulas, to be interpreted respectively as $\mathsf{s}(\mathfrak{x})$ and $\neg\mathsf{s}(\mathfrak{x})$. Given a role r, both $\exists\mathsf{r}$ and $\exists\mathsf{r}^-$ are formulas, to be interpreted respectively as $\exists\mathfrak{y} : \mathsf{r}(\mathfrak{x}, \mathfrak{y})$ and $\exists\mathfrak{y} : \mathsf{r}(\mathfrak{y}, \mathfrak{x})$ (r^- is an *inverse role*). Also, if ϕ and φ are formulas, then $\phi \sqcap \varphi$ is a formula (interpreted as $\phi \wedge \varphi$). Finally, a *concept definition* $\mathsf{s} \equiv \phi$ is interpreted as $\forall\mathfrak{x} : \mathsf{s}(\mathfrak{x}) \Leftrightarrow \phi(\mathfrak{x})$. Note that any formula ϕ can have only one free logical variable. A primitive concept cannot appear in the left-hand side of a concept definition (indeed this characterizes primitive concepts). Inverse roles are defined as $\forall\mathfrak{x}, \mathfrak{y} : \mathsf{r}^-(\mathfrak{y}, \mathfrak{x}) \Leftrightarrow \mathsf{r}(\mathfrak{x}, \mathfrak{y})$. The semantics of DL-Lite uses a domain $\mathcal{D}$ and an interpretation $\mathbb{I}$ that maps each individual to an element of $\mathcal{D}$, each concept to a subset of $\mathcal{D}$, and each role to a set of pairs in $\mathcal{D} \times \mathcal{D}$. The semantics of a formula in essence reads the formula as a first-order formula and uses $\mathcal{D}$ and $\mathbb{I}$ in the usual semantics of first-order logic [7].

Example 1. The following concept definitions express simple facts about families: first, $\mathsf{female} \equiv \neg\mathsf{male}$; also, $\mathsf{father} \equiv \mathsf{male} \sqcap \exists\mathsf{parentOf}$, $\mathsf{mother} \equiv \mathsf{female} \sqcap \exists\mathsf{parentOf}$, $\mathsf{son} \equiv \mathsf{male} \sqcap \exists\mathsf{parentOf}^-$, $\mathsf{daughter} \equiv \mathsf{female} \sqcap \exists\mathsf{parentOf}^-$.

3 DL-Lite Bayesian networks

We now consider the class of relational Bayesian networks over binary variables where each conditional probability is specified through a DL-Lite formula. A *DL-Lite Bayesian network* is a relational Bayesian network that consists of a directed acyclic graph where each node is a unary or binary relation, and such that

- each root relation r is associated with an assessment $\mathbb{P}(\mathsf{r}) = \alpha$, for a rational $\alpha \in [0, 1]$, and
- each non-root relation r is either a unary relation associated with a concept definition $\mathsf{r} \equiv \phi$, where ϕ is a formula in DL-Lite only mentioning parent relations, or an inverse role s^- with s as its single parent.

Example 2. The graph and assessments in Figure 1, plus the concept definitions in Example 1, specify a DL-Lite Bayesian network.

The semantics of DL-Lite Bayesian networks is given by a simple combination of semantics for relational Bayesian networks and for DL-Lite. That is, consider a domain $\mathcal{D}$ containing individuals; in this paper we assume every domain to be finite and given as a list of elements. We also assume, as most first-order probabilistic logics do, that interpretations are *rigid* [3] in that an element corresponds to the same individual in every possible interpretation of relations. For each concept s and individual a, produce the grounding $\mathsf{s}(\mathsf{a})$; likewise, for each role r and each pair of individuals (a, b), produce the grounding $\mathsf{r}(\mathsf{a}, \mathsf{b})$. A set with all possible interpretations is obtained by considering all possible truth assignments for these groundings. We can associate each grounding with a random variable that takes each possible interpretation either to 1 (the grounding is true in that interpretation) or to 0 (otherwise). To simplify the notation, we use the same symbol for a grounding and its associated random variable. Now construct a grounded graph. First, each grounding is a node. Second, take each concept definition $\mathsf{s} \equiv \phi$; for each grounding $\mathsf{s}(\mathsf{a})$, specify as its parents the groundings

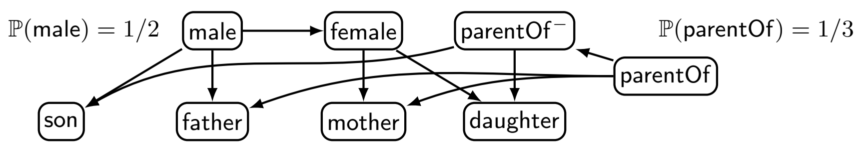


Fig. 1. A DL-Lite Bayesian network.

