Probabilistic Query Answering in the Bayesian Description Logic $\mathcal{BEL}$*

İsmail İlkan Ceylan[1] and Rafael Peñaloza[2]

[1] Theoretical Computer Science, TU Dresden, Germany
ceylan@tcs.inf.tu-dresden.de
[2] KRDB Research Centre, Free University of Bozen-Bolzano, Italy
rafael.penaloza@unibz.it

Abstract. $\mathcal{BEL}$ is a probabilistic description logic (DL) that extends the light-weight DL $\mathcal{EL}$ with a joint probability distribution over the axioms, expressed with the help of a Bayesian network (BN). In recent work it has been shown that the complexity of standard logical reasoning in $\mathcal{BEL}$ is the same as performing probabilistic inferences over the BN.
In this paper we consider conjunctive query answering in $\mathcal{BEL}$. We study the complexity of the three main problems associated to this setting: computing the probability of a query entailment, computing the most probable answers to a query, and computing the most probable context in which a query is entailed. In particular, we show that all these problems are tractable w.r.t. data and ontology complexity.

1 Introduction

Description Logics (DLs) [3] are a family of knowledge representation formalisms that have been successfully employed for modeling the knowledge of many application domains. Its success has been specially clear in the bio-medical sciences, with the development and use of very large ontologies [29]. Very briefly, an ontology is simply a collection of *axioms* that provide some explicit knowledge of the application domain; different reasoning tasks are then used to extract additional knowledge that is implicit within this ontology.

As with most logic-based formalisms, one of the issues that limit the applicability of DLs to real-world ontologies is their incapability to model and handle uncertainty in their statements. To address this limitation, many extensions of DLs for reasoning with uncertainty have been proposed over the last two decades; see e.g. [24] for a thorough, although slightly outdated, survey. A very relevant modeling choice that needs to be made is how to represent and handle the joint probability of axioms. Most probabilistic extensions of DLs avoid this problem by implicitly assuming that all axioms are (probabilistically) independent from

* This work was partially supported by DFG within the Research Training Group "RoSI" (GRK 1907) and the Cluster of Excellence 'cfAED.' Most of the work was developed while R. Peñaloza was still affiliated with TU Dresden and the Center for Advancing Electronics Dresden, Germany.

each other. Unfortunately, this is a very strong assumption that cannot be guaranteed to hold in general. Very recently, it was proposed to represent the logical and probabilistic dependencies of the axioms in an ontology through a Bayesian network (BN) ranging over a class of sub-ontologies, called contexts. This idea gave rise to the family of Bayesian DLs [9].

To understand the properties of Bayesian DLs, the complexity of standard reasoning on $\mathcal{BEL}$, the Bayesian extension of the light-weight DL $\mathcal{EL}$ [2,6], was studied in detail. In particular, it was shown that standard reasoning in this logic remains tractable w.r.t. the size of the logical component of the input, although intractable w.r.t. the BN [11,12]. These analysis have also shown their impact in practice, we refer the reader to the recent prototypical reasoner BORN [8] for such details (available at `http://lat.inf.tu-dresden.de/systems/born`).

In this paper we build on top of previous work [7], and study the complexity of answering conjunctive queries over a probabilistic knowledge base expressed in $\mathcal{BEL}$. Given the probabilistic knowledge, we focus on computing the probability of entailing a given query. Moreover, we study the problem of finding the most probable answers to a query, and the most probable contexts that entail a query. As is standard in query answering, we parameterize the complexity measures according to different input parameters. Among our results, we show that all the reasoning problems that we study remain tractable w.r.t. the size of the ontology. This means that it is possible to handle large ontologies efficiently, assuming that the probabilistic component and the query remain relatively small.

2 Preliminaries

We first briefly introduce the basic notions for query answering in the light-weight DL $\mathcal{EL}$ and its Bayesian extension $\mathcal{BEL}$, and the complexity measures that we will study throughout this paper.

As with all DLs, the main components of $\mathcal{EL}$ are concepts, that are built from concept- and role-names using a set of constructors. Let $\mathsf{N_I}$, $\mathsf{N_C}$ and $\mathsf{N_R}$ be mutually disjoint sets of *individual-*, *concept-* and *role-names*, respectively. $\mathcal{EL}$ *concepts* are built by the grammar rule $C ::= A \mid \top \mid C \sqcap C \mid \exists r.C$, where $A \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$. The *semantics* of $\mathcal{EL}$ is given by interpretations. An *interpretation* is a tuple $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ where $\Delta^{\mathcal{I}}$ is a non-empty *domain* and $\cdot^{\mathcal{I}}$ is an *interpretation function* that maps every individual name a to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, every concept name A to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, and every role name r to a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The interpretation function $\cdot^{\mathcal{I}}$ is extended to $\mathcal{EL}$ concepts as shown in the upper part of Table 1.

The domain knowledge is encoded through a set of axioms that restrict the class of interpretations considered. A *TBox* $\mathcal{T}$ is a finite set of *general concept inclusions (GCIs)* of the form $C \sqsubseteq D$, where C, D are concepts. An *ABox* is a finite set of *concept assertions* $C(a)$ and *role assertions* $r(a, b)$, where $a, b \in \mathsf{N_I}$, C is a concept, and $r \in \mathsf{N_R}$. An *ontology* is a pair $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ where $\mathcal{T}$ is a TBox and $\mathcal{A}$ an ABox. We use the term *axiom* as a general expression for GCIs and assertions. The interpretation $\mathcal{I}$ *satisfies* an axiom λ iff it satisfies the conditions

Table 1: Syntax and Semantics of $\mathcal{EL}$

Name	Syntax	Semantics
Top	$\top$	$\Delta^{\mathcal{I}}$
Conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Exist. Rest.	$\exists r.C$	$\{d \mid \exists e \in \Delta^{\mathcal{I}} : (d,e) \in r^{\mathcal{I}}, e \in C^{\mathcal{I}}\}$
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
Concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
Role assertion	$r(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$

on the lower part of Table 1. It is a *model* of the ontology $\mathcal{O} = (\mathcal{T}, \mathcal{A})$ iff it satisfies all the axioms in $\mathcal{T}$ and $\mathcal{A}$. For the rest of this paper we will denote as $\mathsf{N_I}(\mathcal{A})$ the set of all individual names that appear in the ABox $\mathcal{A}$.

In the presence of an ontology, one is often interested in deciding entailment and finding answers to a (conjunctive) query. Let $\mathsf{N_V}$ be a set of *variables*, which is disjoint from $\mathsf{N_C}$, $\mathsf{N_R}$, and $\mathsf{N_I}$. An *atom* is an expression of the form $A(\chi)$ or $r(\chi, \psi)$, where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and $\chi, \psi \in \mathsf{N_I} \cup \mathsf{N_V}$. A *conjunctive query* (CQ) q is a non-empty set of atoms associated to a set $\mathsf{DV}(\mathsf{q}) \subseteq \mathsf{N_V}$ of *distinguished variables*. If $\mathsf{DV}(\mathsf{q}) = \emptyset$, then q is called a *Boolean CQ*. A special case of a CQ is an *instance query*, which consists of only one atom $A(\chi)$ with $A \in \mathsf{N_C}$.

