Using Ontologies to Query Probabilistic Numerical Data (Extended Abstract)*

Franz Baader, Patrick Koopmann, and Anni-Yasmin Turhan

Theoretical Computer Science, TU Dresden

Abstract. We consider ontology-based query answering in a setting where some of the data are numerical and of a probabilistic nature, such as data obtained from uncertain sensor readings. The uncertainty for such numerical values can be more precisely represented by continuous probability distributions than by discrete probabilities for numerical facts concerning exact values. For this reason, we extend existing approaches using discrete probability distributions over facts by continuous probability distributions over numerical values. We determine the exact (data and combined) complexity of query answering in extensions of the well-known description logics $\mathcal{EL}$ and $\mathcal{ALC}$ with numerical comparison operators in this probabilistic setting.

1 Introduction

Ontology-based query answering (OBQA) has recently attracted considerable attention since it dispenses with the closed world assumption of classical query answering in databases and thus can deal with incomplete data. In addition, background information stated in an appropriate ontology can be used to deduce more answers. OBQA is usually investigated in a setting where queries are (unions of) conjunctive queries and ontologies are expressed using an appropriate Description Logic (DL). Depending on the expressiveness of the DL, the complexity of query answering may vary considerably, starting with data complexity (i.e., complexity measured in the size of the data only) of AC^0 for members of the DL-Lite family [10, 2] to P for DLs of the $\mathcal{EL}$ family [30], all the way up to intractable data complexity for expressive DLs such as $\mathcal{ALC}$ and beyond [19].

In many application scenarios for OBQA, however, querying just symbolic data is not sufficient. One also wants to be able to query numerical data. For example, in a health or fitness monitoring application, one may want to use concepts from a medical ontology such as SNOMED CT [18] or Galen [31] to express information about the health status of a patient, but also needs to store and refer to numerical values such as the blood pressure or heart rate of this patient. As an example, let us consider hypertension management using a smartphone app [25]. What constitutes dangerously high blood pressure (HBP) depends on

* Supported by the DFG within the collaborative research center SFB 912 (HAEC) and the research unit FOR 1513 (HYBRIS).

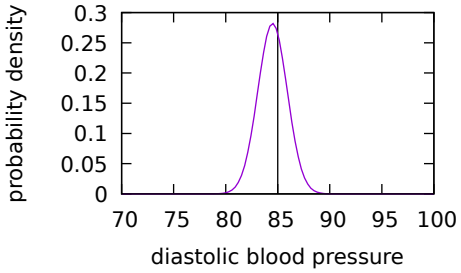

Fig. 1. Measured blood pressure as normal distribution.

the measured values of the diastolic pressure, but also on other factors. For example, if a patient suffers from diabetes, a diastolic blood pressure above 85 may already be classified as too high, whereas under normal circumstances it is only considered to be too high above 90. This could, for example, be modelled as follows by an ontology:

$$\exists \mathsf{diastolicBloodPressure}.{>_{90}} \sqsubseteq \mathsf{PatientWithHBP} \quad (1)$$

$$\exists \mathsf{finding}.\mathsf{Diabetes} \sqcap \exists \mathsf{diastolicBloodPressure}.{>_{85}} \sqsubseteq \mathsf{PatientWithHBP} \quad (2)$$

Note that we have used a DL with concrete domains [6] to refer to numerical values and predicates on these values within concepts. While there has been quite some work on traditional reasoning (satisfiability, subsumption, instance) in DLs with concrete domains [27], there is scant work on OBQA for such DLs. To the best of our knowledge, the only work in this direction considers concrete domain extensions of members of the DL-Lite family [3, 33, 4, 21]. In contrast, we consider concrete domain extensions of $\mathcal{EL}$ and $\mathcal{ALC}$ and determine the (combined and data) complexity of query answering.

However, the main difference to previous work is that we do not assume the numerical values in the data to be exact. In fact, a value of 84.5 for the diastolic pressure given by a blood pressure sensor does not really mean that the pressure is precisely 84.5, but rather that it is around 84.5. The actual value follows a probability distribution—for example a normal distribution with expected value 84.5 and a variance of 2 as shown in Figure 1—which is determined by the measured value and some known variance that is a characteristic of the employed sensor. We can represent this in the knowledge base for example as follows:

$$\mathsf{finding}(\mathsf{otto}, \mathsf{f1}) \qquad \mathsf{Diabetes}(\mathsf{f1}) \qquad \mathsf{diastolicBloodPressure}(\mathsf{otto}) \sim \mathsf{norm}(84.5, 2)$$

From this information, we can derive that the minimal probability for the patient Otto to have high blood pressure is slightly above 36%, which might be enough to issue a warning. In contrast, if instead of using a probability distribution we had asserted 84.5 as the exact value for Otto's diastolic blood pressure, we could not have inferred that Otto is in any danger.

Continuous probability distributions as used in this example also emerge in other potential applications of OBQA such as in robotics [36], tracking of object positions in video analytics [37], and mobile applications using probabilistic sensor data [16], to name a few. The interest in continuous probability distributions is also reflected in the development of database systems that support these [35].