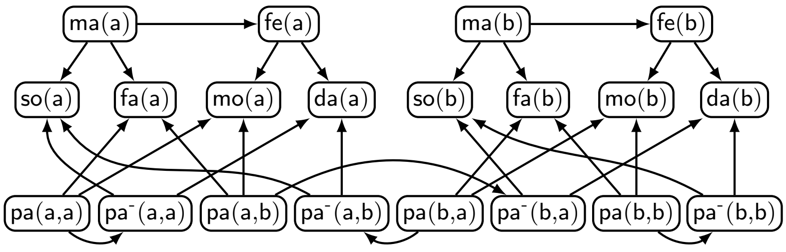


Fig. 2. Grounded network, domain $\mathcal{D} = \{\mathsf{a}, \mathsf{b}\}$.

that appear in $\phi(\mathsf{a})$. Third, associate with each root grounding the corresponding grounded probabilistic assessment. For instance, suppose we have assessments $\mathbb{P}(\mathsf{s}) = \alpha'$ and $\mathbb{P}(\mathsf{r}) = \alpha''$ for concept s and role r; then, attach to grounding $\mathsf{s}(\mathsf{a})$ the assessment $\mathbb{P}(\mathsf{s}(\mathsf{a})) = \alpha'$, and similarly, attach to grounding $\mathsf{r}(\mathsf{a}, \mathsf{b})$ the assessment $\mathbb{P}(\mathsf{r}(\mathsf{a}, \mathsf{b})) = \alpha''$.

Example 3. Take $\mathcal{D} = \{a, b\}$. Then the relational Bayesian network in Example 2 induces the Bayesian network in Figure 2 (where names of relations have been shortened, e.g. parentOf has become pa).

DL-Lite Bayesian networks can be argued for in two ways. First, they offer an intuitive and disciplined language in which to express relational Bayesian networks; in essence, DL-Lite is used to reduce the complexity of Jaeger's combination functions [21, 22]. Second, they offer a simple way to create probabilistic acyclic ontologies; this is particularly valuable as acyclic ontologies are common in practice [2].

To these inviting features we add a third, most important one: useful inferences in DL-Lite Bayesian networks require polynomial effort. The inferences of interest are as follows. Suppose we have a DL-Lite Bayesian network $\mathbb{B}$ and a domain $\mathcal{D}$ (as a list of individuals). To compute a marginal inference $\mathbb{P}(\mathbf{Q}|\mathbf{E})$ for sets of assignments $\mathbf{Q}$ and $\mathbf{E}$, we calculate $\mathbb{P}(\mathbf{Q} \wedge \mathbf{E})/\mathbb{P}(\mathbf{E})$. So, our central *inference* problem is to compute $\mathbb{P}(\mathbf{E})$ for a given set of assignments (the evidence). That is, inference in DL-Lite Bayesian networks boils down to computing marginals, as usual in Bayesian networks.

For reasons to be clear, we say that evidence is *positive* when all assignments attach value 1 (true). For instance, $\{\mathsf{male}(\mathsf{John}) = 1, \mathsf{female}(\mathsf{Mary}) = 1\}$ is positive evidence. Similarly, $\{\mathsf{father}(\mathsf{John}) = 0, \mathsf{mother}(\mathsf{Mary}) = 0\}$ is *negative* evidence.

Another problem is to find a *most probable explanation (MPE)*; that is, to find an interpretation for *all* groundings of the DL-Lite Bayesian network $\mathbb{B}$ with respect to domain $\mathcal{D}$, that maximizes the probability and is consistent with a given set of assignments (evidence) $\mathbf{E}$.

As a digression, note that we argue in Section 8 that results in the next section can be adapted to produce fully-polynomial time approximations in a larger set of languages that can be directly useful in conceptual modeling.

4 Marginal inferences

Our main result is that (marginal and MPE) inference in a DL-Lite Bayesian network requires polynomial effort as long as evidence is positive. One consequence of this result is that DL-Lite Bayesian networks are dqe-/domain-liftable for positive evidence [4, 23]. Another point is that our results offer explicit algorithms for a class of two-variable logics [5].

The focus on positive assignments is justified, as conjuntion (a subset of DL-Lite) leads to #P-hardness with arbitrary evidence [10]. Our first result is:

Theorem 1. *With a DL-Lite Bayesian network, a domain, and positive evidence as input, inference is polynomial-time computable in the size of the input.*

We prove this theorem by a quadratic-time reduction to multiple independent problems of counting weighted edge covers with uniform weights in a very particular class of graphs. Then we show (in Section 6) that the latter problem can be solved in quadratic time (hence the total time is quadratic).

We first transform the relational network into an equal-probability model. Collapse each role r and its inverse r^- into a single node r. For each (collapsed) role r, insert variables $\mathsf{e_r} \equiv \exists\mathsf{r}$ and $\mathsf{e_r^-} \equiv \exists\mathsf{r}^-$; replace each appearance of the formula $\exists\mathsf{r}$ by the variable $\mathsf{e_r}$, and each appearance of $\exists\mathsf{r}^-$ by $\mathsf{e_r^-}$. This transformation does not change the probability of $\mathbf{E}$, and it allows us to easily refer to groundings of formulas $\exists\mathsf{r}$ and $\exists\mathsf{r}^-$ as groundings of $\mathsf{e_r}$ and $\mathsf{e_r^-}$, respectively.