Let q be a Boolean CQ and $\mathsf{IV}(\mathsf{q})$ be the set of all individual names and variables appearing in q. The interpretation $\mathcal{I}$ *satisfies* q if there exists a function $\pi : \mathsf{IV}(\mathsf{q}) \to \Delta^{\mathcal{I}}$ such that (i) $\pi(a) = a^{\mathcal{I}}$ for all $a \in \mathsf{N_I} \cap \mathsf{IV}(\mathsf{q})$, (ii) $\pi(\chi) \in A^{\mathcal{I}}$ for all $A(\chi) \in \mathsf{q}$, and (iii) $(\pi(\chi), \pi(\psi)) \in r^{\mathcal{I}}$ for all $r(\chi, \psi) \in \mathsf{q}$. In this case, we call π a *match* for $\mathcal{I}$ and q. The ontology $\mathcal{O}$ *entails* q ($\mathcal{O} \models \mathsf{q}$) iff every model of $\mathcal{O}$ satisfies q. For an arbitrary CQ q, a function $\mathfrak{a} : \mathsf{DV}(\mathsf{q}) \to \mathsf{N_I}(\mathcal{A})$ is an *answer* to q w.r.t. $\mathcal{O}$ iff $\mathcal{O}$ entails the Boolean CQ $\mathfrak{a}(\mathsf{q})$ obtained by replacing every distinguished variable $\chi \in \mathsf{DV}(\mathsf{q})$ with $\mathfrak{a}(\chi)$. *Conjunctive query answering* (CQA) is the task of finding all answers of a CQ, and query entailment is the problem of deciding whether an ontology entails a given Boolean CQ.

It is known that in $\mathcal{EL}$ query entailment is tractable if the query is fixed, but NP-complete if the query is considered as part of the input [27]. $\mathcal{EL}$ does not enjoy the so-called full *first order rewritability* which has been considered as a key feature for CQA, since it allows one to reduce the problem to standard tasks in relational database management systems. However, other methods like the combined approach [26] have been successfully used in this setting.

The Bayesian DL $\mathcal{BEL}$ [11] has been introduced as a probabilistic extension of $\mathcal{EL}$. In $\mathcal{BEL}$ probabilities are encoded through a *Bayesian network* (BN) [17]; that is, a pair $\mathcal{B} = (G, \Phi)$, where $G = (V, E)$ is a finite directed acyclic graph (DAG) whose nodes represent Boolean random variables, and Φ contains, for every node $x \in V$, a conditional probability distribution $P_{\mathcal{B}}(x \mid \pi(x))$ of x given its parents $\mathsf{pa}(x)$. If V is the set of nodes in G, we say that $\mathcal{B}$ is a BN *over* V. In a BN, every variable $x \in V$ is considered to be conditionally independent of its non-descendants given its parents. Thus, every BN $\mathcal{B}$ defines a unique joint

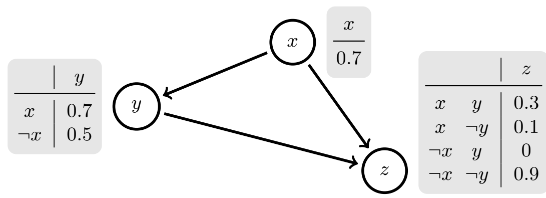

Fig. 1: The BN $\mathcal{B}_{\mathtt{ABC}}$ over the variables $\{x, y, z\}$

probability distribution over V given by the so-called *chain rule*, defined as

$$P_{\mathcal{B}}(V) = \prod_{x \in V} P_{\mathcal{B}}(x \mid \pi(x)).$$

In $\mathcal{BEL}$ concepts are constructed as for $\mathcal{EL}$. The difference appears in encoding the domain knowledge through axioms. $\mathcal{BEL}$ generalizes classical ontologies by annotating the axioms with a context, defined by a set of literals from a BN.

Let V be a finite set of Boolean variables. A *V-context* is a conjunction of literals from V. A *(V-GCI)(resp. V-assertion)* is an expression of the form $\langle \lambda : \kappa \rangle$ where λ is a GCI (resp. an assertion) and κ is a V-context. A *V-TBox (resp. V-ABox)* is a finite set of V-GCIs (resp. V-assertions). A *$\mathcal{BEL}$ knowledge base* (KB) is a tuple $\mathcal{K} = (\mathcal{B}, \mathcal{T}, \mathcal{A})$ where $\mathcal{B}$ is a BN over V, $\mathcal{T}$ is a V-TBox and $\mathcal{A}$ is a V-ABox.

Example 1. The tuple $\mathcal{K} = (\mathcal{T}_{\mathtt{ABC}}, \mathcal{A}_{\mathtt{ABC}}, \mathcal{B}_{\mathtt{ABC}})$ where

$$\mathcal{T}_{\mathtt{ABC}} := \{\, \langle \mathtt{A} \sqsubseteq \exists \mathtt{r}.\mathtt{B} : \{y\} \rangle \,, \langle \mathtt{B} \sqsubseteq \mathtt{C} : \{x\} \rangle \}$$
$$\mathcal{A}_{\mathtt{ABC}} := \{\, \langle \mathtt{A}(\mathtt{a}) : \{\mathtt{x}\} \rangle \,, \langle \mathtt{r}(\mathtt{a}, \mathtt{b}) : \{\mathtt{z}\} \rangle \,, \langle \mathtt{C}(\mathtt{b}) : \{x, z\} \rangle \,, \langle \mathtt{A}(\mathtt{c}) : \{\mathtt{y}\} \rangle \}$$

and $\mathcal{B}_{\mathtt{ABC}}$ is the BN given in Figure 1 represents a $\mathcal{BEL}$ KB.

Intuitively, a $\mathcal{BEL}$ KB provides a propositional abstraction over an $\mathcal{EL}$ KB. More formally, given a $\mathcal{BEL}$ KB $\mathcal{K} = (\mathcal{B}, \mathcal{T}, \mathcal{A})$ and a context κ, we define the *restriction of $\mathcal{K}$ w.r.t.* κ as an $\mathcal{EL}$ ontology $\mathcal{K}_\kappa = (\mathcal{T}_\kappa, \mathcal{A}_\kappa)$ by setting

$$\mathcal{T}_\kappa := \{C \sqsubseteq D \mid \langle C \sqsubseteq D : \mu \rangle \in \mathcal{T},\ \kappa \models \mu\},$$
$$\mathcal{A}_\kappa := \{C(a) \mid \langle C(a) : \mu \rangle \in \mathcal{A}, \kappa \models \mu\} \cup \{r(a, b) \mid \langle r(a, b) : \mu \rangle \in \mathcal{A}, \kappa \models \mu\}.$$

We will usually speak of *contextual axioms*, or *V-axioms* to address both V-GCIs and V-assertions; if it is clear from the context, we will also drop the prefix V. The intuition behind the contextual axioms is to enforce an axiom to hold within a given context, but not necessarily in others. To formalize this intuition, we extend the notion of an interpretation, to also consider the context variables. A *V-interpretation* is a tuple $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, \mathcal{V}^{\mathcal{I}})$ where $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a classical $\mathcal{EL}$ interpretation, and $\mathcal{V}^{\mathcal{I}}$ is a valuation of the variables in V. The V-interpretation $\mathcal{I}$ *satisfies* the axiom $\langle \lambda : \kappa \rangle$ ($\mathcal{I} \models \langle \lambda : \kappa \rangle$) iff either (i) $\mathcal{V}^{\mathcal{I}} \not\models \kappa$, or (ii) $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}) \models \lambda$. It is a *model* of the TBox $\mathcal{T}$ (resp. ABox $\mathcal{A}$) iff it satisfies all the axioms in $\mathcal{T}$ (resp. $\mathcal{A}$).

There is a strong link between the restrictions and the contextual interpretations. For any valuation $\mathcal{W}$ of the variables in V, $\mathcal{K}_\mathcal{W}$ represents all the $\mathcal{EL}$ axioms that must be satisfied by any contextual interpretation of the form $(\Delta^\mathcal{I}, \cdot^\mathcal{I}, \mathcal{W})$.

In $\mathcal{BEL}$, uncertainty is represented through a BN that describes a joint probability distribution over the context variables. $\mathcal{BEL}$ is linked to this distribution using *multiple world semantics*: a probabilistic interpretation defines a probability distribution over a set of (contextual) interpretations; this distribution is required to be consistent with the joint probability distribution provided by the BN. Formally, a *probabilistic interpretation* is a pair $\mathcal{P} = (\mathfrak{I}, P_\mathfrak{I})$, where $\mathfrak{I}$ is a set of V-interpretations and $P_\mathfrak{I}$ is a probability distribution over $\mathfrak{I}$ such that $P_\mathfrak{I}(\mathcal{I}) > 0$ only for finitely many interpretations $\mathcal{I} \in \mathfrak{I}$. $\mathcal{P}$ is a *model* of the TBox $\mathcal{T}$ (resp. ABox $\mathcal{A}$) if every $\mathcal{I} \in \mathfrak{I}$ is a model of $\mathcal{T}$ (resp. $\mathcal{A}$). $\mathcal{P}$ is *consistent* with the BN $\mathcal{B}$ if for every valuation $\mathcal{W}$ of the variables in V it holds that

$$\sum_{\mathcal{I} \in \mathfrak{I}, \mathcal{V}^\mathcal{I} = \mathcal{W}} P_\mathfrak{I}(\mathcal{I}) = P_\mathcal{B}(\mathcal{W}).$$

The probabilistic interpretation $\mathcal{P}$ is a *model* of the KB $(\mathcal{B}, \mathcal{T}, \mathcal{A})$ iff it is a probabilistic model of $\mathcal{T}$, $\mathcal{A}$ and consistent with $\mathcal{B}$.

In previous work, the standard reasoning problems for $\mathcal{EL}$ have been extended to their probabilistic variant in $\mathcal{BEL}$, leading to tight complexity bounds for several problems [11, 12]. Particularly, it has been shown that the complexity of these tasks is bounded by the complexity of reasoning in $\mathcal{EL}$ and in the BN.

In the next sections we will study the complexity of different query-related reasoning tasks in $\mathcal{BEL}$. As is customary in the context of conjunctive queries, we will consider the complexity w.r.t. different parameters. The measures we consider here are: (i) *data complexity*, where only the ABox is considered as part of the input; (ii) *ontology complexity*, which considers both, the ABox and the TBox; (iii) *network complexity*, w.r.t. the size of the BN; (iv) *KB complexity*, which uses the whole KB as input; and (v) *combined complexity* in which the input is measured in terms of the KB and the query.

3 Probabilistic Query Entailment

The problem of deciding whether a Boolean CQ is entailed by a $\mathcal{BEL}$ KB is not interesting, since it ignores the probabilistic information encoded in the BN. Recall that a $\mathcal{BEL}$ KB describes a probability distribution over different worlds, in which some conditions must hold. In this setting, we are interested in finding the probability of observing a world in which the query is entailed.

Definition 2 (probabilistic entailment). *Let $\mathcal{K}$ be a $\mathcal{BEL}$ KB, $\mathcal{P} = (\mathfrak{I}, P)$ a probabilistic interpretation and q a Boolean CQ. The* probability of q w.r.t. $\mathcal{P}$ *is*

$$P_\mathcal{P}(\mathsf{q}) := \sum_{\mathcal{I} \in \mathfrak{I},\ (\Delta^\mathcal{I}, \cdot^\mathcal{I}) \models \mathsf{q}} P(\mathcal{I}).$$

The probability *of q w.r.t. $\mathcal{K}$ is $P_\mathcal{K}(\mathsf{q}) := \inf_{\mathcal{P} \models \mathcal{K}} P_\mathcal{P}(\mathsf{q})$. The query q is entailed with probability $p \in (0, 1]$ iff $P_\mathcal{K}(\mathsf{q}) \geq p$.*

Recall that for a given $\mathcal{EL}$ KB $\mathcal{K}$ and a valuation $\mathcal{W}$, $\mathcal{K}_\mathcal{W}$ defines an $\mathcal{EL}$ ontology that contains all the axioms that must be satisfied by any contextual interpretation using the valuation $\mathcal{W}$. We show that considering the restrictions $\mathcal{K}_\mathcal{W}$ over valuations $\mathcal{W}$ is enough to decide probabilistic query entailment.

Theorem 3. *For every $\mathcal{BEL}$ KB $\mathcal{K} = (\mathcal{B}, \mathcal{T}, \mathcal{A})$ and Boolean CQ q it holds that $P_\mathcal{K}(\mathsf{q}) = \sum_{\mathcal{K}_\mathcal{W} \models \mathsf{q}} P_\mathcal{B}(\mathcal{W})$.*

Proof. We define the probabilistic interpretation $\mathcal{R} = (\mathcal{J}_\mathcal{R}, P_{\mathcal{J}_\mathcal{R}})$ where

i) $\mathcal{J}_\mathcal{R} = \bigcup_{i=0}^{2^n-1} \mathcal{I}_i = (\Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i}, \mathcal{V}^{\mathcal{I}_i})$
ii) $P_{\mathcal{J}_\mathcal{R}}(\mathcal{I}_i) = P_\mathcal{B}(\mathcal{W}_i)$ with $\mathcal{W}_i = \mathcal{V}^{\mathcal{I}_i}$ for all $0 \leq i \leq 2^n - 1$
iii) $(\Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i}) \models \mathcal{K}_{\mathcal{W}_i}$ for all $0 \leq i \leq 2^n - 1$
iv) $(\Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i}) \models \mathsf{q}$ iff $\mathcal{K}_{\mathcal{W}_i} \models \mathsf{q}$ for all $0 \leq i \leq 2^n - 1$

Notice that, we can ensure iii) by the fact that every $\mathcal{EL}$ ontology has a model.