Observe that only the nodes with assignments in $\mathbf{E}$ and their ancestors are relevant for the computation of $\mathbb{P}(\mathbf{E})$, as every other node in the Bayesian network is barren [14]. Hence, we can assume without loss of generality that $\mathbf{E}$ contains only leaves of the network. If $\mathbf{E}$ contains only root nodes, then $\mathbb{P}(\mathbf{E})$ can be computed trivially as the product of marginal probabilities which are readily available from the specification. Thus assume that $\mathbf{E}$ assigns a positive value to at least one leaf grounding $\mathsf{s(a)}$, where a is some individual in the domain. Then by construction $\mathsf{s(a)}$ is associated with a logical sentence $X_1 \wedge \cdots \wedge X_k$, where each X_i is either a grounding of non-primitive unary relation in individual a, a grounding of a primitive unary relation in a, or the negation of a grounding of a primitive unary relation in a. It follows that $\mathbb{P}(\mathbf{E}) = \mathbb{P}(\mathsf{s(a)} = 1 | X_1 = 1, \ldots, X_k = 1)\mathbb{P}(\mathbf{E}') = \mathbb{P}(\mathbf{E}')$, where $\mathbf{E}'$ is $\mathbf{E}$ after removing the assignment $\mathsf{s(a)} = 1$ and adding the assignments $\{X_1 = 1, \ldots, X_k = 1\}$. Now it might be that $\mathbf{E}'$ contains both the assignments $\{X_i = 1\}$ and $\{X_i = 0\}$. Then $\mathbb{P}(\mathbf{E}) = 0$ (this can be verified efficiently). So assume there are no such inconsistencies. The problem of computing $\mathbb{P}(\mathbf{E})$ boils down to computing $\mathbb{P}(\mathbf{E}')$; in the latter problem the node $\mathsf{s(a)}$ is discarded for being barren. Moreover, we can replace any assignment $\{\neg\mathsf{r(a)} = 1\}$ in $\mathbf{E}'$ for some primitive concept r with the equivalent assignment $\{\mathsf{r(a)} = 0\}$. By repeating this procedure for all internal nodes which are not groundings of $\mathsf{e_r}$ or $\mathsf{e_r^-}$, we end up with a set $\mathbf{A}$ containing positive assignments of groundings of roles and of concepts $\mathsf{e_r}$ and $\mathsf{e_r^-}$, and (not necessarily positive) assignments of groundings of primitive concepts. Each grounding of a primitive concept or role is (a root node hence) marginally independent from all other groundings in $\mathbf{A}$; hence

$\mathbb{P}(\mathbf{A}) = \mathbb{P}(\mathbf{B}|\mathbf{C}) \prod_i \mathbb{P}(A_i)$, where each A_i is an assignment to a root node, $\mathbf{B}$ are (positive) assignments to groundings of concepts $\mathsf{e_r}$ and $\mathsf{e_r^-}$ for relations r, and $\mathbf{C} \subseteq \{A_1, A_2, \dots\}$ are groundings of roles (if $\mathbf{C}$ is empty then assume it expresses a tautology). Since the marginal probabilities $\mathbb{P}(A_i)$ are available from the specification the joint $\prod_i \mathbb{P}(A_i)$ can be computed in linear time in the input. We thus focus on computing $\mathbb{P}(\mathbf{B}|\mathbf{C})$ as defined (if $\mathbf{B}$ is empty, we are done). To recap, $\mathbf{B}$ is a set of assignments $\mathsf{e_r(a)} = 1$ and $\mathsf{e_r^-(b)} = 1$ and $\mathbf{C}$ is a set of assignments $\mathsf{r(c, d)} = 1$ for arbitrary roles r and individuals $\mathsf{a, b, c}$ and d.

For a role r, let $\mathcal{D}_\mathsf{r}$ be the set of individuals $\mathsf{a} \in \mathcal{D}$ such that $\mathsf{e_r(a)} = 1$ is in $\mathbf{B}$, and let $\mathcal{D}_\mathsf{r}^-$ be the set of individuals $\mathsf{a} \in \mathcal{D}$ such that $\mathbf{B}$ contains $\mathsf{e_r^-(a)} = 1$. Let $\mathsf{gr(r)}$ be the set of all groundings of relation r, and let $\mathsf{r}_1, \dots, \mathsf{r}_k$ be the roles in the (relational) network. By the factorization property of Bayesian networks it follows that

$$\mathbb{P}(\mathbf{B}|\mathbf{C}) = \sum_{\mathsf{gr(r_1)}} \cdots \sum_{\mathsf{gr}(\mathsf{r}_k)} \prod_{i=1}^{k} \prod_{\mathsf{a} \in \mathcal{D}_{\mathsf{r}_i}} \mathbb{P}(\mathsf{e}_{\mathsf{r}_i}(\mathsf{a}) = 1 | \mathrm{pa}(\mathsf{e}_{\mathsf{r}_i}(\mathsf{a})), \mathbf{C}) \times \prod_{\mathsf{a} \in \mathcal{D}_{\mathsf{r}_i}^-} \mathbb{P}(\mathsf{e}_{\mathsf{r}_i}^-(\mathsf{a}) = 1 | \mathrm{pa}(\mathsf{e}_{\mathsf{r}_i}^-(\mathsf{a})), \mathbf{C}) \mathbb{P}(\mathsf{gr}(\mathsf{r}_k) | \mathbf{C}) ,$$

which by distributing the products over sums is equal to

$$\prod_{i=1}^{k} \sum_{\mathsf{gr}(\mathsf{r}_i)} \prod_{\mathsf{a} \in \mathcal{D}_\mathsf{r}} \mathbb{P}(\mathsf{e_r(a)} = 1 | \mathrm{pa}(\mathsf{e_r(a)}), \mathbf{C}) \prod_{\mathsf{a} \in \mathcal{D}_\mathsf{r}^-} \mathbb{P}(\mathsf{e_r^-(a)} = 1 | \mathrm{pa}(\mathsf{e_r^-(a)}), \mathbf{C}) \mathbb{P}(\mathsf{gr}(\mathsf{r}_k) | \mathbf{C}) .$$