It follows from the construction that $\mathcal{R} \models (\mathcal{T}, \mathcal{A})$ and $\mathcal{R}$ is consistent with $\mathcal{B}$. Hence, $\mathcal{R}$ is a model of $\mathcal{K}$. We show the probability of q w.r.t. $\mathcal{R}$ to be

$$P_\mathcal{R}(\mathsf{q}) := \sum_{\substack{\mathcal{I}_i \in \mathcal{J}_\mathcal{R} \\ (\Delta^{\mathcal{I}_i}, \cdot^{\mathcal{I}_i}) \models \mathsf{q}}} P_{\mathcal{J}_\mathcal{R}}(\mathcal{I}_i) = \sum_{\mathcal{K}_{\mathcal{W}_i} \models \mathsf{q}} P_\mathcal{B}(\mathcal{W}_i),$$

which concludes $P_\mathcal{K}(\mathsf{q}) \leq \sum_{\mathcal{K}_\mathcal{W} \models \mathsf{q}} P_\mathcal{B}(\mathcal{W})$.
Assume now that the inequality is strict. This implies the existence of a model $\mathcal{S} = (\mathcal{J}_\mathcal{S}, P_{\mathcal{J}_\mathcal{S}})$ such that

$$P_\mathcal{S}(\mathsf{q}) = \sum_{\substack{\mathcal{I} \in \mathcal{J}_\mathcal{S} \\ (\Delta^\mathcal{I}, \cdot^\mathcal{I}) \models \mathsf{q}}} P_{\mathcal{J}_\mathcal{S}}(\mathcal{I}) < \sum_{\mathcal{K}_\mathcal{W} \models \mathsf{q}} P_\mathcal{B}(\mathcal{W}).$$

This holds iff for some $\mathcal{W}$ where $\mathcal{K}_\mathcal{W} \models \mathsf{q}$ and $P_\mathcal{B}(\mathcal{W}) > 0$ it holds that

$$\sum_{\substack{(\Delta^\mathcal{I}, \cdot^\mathcal{I}, \mathcal{W}) \in \mathcal{J}_\mathcal{S} \\ (\Delta^\mathcal{I}, \cdot^\mathcal{I}) \models \mathsf{q}}} P_{\mathcal{J}_\mathcal{S}}(\mathcal{I}) < P_\mathcal{B}(\mathcal{W}).$$

Since $\sum_{\mathcal{I} \in \mathcal{J}_\mathcal{S}, \mathcal{V}^\mathcal{I} = \mathcal{W}} P_{\mathcal{J}_\mathcal{S}}(\mathcal{I}) = P_\mathcal{B}(\mathcal{W})$ by the definition of a model, there exists a contextual interpretation $(\Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'}, \mathcal{V}^{\mathcal{I}'}) \in \mathcal{J}_\mathcal{S}$ where $\mathcal{V}^{\mathcal{I}'} = \mathcal{W}$ and $(\Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'}) \not\models \mathsf{q}$ while $\mathcal{K}_{\mathcal{W}'} \models \mathsf{q}$. It follows that $(\Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'}) \not\models \mathcal{K}_\mathcal{W}$ and $(\Delta^{\mathcal{I}'}, \cdot^{\mathcal{I}'}, \mathcal{V}^{\mathcal{I}'}) \not\models (\mathcal{T}, \mathcal{A})$, which contradicts with the assumption that $\mathcal{S}$ is a model. □

Theorem 3 provides a simple method for computing the probability of a query q w.r.t. a $\mathcal{BEL}$ KB: one needs only to compute, for each valuation $\mathcal{W}$, the $\mathcal{EL}$ ontology $\mathcal{K}_\mathcal{W}$ and decide whether this ontology entails q, adding the probabilities (w.r.t. $\mathcal{B}$) of all the worlds for which this test is positive. We illustrate probabilistic query entailment on our running example.

Example 4. Consider the $\mathcal{BEL}$ KB provided in Example 1 and the Boolean CQ $\mathsf{q} = \{A(\chi), r(\chi, \psi), C(\psi)\}$. Clearly, $\mathcal{K}_\mathcal{W} \models \mathsf{q}$ only for worlds $\mathcal{W}$ such that $\mathcal{W} \models (x \wedge y) \vee (x \wedge z)$. Hence, we get $P_\mathcal{K}(\mathsf{q}) = P_{\mathcal{B}_{\mathsf{ABC}}}((x \wedge y) \vee (x \wedge z)) = 0.511$.

Since there are $2^{|V|}$ valuations, $\mathcal{EL}$ query entailment is decidable in polynomial time in ontology complexity, and computing the probability of a valuation is polynomial in $|V|$, we obtain the following result.

Theorem 5. *Probabilistic query entailment is polynomial w.r.t. data and ontology complexity and in* ExpTime *w.r.t. network, KB, and combined complexity.*

Notice that the algorithm sketched above iterates over all the possible scenarios described by the BN and performs an entailment test in each of them. The positive complexity results w.r.t. data and ontology complexity arise from the fact that in these settings the size of the BN is assumed to be constant. In order to obtain a better upper bound w.r.t. network complexity, we can dualize this idea; i.e., iterate over all the sub-ontologies performing standard probabilistic inferences at each iteration.

Theorem 6. *Probabilistic query entailment is* PP*-complete w.r.t. network complexity.*

Proof. The lower complexity bound follows from the complexity of standard reasoning in $\mathcal{BEL}$ [12]. To show membership, we define a *sub-ontology* of a given $\mathcal{BEL}$ KB $\mathcal{K} = (\mathcal{B}, \mathcal{T}, \mathcal{A})$ as a pair $\mathcal{O} = (\mathcal{T}', \mathcal{A}')$ such that $\mathcal{T}' \subseteq \mathcal{T}$ and $\mathcal{A}' \subseteq \mathcal{A}$. Each sub-ontology $\mathcal{O} = (\mathcal{T}', \mathcal{A}')$ defines a context

$$\mathsf{con}(\mathcal{O}) = \bigwedge_{\langle\lambda:\kappa\rangle\in\mathcal{T}'} \kappa \wedge \bigwedge_{\langle\lambda:\kappa\rangle\in\mathcal{A}'} \kappa,$$

and an $\mathcal{EL}$ ontology $\mathcal{O}_{\mathcal{EL}} = (\mathcal{T}'_{\mathcal{EL}}, \mathcal{A}'_{\mathcal{EL}})$

$$\begin{aligned}
\mathcal{T}'_{\mathcal{EL}} &:= \{C \sqsubseteq D \mid \langle C \sqsubseteq D : \kappa\rangle \in \mathcal{T}' \text{ for some context } \kappa\},\\
\mathcal{A}'_{\mathcal{EL}} &:= \{C(a) \mid \langle C(a) : \kappa\rangle \in \mathcal{A}' \text{ for some context } \kappa\} \cup\\
&\ \{r(a,b) \mid \langle r(a,b) : \kappa\rangle \in \mathcal{A}' \text{ for some context } \kappa\}.
\end{aligned}$$