Consider an assignment $\mathsf{r(a, b)} = 1$ in $\mathbf{C}$. By construction, the children of the grounding $\mathsf{r(a, b)}$ are $\mathsf{e_r(a)}$ and $\mathsf{e_r^-(b)}$. Moreover, the assignment $\mathsf{r(a, b)} = 1$ implies that $\mathbb{P}(\mathsf{e_r(a)} = 1 | \mathrm{pa}(\mathsf{e_r(a)}), \mathbf{C}) = 1$ (for any assignment to the other parents) and $\mathbb{P}(\mathsf{e_r^-(b)} = 1 | \mathrm{pa}(\mathsf{e_r(a)}), \mathbf{C}) = 1$ (for any assignment to the other parents). This is equivalent in the factorization above to removing $\mathsf{r(a, b)}$ from $\mathbf{C}$ (as it is independent of all other groundings), and removing individuals a from $\mathcal{D}_\mathsf{r}$ and b from $\mathcal{D}_\mathsf{r}^-$. So repeat this procedure for every grounding in $\mathbf{C}$ until this set is empty (this can be done in polynomial time). The inference problem becomes one of computing $\gamma(\mathsf{r}) = \sum_{\mathsf{gr}(\mathsf{r}_i)} \prod_{\mathsf{a} \in \mathcal{D}_\mathsf{r}} \mathbb{P}(\mathsf{e_r(a)} = 1 | \mathrm{pa}(\mathsf{e_r(a)})) \prod_{\mathsf{a} \in \mathcal{D}_\mathsf{r}^-} \mathbb{P}(\mathsf{e_r^-(a)} = 1 | \mathrm{pa}(\mathsf{e_r^-(a)})) \mathbb{P}(\mathsf{gr}(\mathsf{r}_k))$ for every relation r_i, $i = 1, \dots, k$. We will show that this problem can be reduced to a tractable instance of counting weighted edge covers.

To this end, consider the graph G whose node set V can be partitioned into sets $V_1 = \{\mathsf{e_r^-(a)} : \mathsf{a} \in \mathcal{D} \setminus \mathcal{D}_\mathsf{r}^-\}$, $V_2 = \{\mathsf{e_r(a)} : \mathsf{a} \in \mathcal{D}_\mathsf{r}\}$, $V_3 = \{\mathsf{e_r^-(a)} : \mathsf{a} \in \mathcal{D}_\mathsf{r}^-\}$, $V_4 = \{\mathsf{e_r(a)} : \mathsf{a} \in \mathcal{D} \setminus \mathcal{D}_\mathsf{r}\}$, and for $i = 1, 2, 3$ the graph obtained by considering nodes $V_i \cup V_{i+1}$ is bipartite complete. An edge with endpoints $\mathsf{e_r(a)}$ and $\mathsf{e_r^-(b)}$ represents the grounding $\mathsf{r(a, b)}$; we identify every edge with its corresponding grounding. We call this graph the *intersection graph* of $\mathbf{B}$ with respect to r and $\mathcal{D}$. The parents of a node in the graph correspond exactly to the parents of the node in the Bayesian network. For example, the graph in Figure 3 represents the assignments $\mathbf{B} = \{\mathsf{e_r(a)} = 1, \mathsf{e_r(b)} = 1, \mathsf{e_r(d)} = 1, \mathsf{e_r^-(b)} = 1, \mathsf{e_r^-(c)} = 1\}$, with respect to domain $\mathcal{D} = \{\mathsf{a, b, c, d, e}\}$. The black nodes (resp., white nodes)

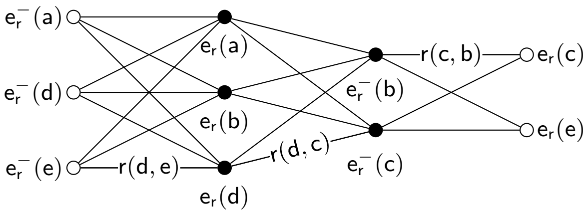


Fig. 3. Representing assignments by graphs.

represent groundings in (resp., not in) **B**. For clarity's sake, we label only a few edges.

Before showing the equivalence between the inference problem and counting edges covers, we need to introduce some graph-theoretic notions and notation. Consider a (simple, undirected) graph $G = (V, E)$. Denote by $E_G(u)$ the set of edges incident on a node $u \in V$, and by $N_G(u)$ the open neighborhood of u. For $U \subseteq V$, we say that $C \subseteq E$ is a U-cover if for each node $u \in U$ there is an edge $e \in C$ incident in u (i.e., $e \in E_G(u)$). For any fixed real λ, we say that $\lambda^{|C|}$ is the weight of cover C. The *partition function* of G is $Z(G, U, \lambda) = \sum_{C \in EC(G,U)} \lambda^{|C|}$, where $U \subseteq V$, $EC(G, U)$ is the set of U-covers of G and λ is a positive real. If $\lambda = 1$ and $U = V$, the partition function is the number of edge covers. The following result connects counting edge covers to marginal inference in DL-Lite Bayesian networks.

Proposition 1. *Let $G = (V_1, V_2, V_3, V_4, E)$ be the intersection graph of* **B** *with respect to a relation* r *and domain $\mathcal{D}$. Then $\gamma(\mathsf{r}) = Z(G, V_2 \cup V_3, \alpha/(1-\alpha))/(1-\alpha)^{|E|}$, where $\alpha = \mathbb{P}(\mathsf{r}(\mathtt{x}, \mathtt{y}))$.*

Proof. Let $B = V_2 \cup V_3$, and consider a B-cover C. The assignment that sets to true all groundings $\mathsf{r}(\mathsf{a}, \mathsf{b})$ corresponding to edges in C, and sets to false the remaining groundings of r makes $\mathbb{P}(\mathsf{e_r}(\mathsf{a}) = 1|\mathrm{pa}(\mathsf{e_r}(\mathsf{a}))) = \mathbb{P}(\mathsf{e_r^-}(\mathsf{b}) = 1|\mathrm{pa}(\mathsf{e_r^-}(\mathsf{b}))) = 1$ for every $\mathsf{a} \in \mathcal{D}_\mathsf{r}$ and $\mathsf{b} \in \mathcal{D}_\mathsf{r}^-$; it makes $\mathbb{P}(\mathsf{gr}(\mathsf{r})) = \mathbb{P}(\mathsf{r})^{|C|}(1 - \mathbb{P}(\mathsf{r}))^{|E|-|C|} = (1-\alpha)^{|E|}\alpha^{|C|}/(1-\alpha)^{|C|}$, which is the weight of the cover C scaled by $(1-\alpha)^{|E|}$. Now consider a set of edges C which is not a B-cover and obtains an assignment to groundings $\mathsf{gr}(\mathsf{r})$ as before. There is at least one node in B that does not contain any incident edges in C. Assume that node is $\mathsf{e}(\mathsf{a})$; then all parents of $\mathsf{e}(\mathsf{a})$ are assigned false, which implies that $\mathbb{P}(\mathsf{e_r}(\mathsf{a}) = 1|\mathrm{pa}(\mathsf{e_r}(\mathsf{a}))) = 0$. The same is true if the node not covered is a grounding $\mathsf{e}^-(\mathsf{a})$. Hence, for each B-cover C the probability of the corresponding assignment equals its weight up to the factor $(1-\alpha)^{|E|}$. And for each edge set C which is not a B-cover its corresponding assignment has probability zero. □

We have thus established that, if a particular class of edge cover counting problems is polynomial, then marginal inference in DL-Lite Bayesian networks is also polynomial. Since the former is shown to be true in Section 6, this concludes the proof of Theorem 1.

5 Most probable explanations

Using previous techniques, we can also show the following result:

Theorem 2. *With a DL-Lite Bayesian network, a domain, and positive evidence as input, finding a most probable explanation is polynomial-time computable in the size of the input.*

In this theorem we are interested in finding an assignment $\mathbf{X}$ to all groundings that maximizes $\mathbb{P}(\mathbf{X} \wedge \mathbf{E})$, where $\mathbf{E}$ is a set of positive assignments. Perform the substitution of formulas $\exists \mathsf{r}$ and $\exists \mathsf{r}^-$ by logically equivalent concepts $\mathsf{e_r}$ and $\mathsf{e_r^-}$ as before. Consider a non-root grounding $\mathsf{s(a)}$ in $\mathbf{E}$ which is not the grounding of $\mathsf{e_r}$ or $\mathsf{e_r^-}$; by construction, $\mathsf{s(a)}$ is logically equivalent to a conjunction $X_1 \wedge \cdots \wedge X_k$, where $X_1, \ldots, X_k$ are unary groundings. Because $\mathsf{s(a)}$ is assigned to true, any assignment $\mathbf{X}$ with nonzero probability assigns $X_1, \ldots, X_k$ to true. Moreover, since $\mathsf{s(a)}$ is an internal node, its corresponding probability is one. Hence, if we include all the assignments $X_i = 1$ to its parents in $\mathbf{E}$, the MPE value does not change. As in the computation of inference, we might generate an inconsistency when setting the values of parents; in this case halt and return zero (and an arbitrary assignment). So assume we repeated this procedure until $\mathbf{E}$ contains all ancestors of the original groundings which are groundings of unary relations, and that no inconsistency was found. Note that at this point we only need to assign values to nodes which are either not ancestors of any node in the original set $\mathbf{E}$, and to groundings of (collapsed) roles r.

Consider the groundings of primitive concepts r which are not ancestors of any grounding in $\mathbf{E}$. Setting its value to maximize its marginal probability does not introduce any inconsistency with respect to $\mathbf{E}$. Moreover, for any assignment to these groundings, we can find a consistent assignment to the remaining groundings (which are internal nodes and not ancestors of $\mathbf{E}$), that is, an assignment which assigns positive probability. Since this is the maximum probability we can obtain for these groundings, this is a partial optimum assignment.

We are thus only left with the problem of assigning values to the groundings of relations r which are ancestors of $\mathbf{E}$. Consider a relation r such that $\mathbb{P}(\mathsf{r}) \geq 1/2$. Then assigning all groundings of r to true maximizes their marginal probability and satisfies the logical equivalences of all groundings in $\mathbf{E}$. Hence, this is a maximum assignment (and its value can be computed efficiently). So assume there is a relation r with $\mathbb{P}(\mathsf{r}) < 1/2$ such that a grounding of $\mathsf{e_r}$ or $\mathsf{e_r^-}$ appear in $\mathbf{E}$. In this case, the greedy assignment sets every grounding of r; however, such an assignment is inconsistent with the logical equivalence of $\mathsf{e_r}$ and $\mathsf{e_r^-}$, hence obtains probability zero. Now consider an assignment that assigns exactly one grounding $\mathsf{r(a, b)}$ to true and all the other to false. This assignment is consistent with $\mathsf{e_r(a)}$ and $\mathsf{e_r}(\mathbf{b})$, and maximizes the probability; any assignment that sets more groundings to true has a lower probability since it replaces a term $1 - \mathbb{P}(\mathsf{r}) \geq 1/2$ with a term $\mathbb{P}(\mathsf{r}) < 1/2$ in the joint probability. More generally, to maximize the joint probability we need to assign to true as few groundings $\mathsf{r(a, b)}$ which are ancestors of $\mathbf{E}$ as possible. This is equivalent to a minimum cardinality edge covering problem as follows.