For every contextual interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, \mathcal{V}^{\mathcal{I}})$ with $\mathcal{I} \models (\mathcal{T}, \mathcal{A})$, we observe that if $\mathcal{V}^{\mathcal{I}} \models \mathsf{con}(\mathcal{O})$, then $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}) \models \mathcal{O}_{\mathcal{EL}}$. For a given Boolean CQ q, we define

$$\mathsf{con}_{\mathcal{K}}(\mathsf{q}) := \bigvee_{\mathcal{O}_{\mathcal{EL}} \models \mathsf{q}} \mathsf{con}(\mathcal{O}).$$

From Theorem 3, we know that $P_{\mathcal{K}}(\mathsf{q}) = P_{\mathcal{B}}(\mathsf{con}(\mathsf{q}))$. Thus, it suffices to compute the probability of the DNF formula $\mathsf{con}(\mathsf{q})$ to obtain the probability of the query q. Since Bayesian network inferences are PP-complete [28], and the class PP is closed under intersection and complementation [4], it follows that probabilistic query entailment is also in PP w.r.t. network complexity . □

We consider now the case of combined complexity, in which the ontology, the BN, and the query are all considered as part of the input. We show that in this case, the complexity of probabilistic query entailment is at most PSpace.

Theorem 7. *Probabilistic query entailment is in* PSPACE *w.r.t. combined complexity.*

Proof. Theorem 3 ensures that to compute $P_{\mathcal{K}}(\mathsf{q})$ it suffices to check for every valuation $\mathcal{W}$, whether $\mathcal{K}_{\mathcal{W}} \models \mathsf{q}$, and in case it does, compute $P_{\mathcal{B}}(\mathcal{W})$. $\mathcal{K}_{\mathcal{W}}$ can be constructed by adding all axioms λ to $\mathcal{K}_{\mathcal{W}}$ where $\langle \lambda : \kappa \rangle \in \mathcal{K}$ and $\mathcal{W} \models \kappa$. This requires only linear time on both $|\mathcal{K}|$ and $|V|$. Deciding whether $\mathcal{K}_{\mathcal{W}} \models \mathsf{q}$ is an NP-complete problem w.r.t. the sizes of $\mathcal{K}$ and q. Finally, $P_{\mathcal{B}}(\mathcal{W})$ can be computed in time polynomial on the size of $\mathcal{B}$, using the chain rule for BNs. A PSPACE algorithm avoids storing exponentially many valuations of the variables in V simultaneously; instead iterates for each valuation independently. □

Obviously, this result also yields a PSPACE upper bound for this problem w.r.t. KB complexity. In terms of lower bounds, Theorem 6 shows that probabilistic query entailment is also PP-hard w.r.t. KB and combined complexity. Unfortunately, we were unable to obtain tight complexity bounds for these measures.

4 Probabilistic Query Answering

In query answering we do not restrict to Boolean CQs anymore, but consider queries that may contain distinguished variables. As described before, in this case we are interested in finding the possible substitutions of these distinguished variables into individuals appearing in the ontology such that the resulting Boolean CQ is entailed; these substitutions are called *answers*. To find all these answers, one could simply perform a query entailment test for each of the possible substitutions. There are exponentially many such substitutions, measured on the number of individuals in the ontology, and potentially all of them can be answer to a given query, and receiving so many results might be uninformative to a user. Rather than providing all possible answers to a query, we are interested in finding a limited number of them having the highest probability of being entailed.

Let q be a query with the distinguished variables $\mathsf{DV}(\mathsf{q})$, and $\mathcal{K} = (\mathcal{B}, \mathcal{T}, \mathcal{A})$ a $\mathcal{BEL}$ KB. Recall that every function $\mathfrak{a} : \mathsf{DV}(\mathsf{q}) \to \mathsf{N_I}(\mathcal{A})$ defines a Boolean CQ obtained by replacing every $\chi \in \mathsf{DV}(\mathsf{q})$ in q with $\mathfrak{a}(\chi)$. Abusing of the notation, we call this query $\mathfrak{a}(\mathsf{q})$. We call any function $\mathfrak{a} : \mathsf{DV}(\mathsf{q}) \to \mathsf{N_I}(\mathcal{A})$ an *answer* to q w.r.t. $\mathcal{K}$, and define its probability as $P_{\mathcal{K}}(\mathfrak{a}) := P_{\mathcal{K}}(\mathfrak{a}(\mathsf{q}))$. Clearly, since an answer defines a Boolean CQ, computing the probability of such an answer is exactly as hard as probabilistic query entailment in all measures considered. We use this probability as a means to identify the most relevant answers, returning only those that are most likely to be observed.

Definition 8 (top-k answer). *Let* q *be a query,* $\mathcal{K}$ *be a* $\mathcal{BEL}$ *KB, and* $k \in \mathbb{N}$. *A* top-k answer *to* q *w.r.t.* $\mathcal{K}$ *is a tuple* $(\mathfrak{a}_1, \dots, \mathfrak{a}_k)$ *of different answers to* q *w.r.t.* $\mathcal{K}$ *such that (i) for all* $i, 1 \leq i < k$, $P_{\mathcal{K}}(\mathfrak{a}_i) \geq P_{\mathcal{K}}(\mathfrak{a}_{i+1})$, *and (ii) for every other answer* $\mathfrak{a}$, $P_{\mathcal{K}}(\mathfrak{a}_k) \geq P_{\mathcal{K}}(\mathfrak{a})$.

In other words, a top-k answer is an ordered tuple of the k answers that have the highest probability. We assume that k is a constant that is fixed *a priori*. Thus, it is not considered part of the input of the problem. Obviously, since different answers may have the same degree, top-k answers are not unique. Here we are only interested in finding one of these tuples. Stating it as a decision problem, we want to verify whether a given tuple is a top-k answer.