For every relation r in the relational network, construct the bipartite complete graph $G_\mathsf{r} = (V_1, V_2, E)$ such that V_1 is the set of groundings $\mathsf{e_r(a)}$ that appears and have no parent $\mathsf{r}(\mathbf{a}, \mathsf{b})$ in $\mathbf{E}$, and V_2 is the set of groundings $\mathsf{e_r^-(a)}$ that appears and have no parents in $\mathbf{E}$. We identify an edge connecting $\mathsf{e_r(a)}$ and $\mathsf{e_r^-(b)}$ with the grounding $\mathsf{r(a, b)}$. For any set $C \subseteq E$, construct an assignment by attaching true to the groundings $\mathsf{r(a, b)}$ in C and false to every other grounding $\mathsf{r(a, b)}$. This assignment is consistent with $\mathbf{E}$ if and only if C is an edge cover; hence the minimum cardinality edge cover maximizes the joint probability (it is consistent with $\mathbf{E}$ and attaches true to the least number of groundings of $\mathbf{r}$). This concludes the proof of Theorem 2.

6 Counting edge covers

In this section we discuss the fact that, for graphs such as those representing formulas in DL-Lite, the partition function can be computed in polynomial time. Specifically, we consider graphs $G = (V, E)$ whose nodes can be partitioned into four disjoint sets V_1, V_2, V_3, V_4 such that the subgraph obtained by considering only edges V_i and V_{i+1} is complete bipartite ($i = 1, 2, 3$). We call such graphs stepwise bipartite complete. For lack of space, we only present the main ideas; details and proofs can be found elsewhere [30].

We partition the nodes into *white* nodes $W = V_1 \cup V_4$ and *black* nodes $B = V_2 \cup V_3$. As we will be interested only in B-covers, we will refer to them simply as covers. An edge $e = (u, v)$ is classified into one of three categories with respect to the partition W, B: it is a *free edge* if $u, v \in W$; a *dangling edge* if $u \in W, v \in B$, or a *regular edge* if $u, v \in B$. For convenience, we fix λ and write $Z(G)$ to denote $Z(G, \lambda)$. Computing $Z(G)$ for general graphs is $\#\mathsf{P}$-complete even for $\lambda = 1$ [6], and admits a FPTAS for bounded λ [27, 28]. We will show that for stepwise bipartite complete graphs, the problem can be solved in polynomial time.

Let e be an edge and u be a vertex in $G = (W, B, E)$. We define the following operations and notation: $G - e = (W, B, E \setminus \{e\})$ and $G - u = (W \cup \{u\}, B \setminus \{u\}, E)$. These operations do not change the vertex set (only the partition), and are associative (e.g., $G - e - f = G - f - e$, $G - u - v = G - v - u$, and $G - e - u = G - u - e$). Hence, if $E = \{e_1, \cdots, e_d\}$ is a set of edges, we can write $G - E$ to denote $G - e_1 - \cdots - e_d$ applied in any arbitrary order. The same is true for any combination of these operations.

The following results, easily derived from the work of Lin, Liu and Lu [27], show that the partition function can be computed recursively on smaller graphs and solved efficiently when no black nodes exist:

Proposition 2. *Let $e = (u, v)$ be an edge. 1) If e is dangling edge with u colored black then $Z(G) = (1 + \lambda)Z(G - e - u) - Z(G - E_G(u) - u)$. 2) If e is a free edge of G then $Z(G) = (1 + \lambda)Z(G - e)$. 3) If u is an isolated white node (i.e., $N_G(u) = \emptyset$) then $Z(G) = Z(G - u)$.*

The result above allows us to decompose the problem of computing $Z(G)$ into two smaller problems until the the problems are simple enough to be solved by

```
procedure PF(G, w, k1, k2):
if (w ∈ V1 and k1 + k2 < n) or (w ∈ V4 and k1 + k2 < m):
    if w ∈ V1 do v ← u_{k1+k2} else do v ← v_{k1+k2};
    return (1+λ) ∗ PF(G−(v, w)−v, w, k1, k2+1) − PF(G−E_G(v)−v, w, k1+1, k2);
else:
    remove free edges and isolate nodes; let k be the number of edges removed;
    do w ← w ∈ V1; V1 ← V1 ∪ V3; V3 ← ∅;
    return (1 + λ)^k ∗ PF(G, w, 0, 0);
```

Fig. 4. Algorithm PF. Takes a graph $G = (V_1, V_2, V_3, V_4, E)$ with $V_2 = \{u_1, \ldots, u_n\}$ and $V_3 = \{v_1, \ldots, v_m\}$, a node $w \in V_1 \cup V_4$, and nonnegative integers k_1 and k_2.

a simple count (of free edges). That approach however generates an exponential number of recursions. A polynomial-time algorithm can be obtained by exploiting the symmetries in the graphs, obtained through graph isomorphism.