Example 9. Consider the $\mathcal{BEL}$ KB $\mathcal{K} = (\mathcal{T}_{\mathtt{ABC}}, \mathcal{A}_{\mathtt{ABC}}, \mathcal{B}_{\mathtt{ABC}})$ provided in Example 1 and the query $\mathsf{q} = \{A(\chi)\}$ with $\chi \in \mathsf{DV}$. We are interested in identifying the top-1 answer to q w.r.t. $\mathcal{K}$. Notice that both $\mathfrak{a}_0 : \chi \mapsto a$ and $\mathfrak{a}_1 : \chi \mapsto c$ are answers to q with positive probability. Clearly, $\mathfrak{a}_0$ is the top-1 answer since $P_{\mathcal{K}}(\mathfrak{a}_0) > P_{\mathcal{K}}(\mathfrak{a}_1)$.

Assuming that the size of q and the BN $\mathcal{B}$ are fixed, then there are polynomially many answers to q w.r.t. $\mathcal{K}$, and for each answer $\mathfrak{a}$, we can compute $P_{\mathcal{B}}(\mathfrak{a})$ performing constantly many $\mathcal{EL}$ query entailment tests. Thus, it is possible to verify whether $(\mathfrak{a}_1, \dots, \mathfrak{a}_k)$ is a top-k answer in polynomial time w.r.t. ontology complexity. Likewise, if the ontology and the query are constant, then we can compute $P_{\mathcal{B}}(\mathfrak{a})$ through constantly many probabilistic inferences in the BN, as described in the previous section. Overall, we obtain the following result.

Corollary 10. *Deciding top-k answers is in* PTime *w.r.t. data and ontology complexity and* PP*-complete w.r.t. network complexity.*

We consider now the case of the combined complexity, in which all the elements are considered as part of the input and show that our problem is at least hard to the level coNP^{PP}; that is a class known to be between PH (the limit of the polynomial hierarchy) and PSpace [30]

Theorem 11. *Deciding whether a tuple $\mathfrak{A}$ is a top-k answer is co*NP^{PP}*-hard w.r.t. KB complexity.*

Proof. We provide a reduction from the decision version of the *maximum a-posteriori* (D-MAP) problem for BNs [17]. Formally, given a BN $\mathcal{B}$ over V, a set $Q \subseteq V$, a context κ, and $p > 0$, the D-MAP problem consists of deciding whether there exists a valuation μ of the variables in Q such that $P_{\mathcal{B}}(\kappa \wedge \mu) > p$. Consider an arbitrary but fixed instance of D-MAP described by the BN $\mathcal{B} = ((V, E), \Phi)$, the context κ, $Q \subseteq V$, and $p > 0$. We introduce a new Boolean random variable z not appearing in V. Using this variable, we construct a new DAG (V', E) with $V' = V \cup \{z\}$ and a new BN $\mathcal{B}' = ((V', E), \Phi')$, where $P_{\mathcal{B}'}(v \mid \mathsf{pa}(v)) = P_{\mathcal{B}}(v \mid \mathsf{pa}(x))$ for all $v \in V$, and $P_{\mathcal{B}'}(z) = p$. Consider the $\mathcal{BEL}$ KB $\mathcal{K} = (\mathcal{B}', \emptyset, \mathcal{A})$ where

$$\mathcal{A} := \{\langle A_x(a_x) : x\rangle, \langle A_x(b_x) : \neg x\rangle, \langle A_x(c) : z\rangle \mid x \in Q\} \cup \{\langle B(a) : \kappa\rangle, \langle B(c) : z\rangle\},$$

and query $\mathsf{q} := \{A_x(\chi_x) \mid x \in Q\} \cup \{B(\chi)\}$, where all the variables are distinguished; i.e., $\mathsf{DV}(\mathsf{q}) = \{\chi_x \mid x \in Q\} \cup \{\chi\}$. It is easy to see that the mapping $\mathfrak{a}_0 : \mathsf{DV}(\mathsf{q}) \to \{c\}$ that maps all the distinguished variables to the individual name $c \in \mathsf{N_I}(\mathcal{A})$ is an answer to this query and $P_{\mathcal{K}}(\mathfrak{a}_0) = p$. Moreover, any other answer that maps any variable to c will have probability at most p, since it can only be entailed in contexts satisfying z. Suppose that there is an answer $\mathfrak{a}$ such that $P_{\mathcal{K}}(\mathfrak{a}) > p$. This answer must map every variable χ_x to either a_x or b_x and χ to a. Let $\mu_{\mathfrak{a}} := \bigwedge_{\mathfrak{a}(\chi_x)=a_x} x \wedge \bigwedge_{\mathfrak{a}(\chi_x)=b_x} \neg x$. By construction, $\mu_{\mathfrak{a}}$ is a valuation of the variables in Q, $P_{\mathcal{B}}(\kappa \wedge \mu_{\mathfrak{a}}) > p$, and $\mathfrak{a}(\mathsf{q})$ is only entailed by valuations satisfying the context $\kappa \wedge \mu_{\mathfrak{a}}$. Overall this means that $\mathfrak{a}_0$ is *not* a top-1 answer iff there is a valuation μ of the variables in Q such that $P_{\mathcal{B}}(\kappa \wedge \mu) > p$. □

In the previous section we have shown that probabilistic query entailment is decidable in PSpace w.r.t. combined complexity. Since PSpace is a deterministic complexity class, we can in fact compute the precise probability of an entailment using only polynomial space. To show that a tuple is *not* a top-k answer, we can guess a new answer and show that its probability is strictly larger than some answer in the tuple. Overall, this means that top-k query answering remains in PSpace w.r.t. combined complexity.

Obtaining most probable answers for a query is a crucial task for the domains, where imprecise characterizations of knowledge is necessary. The next section is dedicated to another reasoning task that can be seen dual to top-k answers, namely top-k contexts.

5 Most Likely Contexts for a Query

Dually to finding the most likely answers to a query, we are also interested in finding the k most likely contexts that entail a given Boolean query q. More precisely, suppose that we have already observed that the query q holds; then, we are interested in finding out which is the current context. As in the previous section, we do not consider one, but search for a fixed number of contexts that are the most likely to hold.

As explained before, $\mathcal{K}_\kappa$ specifies the minimal conditions that must be satisfied in any contextual interpretation that satisfies the context κ. If $\mathcal{K}_\kappa$ entails the Boolean query q, then we say that q holds in context κ. We are interested in finding out the most likely contexts in which a given query holds.

Definition 12 (top-k mlc). *Let q be a CQ, $\mathcal{K}$ a $\mathcal{BEL}$ KB, and $k \in \mathbb{N}$. $\kappa_1, \ldots, \kappa_k$ are* top-k most likely contexts *(top-k mlc) for q w.r.t. $\mathcal{K}$ if $\mathcal{K}_{\kappa_i}$ entails q for all $i, 1 \leq i \leq k$; $P_{\mathcal{B}}(\kappa_i) \geq P_{\mathcal{B}}(\kappa_{i+1})$ for all $i, 1 \leq i \leq k$; and there is no other context κ such that $\mathcal{K}_\kappa \models \mathsf{q}$ and $P_{\mathcal{B}}(\kappa) > P_{\mathcal{B}}(\kappa_k)$.*

We illustrate top-k mlc with our continuing example. In this case, we are interested in finding out the 2 most likely context that entail the query.