Two graphs are isomorphic if there is an edge-preserving bijection between the nodes of the two graphs that also preserves their color. Two isomorphic graphs have the same value of the partition function. The next result shows that the order in which operations of edge removal and and node whitening are performed among isomorphic nodes does not affect the value of $Z(G)$:

Proposition 3. *Let $u_1, \ldots, u_n$ be the nodes in V_2 (V_3) and w be a node in V_1 (V_4). Given any permutation σ on V_2 (V_3) and nonnegative integers $k_1 + k_2 \leq n$ the graphs $G' = G - E_G(u_1) - \cdots - E_G(u_{k_1}) - (w, u_{k_1+1}) - \cdots - (w, u_{k_1+k_2}) - u_1 - \cdots - u_{k_1+k_2}$ and $G'' = G - E_G(\sigma(u_1)) - \cdots - E_G(\sigma(u_{k_1})) - (w, \sigma(u_{k_1+1})) - \cdots - (w, \sigma(u_{k_1+k_2})) - \sigma(u_1) - \cdots - \sigma(u_{k_1+k_2})$ are isomorphic.*

Using these facts, Algorithm PF (Figure 4) produces $Z(G)$.

Theorem 3. *Let G be a stepwise bipartite complete graph. Then Algorithm PF with an arbitrary node w in V_4 and $k_1 = k_2 = 0$ outputs $Z(G)$ in time and memory polynomial in the number of nodes and edges of G if the calls are cached (so that no two calls with same arguments are performed).*

The algorithm PF requires the existence of dangling edges. Now it might be that the graph contains no white nodes (hence no dangling edges), that is, that G is complete bipartite graph. The next result shows how to decompose the problem of computing the partition function into problems of computing the partition function in smaller graphs.

Proposition 4. *Let G be a bipartite complete graph and $e = (u, v)$ be some edge. Then $Z(G) = (1+\lambda)Z(G-e-u-v) - Z(G-E_G(v)-v) - Z(G-E_G(u)-u) - Z(G-E_G(u)-E_G(v)-u-v)$. The graphs in the right-hand side are either bipartite complete or stepwise bipartite complete with a dangling edge.*

7 Related work

In previous work [10] we have shown that marginal inference can be computed in polynomial-time in models described by a restricted version of the language considered here, one that does not admit inverse roles. The absence of inverse roles leads to Bayesian networks composed of disconnected components, where each component contains all concepts related to an individual; the complexity of inferences with "positive" evidence is then easily seem to be polynomial by applying d-separation. The use of inverse roles connects components related to different individuals, so the same argument cannot be used.

There have been many attempts at combining description logics with probabilities. Heinsohn [20] was one of the first to propose modeling languages that allow uncertainty into terminological descriptions. Much of the work in probabilistic description logics is however hindered by intractability of inferences. The DL-Lite language was conceived as a lightweight knowledge representation scheme to represent large bases of relational data with very efficient reasoning services. The simplicity and computational efficiency of the DL-Lite language have led many researchers to use it as a building block of modeling languages that combine description logics and Bayesian networks. For instance, D'Amato et. al [12] propose a variant of DL-Lite where the interpretation of each sentence is conditional on a context. The context is specified by a Bayesian network, and is hence probabilistic. The probability of concepts can then be extended to determine the probability of logical expressions. A similar approach was takes by Ceylan and Peñalosa in their Bayesian Description Logic [8], with minor semantic differences. A different approach is to extend the syntax of DL-Lite sentences with probabilistic subsumption connectives, as in the Probabilistic DL-Lite [34]. Differently from our proposal here, none of those works use DL-Lite to specify (large) Bayesian networks.

8 Extensions, and conclusion

The previous results can be directly extended in some important ways. For example, if we allow negative groundings of roles in the evidence, then most of the proof of Theorem 1 follows; the difference is that the intersection graphs obtained do not satisfy the same symmetries. We can then resort to approximations for weighted edge cover counting [28], so as to develop a fully polynomial-time approximation scheme (FPTAS) for inference. For most probable explanations, the problem remains polynomial. Similarly, we could allow for different groundings of the same relation to be associated with different probabilities; the proofs given here can be modified to develop a FPTAS for inference. This implies that both *probabilistic relational models (PRMs)* [17] and *recursive relational Bayesian networks (RRBNs)* [22], when appropriately restricted to DL-Lite constructs, allow for inference through FPTAS. We intend to pursue details of such conceptual modeling tools in the future.

Other possible extensions of our results merit attention. First, one might investigate whether there are similar polynomial/FPTAS results not only for the many existing variants of DL-Lite [1], but also for networks specified through other popular description logics such as $\mathcal{EL}$ and $\mathcal{ALC}$ [2], or even other languages such as temporal logics.

To conclude, DL-Lite Bayesian networks offer a flexible and effective language, that can be used to specify probabilistic acyclic ontologies or entity-relationship diagrams. Usual services, such as inference and explanations, have tractable algorithms that can be used directly or called during learning.

Acknowledgements

The second author was partially supported by CNPq.