Example 13. Consider the $\mathcal{BEL}$ KB $\mathcal{K}$ and query q provided in Example 1. Clearly all contexts κ that entail q are such that $\kappa \models \{x, y\} \vee \{x, z\}$. The top-2 contexts are then $\langle\{x, y\}, \{x, z\}\rangle$ since $P_{\mathcal{B}_{\mathtt{ABC}}}(\{x, y\}) > P_{\mathcal{B}_{\mathtt{ABC}}}(\{x, z\}) > P_{\mathcal{B}_{\mathtt{ABC}}}(\kappa)$ for any other context κ.

We show that deciding top-k mlc is tractable w.r.t. ontology complexity. Furthermore, we obtain a coNP^{PP} lower bound for the combined complexity as an analogous result to top-k answer. Differently from top-k answer; for this reasoning problem, we are able to show that this complexity bound is tight.

Theorem 14. *Top-k mlc is polynomial w.r.t. data, and ontology complexity, and co*NP^{PP}*-complete w.r.t. KB and combined complexity.*

Proof. If the BN is fixed, then the number of contexts is constant, and they can be ordered w.r.t. their complexity in constant time. The top-k mlc problem is then solved by applying a constant number of $\mathcal{EL}$ CQ entailment tests, yielding a polynomial upper bound w.r.t. ontology complexity.

For the combined complexity, coNP^{PP}-hardness is immediate since deciding one most likely context for simple queries is already coNP^{PP}-hard w.r.t. KB complexity [12]. We prove that top-k mlc is in coNP^{PP} : If a tuple is not a top-k mlc, then guess a new context κ and show using a PP oracle that $\mathcal{K}_\kappa \models \mathsf{q}$ and $P_\mathcal{B}(\kappa) > P_\mathcal{B}(\kappa_k)$. □

In terms of network complexity, a PP-hardness follows easily from the complexity of probabilistic entailment in BNs. The upper bound w.r.t. network complexity requires polynomially many calls to a PP oracle.

Theorem 15. *Top-k mlc is* PP*-hard and in* P^{PP} *w.r.t. network complexity.*

Proof. We show that top-k mlc is in P^{PP} w.r.t. networks complexity. Recall that if $\mathcal{T}$, $\mathcal{A}$ and q are fixed, then there is a constant number of contexts that entail the consequence, using only the Boolean variables that appear in $\mathcal{T}$ and $\mathcal{A}$; call this number ℓ. However, the BN $\mathcal{B}$ may also contain other variables. If $\ell < k$, then we need to expand the previously found contexts with new literals from $\mathcal{B}$ until enough contexts have been found. In the worst case, this would require a polynomial number (in the size of $\mathcal{B}$) of probabilistic entailments. Thus, this algorithm only yields a P^{PP} upper bound w.r.t. network complexity. □

To reduce the complexity of finding the most likely contexts, we consider a special case of the problem in which we are interested in full valuations of all the variables in the BN $\mathcal{B}$. We call this problem *top-k worlds*. In this case, deciding $P_\mathcal{B}(\mathcal{W}) > P_\mathcal{B}(\mathcal{W}_k)$ requires only polynomial time w.r.t. network complexity, since the chain rule of BNs yields the probability of a valuation in polynomial time. The problem is also easier than the top-k mlc w.r.t. the combined complexity: simply check whether $P_\mathcal{K}(\mathcal{W}) > P_\mathcal{K}(\mathcal{W}_k)$ and decide $\mathcal{K}_\mathcal{W} \models \mathsf{q}$, where the former can be done in time polynomial and the latter is complete for the class NP.

Notice that, top-k contexts and top-k answers are dual to each other, but they do not necessarily overlap. Consider for instance the case, where all top-k answers to a query q are retrieved from the same context κ. In this case, top-k contexts for q will contain other contexts than κ with the assumption that $k > 1$. Top-k contexts is particularly informative where the diversity of knowledge is important.

6 Related Work

Probabilistic query answering is an important reasoning task that has been widely studied in different domains such as relational databases [15,18,20], RDF

Table 2: $\mathcal{BEL}$ reasoning problems and their complexity

Problem	**data**	**ont.**	**network**	**KB**	**combined**
probabilistic entailment	P	P	PP-c	PP-h	in PSPACE
probability of an answer	P	P	PP-c	PP-h	in PSPACE
top-k answer	P	P	PP-c	PP-h	coNP^{PP}/PSPACE
top-k mlc	P	P	PP/P^{PP}	coNP^{PP}-c	coNP^{PP}-c
top-k worlds	P	P	P	coNP-c	coNP/Π_2^p

graphs [21] and XML databases [1, 22]. As mentioned before, there are many DL-based probabilistic ontology languages [24]. Surprisingly, only few of them concentrate on query answering.

In the probabilistic extension of Datalog+/- [19] authors are interested in retrieving the answers that are above a threshold value that is set *a priori*. In contrast to $\mathcal{BEL}$, in probabilistic Datalog+/- the underlying semantics is based on Markov logic networks. The Prob-DL family [25] extends classical DLs with subjective probabilities, also known as Type II probabilities [23]. The main difference with our logic is that Prob-$\mathcal{EL}$ introduces probabilities as a concept constructor, whereas we allow only probabilities over axioms.

More closely related to $\mathcal{BEL}$ is BDL-Lite [16]. As is in $\mathcal{BEL}$, BDL-Lite only allows probabilities over axioms and conditional dependencies are represented faithfully. However, as it has been pointed before [12], the authors use a closed world assumption, which easily leads to inconsistencies.

7 Conclusions

In this paper we continued the analysis of the complexity of reasoning in the Bayesian DL $\mathcal{BEL}$, and considered tasks associated to conjunctive queries. Specifically, we have studied the complexity of deciding probabilistic entailment of a Boolean CQ, and of verifying that a tuple of answers to a CQ are those with the highest probability of being entailed. Dually, we consider also the problem of finding the most likely contexts that entail a Boolean query. All these complexity results are summarized in Table 2.

As it can be seen from the table, if one considers only the purely logical components of the problem (ontology complexity), then reasoning is tractable, which is consistent with the complexity of reasoning in classical $\mathcal{EL}$. The network complexity is also typically the same as performing standard probabilistic inferences over a BN. However, the complexity tends to increase if we combine these factors and consider also the query. Unfortunately, to the best of our efforts, we were unable to close all the gaps in the complexity results. Our conjecture is that the KB and the combined complexity coincide in all the problems considered here;

in particular, we expect all the problems described in Table 2, with the exception of top-k worlds, to be (co)NP$^{\text{PP}}$-complete w.r.t. these two complexity measures.