References

1. Artale, A., Calvanese, D., Kontchakov, R., Zakharyashev, M.: The *dl-lite* family and relations. Journal of Artificial Intelligence Research 36, 1–69 (2009)
2. Baader, F., Nutt, W.: Description Logic Handbook, chap. Basic description logics, pp. 47–100. Cambridge University Press (2002)
3. Bacchus, F.: Representing and Reasoning with Probabilistic Knowledge: A Logical Approach. MIT Press (1990)
4. van den Broeck, G.: On the completeness of first-order knowledge compilation for lifted probabilistic inference. In: Advances in Neural Processing Information Systems (2011)
5. van den Broeck, G., Wannes, M., Darwiche, A.: Skolemization for weighted first-order model counting. In: Proc. of the International Conference on Principles of Knowledge Representation and Reasoning (2014)
6. Cai, J.Y., Lu, P., Xia, M.: Holographic reduction, interpolation and hardness. Computational Complexity 21(4), 573–604 (2012)
7. Calvanese, D., De Giacomo, G., Lembo, D. abd Lenzerini, M., Rosati, R.: *DL-Lite*: Tractable description logics for ontologies. In: Proc. of the AAAI Conference. pp. 602–607 (2005)
8. Ceylan, I., Peñaloza, R.: The bayesian description logic ⌊⌉↕. In: Proc. of the 7th International Joint Conference on Automated Reasoning. pp. 480–494 (2014)
9. Chavira, M., Darwiche, A.: Compiling bayesian networks with local structure. In: Proc. of the Nineteenth International Joint Conference on Artificial Intelligence. pp. 1306–1312 (2005)
10. Cozman, F.G., Mauá, D.D.: Bayesian networks specified using propositional and relational constructs: Combined, data, and domain complexity. In: Proc. of the 29th AAAI Conference on Artificial Intelligence. pp. 3519–3525 (2015)
11. Dagum, P., Luby, M.: Approximating probabilistic inference in Bayesian belief networks is NP-hard. Artif. Intell. 60 (1), 141–153 (1993)
12. d'Amato, C., Fanizzi, N., Lukasiewicz, T.: Tractable reasoning with Bayesian description logics. In: Proc. of the 2nd International Conference on Scalable Uncertainty Management. pp. 146–159 (2008)
13. D'Ambrosio, B.: Local expression languages for probabilistic dependence. International Journal of Approximate Reasoning 13(1), 61–81 (1995)
14. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press (2009)

15. Domingos, P., Webb, W.: A tractable first-order probabilistic logic. In: Proc. of the AAAI Conference on Artificial Intelligence (2012)
16. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: Proc. of the International Joint Conference on Artificial Intelligence. pp. 1300–1309 (1999)
17. Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning. MIT Press (2007)
18. Halpern, J.Y.: Reasoning about uncertainty. MIT Press (2003)
19. Heckerman, D.: A tractable inference algorithm for diagnosing multiple diseases. In: Proc. of the 5th Conference on Uncertainty in Artificial Intelligence. pp. 174–181 (1989)
20. Heinsohn, J.: Probabilistic description logics. In: Proc. of the 10th International Conference on Uncertainty in Artificial Intelligence. pp. 311–318 (1994)
21. Jaeger, M.: Relational Bayesian networks. In: Proc. of the Conference on Uncertainty in Artificial Intelligence. pp. 266–273 (1997)
22. Jaeger, M.: Complex probabilistic modeling with recursive relational Bayesian networks. In: Annals of Mathematics and Artificial Intelligence, vol. 32, pp. 179–220 (2001)
23. Jaeger, M., van Den Broeck, G.: Liftability of probabilistic inference: Upper and lower bounds. In: Proc. of the 2nd Statistical Relational AI Workshop (2012)
24. Koller, D., Friedman, N.: Probabilistic Graphical Models. MIT press (2009)
25. Kwisthout, J.: Treewidth and the computational complexity of MAP approximations. In: Proc. of the 7th European Workshop on Probabilistic Graphical Models. pp. 271–285 (2014)
26. Kwisthout, J.H.P., Bodlaender, H.L., van der Gaag, L.C.: The necessity of bounded treewidth for efficient inference in Bayesian networks. In: Proc. of the 19th European Conference on Artificial Intelligence. pp. 237–242 (2010)
27. Lin, C., Liu, J., Lu, P.: A simple FPTAS for counting edge covers. In: Proc. of the 8th Annual ACM-SIAM Symposium on Discrete Algorightms. pp. 341–348 (2014)
28. Liu, J., Lu, P., Zhang, C.: FPTAS for counting weighted edge covers. In: Proc. of the 22nd Annual European Symposium on Algorithms. pp. 654–665 (2014)
29. Lowd, D., Rooshenas, A.: Learning markov networks with arithmetic circuits. In: Proc. of the 16th International Conference on Artificial Intelligence and Statistics. pp. 406–414 (2013)
30. Mauá, D.D., Cozman, F.G.: A tractable class of model counting problems. Tech. rep., Decision Making Laboratory, University of São Paulo (2015)
31. Poon, H., Domingos, P.: Sum-product networks: A new deep architecture. In: Proc. of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence. pp. 337–346 (2011)
32. de Raedt, L.: Logical and Relational Learning. Springer (2008)
33. de Raedt, L., Frasconi, P., Kersting, K., Muggleton, S.H.: Probabilistic Inductive Logic Programming. Springer (2010)
34. Ramachandran, R., Qi, G., Wang, K., Wang, J., Thornton, J.: Probabilistic reasoning in DL-lite. In: Proc. of the 12th Pacific Rim International Conference on Trends in Artificial Intelligence. pp. 480–491 (2012)
35. Rosenkrantz, D.J., Marathe, M.V., s. Ravi, S., Vullikanti, A.K.: Bayesian inference in treewidth-bounded graphical models without indegree constraints. In: Proc. of the 30th Conference on Uncertainty in Artificial Intelligence. pp. 702–711 (2014)
36. Roth, D.: On the hardness of approximate reasoning. Artificial Intelligence 82(1–2), 273–302 (1996)