The algorithm for deciding query entailment through the computation of $\mathsf{con}(\mathsf{q})$ provides a tight upper bound for this problem w.r.t. network complexity. However, it would be impractical to implement as it iterates over all possible sub-ontologies. Arguably, techniques such as weighted model counting [14] would lead towards more practical algorithms for this problem. We will explore the possibility of extending the Bayesian ontology reasoner BORN with an efficient query entailment service using such techniques.

The proof of hardness for top-k query answering w.r.t. combined complexity uses a very simple query which is in fact acyclic. Thus, contrary to classical $\mathcal{EL}$ [5], restricting to acyclic queries does not suffice for reducing the complexity of reasoning. On the other hand, for simple instance queries the combined complexity should not be higher than the network complexity. This claim can be shown by adapting the proof structures from [10] to the completion-based algorithm for $\mathcal{ELO}$ as pointed in [7]. It would be interesting to find other meaningful restrictions that reduce the complexity of these reasoning tasks.

One important open issue is the use of partial information in our reasoning problems, through conditioning. For example, one could be interested in finding the context κ with the highest probability of occurring, given that a query q holds. Notice that this problem is different from finding the most likely context since in this case, we do not require that $\mathcal{K}_\kappa$ entails the query q.

Another future direction id to extend the framework to consider also temporal queries over dynamic ontologies in which the probabilistic knowledge evolves over time as described in [13].

Most of the notions and ideas presented here are independent of the logical formalism used. Indeed, although the specific complexity bounds found are specific to the properties of the DL $\mathcal{EL}$, the reasoning algorithms presented usually require only classical query entailment tests, and hence can be adapted to other ontological languages where these tests are decidable, without major trouble.

References

1. Abiteboul, S., Senellart, P.: Querying and updating probabilistic information in XML. In: Proc. of EDBT'06. LNCS, vol. 3896. Springer Verlag (2006)
2. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proc. of IJCAI'05. Morgan Kaufmann Publishers (2005)
3. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2nd edn. (2007)
4. Beigel, R., Nick, R., Spielman, D.A.: PP Is Closed under Intersection. J. of Computer and System Sciences 50(2) (1995)
5. Bienvenu, M., Ortiz, M., Šimkus, M., Xiao, G.: Tractable Queries for lightweight Description Logics. In: Proc. of IJCAI'13. AAAI (2013)
6. Brandt, S.: Polynomial Time Reasoning in a Description Logic with Existential Restrictions, GCI Axioms, and—What Else? In: Proc. of ECAI'04. vol. 110. IOS Press (2004)

7. Ceylan, İ.İ.: Query Answering in Bayesian Description Logics. In: Proc. of DL'15. CEUR Workshop Proceedings, vol. 1350. CEUR-WS (2015)
8. Ceylan, İ.İ., Mendez, J., Peñaloza, R.: The Bayesian Ontology Reasoner is BORN! In: Proc. of ORE'15. CEUR Workshop Proceedings, vol. 1387. CEUR-WS (2015)
9. Ceylan, İ.İ., Peñaloza, R.: Bayesian Description Logics. In: Proc. of DL'14. CEUR Workshop Proceedings, vol. 1193. CEUR-WS (2014)
10. Ceylan, İ.İ., Peñaloza, R.: Reasoning in the Description Logic $\mathcal{BEL}$ using Bayesian Networks. In: Proc. of StarAI'14. AAAI Workshops, vol. WS-14-13. AAAI (2014)
11. Ceylan, İ.İ., Peñaloza, R.: The Bayesian Description Logic $\mathcal{BEL}$. In: Proc. of IJCAR'14. LNCS, vol. 8562. Springer Verlag (2014)
12. Ceylan, İ.İ., Peñaloza, R.: Tight Complexity Bounds for Reasoning in the Description Logic $\mathcal{BEL}$. In: Proc. of JELIA'14. LNCS, vol. 8761. Springer Verlag (2014)
13. Ceylan, İ.İ., Peñaloza, R.: Dynamic Bayesian Ontology Languages. CoRR abs/1506.08030 (2015)
14. Chavira, M., Darwiche, A.: On Probabilistic Inference by Weighted Model Counting. Artificial Intelligence 172(6-7), 772–799 (2008)
15. Dalvi, N., Suciu, D.: Efficient Query Evaluation on Probabilistic Databases. VLDB Journal 16(4) (2007)
16. D'Amato, C., Fanizzi, N., Lukasiewicz, T.: Tractable Reasoning with Bayesian Description Logics. In: Proc. of SUM'08. LNCS, vol. 5291. Springer Verlag (2008)
17. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press (2009)
18. Fuhr, N., Rölleke, T.: A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems. ACM TOIS'97 15(1) (1997)
19. Gottlob, G., Lukasiewicz, T., Martinez, M.V., Simari, G.I.: Query answering under probabilistic uncertainty in datalog +/- ontologies. Ann. Math. AI 69(1) (2013)
20. Grädel, E., Gurevich, Y., Hirsch, C.: The Complexity of Query Reliability. In: Proc. ACM SIGACT-SIGMOD-SIGART'98 (1998)
21. Huang, H., Liu, C.: Query evaluation on probabilistic RDF databases. In: WISE09,. LNCS, vol. 5802. Springer Verlag (2009)
22. Hung, E., Getoor, L., Subrahmanian, V.S.: PXML: A probabilistic semistructured data model and algebra. In: Proc. ICDE'03 (2003)
23. Joseph, H.: An Analysis of First - Order Logics of Probability. In: Proc. of IJCAI'89. Morgan Kaufmann Publishers (1989)
24. Lukasiewicz, T., Straccia, U.: Managing Uncertainty and Vagueness in Description Logics for the Semantic Web. Web Semantics: Science, Services and Agents on the World Wide Web 6(4), 291–308 (Nov 2008)
25. Lutz, C., Schröder, L.: Probabilistic Description Logics for Subjective Uncertainty. In: Proc. of KR'10. AAAI (2010)
26. Lutz, C., Toman, D., Wolter, F.: Conjunctive Query Answering in the Description Logic $\mathcal{EL}$ Using a Relational Database System. In: Proc. of IJCAI'09. AAAI (2009)
27. Rosati, R.: On Conjunctive Query Answering in EL. In: Proc. of DL'07. CEUR Workshop Proceedings, vol. 250. CEUR-WS (2007)
28. Roth, D.: On the Hardness of Approximate Reasoning. Artificial Intelligence 82(1-2), 273–302 (1996)
29. Schulz, S., Suntisrivaraporn, B., Baader, F., Boeker, M.: SNOMED reaching its adolescence: Ontologists' and logicians' health check. International Journal of Medical Informatics 78(Supplement 1), 86–94 (2009)
30. Toda, S.: On the Computational Power of PP and +P. In: Proc. of SFCS'89. pp. 514–519. IEEE (1989)