In addition to using continuous probability distributions for sensor values, we also consider discrete probability distributions for facts. For example, it might be that the finding f1 for Otto is diabetes only with a certain probability. While OBQA for probabilistic data with discrete probability distributions has been considered before for DL-Lite and $\mathcal{EL}$ without concrete domains [15, 23, 13], as well as for datalog [12], OBQA for probabilistic data with both discrete and continuous probability distributions is investigated here for the first time. A rather expressive combination we consider is the DL $\mathcal{ALC}$ extended with a concrete domain in which real numbers can be compared using the (binary) predicates $>$ and $=$. A less expressive combination we consider is the DL $\mathcal{EL}$ extended with a concrete domain in which real numbers can be compared with a fixed number using the (unary) predicates $>_n$ for $n \in \mathbb{R}$. Since OBQA for classical knowledge bases (i.e., without probabilities) in these two DLs has not been investigated before, we first determine their (data and combined) complexity of query answering. When considering probabilistic KBs with continuous probability distributions (modelled as real-valued functions), the resulting probabilities may be numbers without a finite representation. To overcome this problem, we define probabilistic query entailment with respect to a given precision parameter. To allow a reasonable complexity analysis, we define a set of feasibility conditions for probability distributions, based on the complexity theory of real functions [24], which capture most typical probability distributions that appear in practical applications. For probabilistic KBs that satisfy these conditions, we give tight bounds on the complexity of probabilistic query answering w.r.t data and combined complexity for all considered DLs.

A more detailed version of this paper is published in in [7]. Proofs of all results can be found in [8].

2 Description Logics with Numerical Domains

We focus on the description logics $\mathcal{ALC}(\mathsf{R})$ and $\mathcal{EL}(\mathsf{R}_>)$, which extend the description logics $\mathcal{ALC}$ and $\mathcal{EL}$ with a concrete domain over the real numbers as in [6]. We assume notions of $\mathcal{ALC}/\mathcal{EL}$ TBox, ABox, and knowledge base, as well as entailment, as in [34, 5]. $\mathcal{ALC}(\mathsf{R})$ and $\mathcal{EL}(\mathsf{R}_>)$ allow to assign real numbers to individuals using concrete features, which are interpreted as partial functions. This is done using ABox statements of the form $g(a, r)$ in the ABox, where a is an individual, g a concrete feature and $r \in \mathbb{R}$ a real number.

$\mathcal{EL}(\mathsf{R}_>)$ further extends $\mathcal{EL}$ by concepts of the form $\exists g.>_r$, where g is is a concrete feature and $r \in \mathbb{R}$, to describe objects whose concrete feature g has a value larger than r. $\mathcal{ALC}(\mathsf{R})$ allows to compare different concrete features reachable via paths along *abstract features*, which are just functional roles. Namely,

	$\mathcal{EL}(\mathsf{R}_>)$		$\mathcal{ALC}(\mathsf{R})$	
	AQs	UCQs	AQs	UCQs
Data complexity	P	P	coNP	coNP
Combined Complexity	P	NP	ExpTime	ExpTime

Table 1. Complexity of classical query entailment.

$\mathcal{ALC}(\mathsf{R})$ extends $\mathcal{ALC}$ with concepts of the form $\exists g.\oplus_r$ and $\exists(u_1, u_2).\oplus$, where g is a concrete feature, $\oplus \in \{<, =, >\}$, $r \in \mathbb{R}$, and u_1,u_2 are *paths* of the form $s_1 \dots s_n g$, where s_i are abstract features and g is a concrete feature. For example, in $\mathcal{ALC}(\mathsf{R})$, we can give a characterisation of patients with HBP such as the following:

$$\exists(\mathsf{diastolicBP}, \mathsf{belongsToAgeGroup\ maxDiastolicBP}).{>} \sqsubseteq \mathsf{PatientWithHBP}.$$

This definition assumes patients to be related to an age group via the abstract feature belongsToAgeGroup, which has an assigned maximal diastolic blood pressure. If the patient has a diastolic blood pressure that is above this value, he is a patient with HBP. The more expressive concrete domain R used by $\mathcal{ALC}(\mathsf{R})$ admits ExpTime reasoning, while even small extensions lead to undecidability [26]. In contrast, the concrete domain $\mathsf{R}_>$ used in $\mathcal{EL}(\mathsf{R}_>)$ is a *convex* domain, which allows to perform standard reasoning tasks in polynomial time [5]. For more details on $\mathcal{EL}(\mathsf{R}_>)$ and $\mathcal{ALC}(\mathsf{R})$, we refer to to [6, 26, 5] or the extended version of this paper. We assume both ABoxes and TBoxes to contain only a finite set of axioms and ABoxes to contain no complex concepts, and we do not impose a unique name assumption.

As main reasoning task, we consider entailment of *atomic queries* (AQs), *conjunctive queries* (CQs) and *unions of conjunctive queries* (UCQs), defined as for description logics without concrete domains [19]. We allow for constants and complex concepts in the query, but for concrete features only inside concepts. This means that our queries can only express relations between concrete features that can be captured by a concept in our language. For example, the FOL formula

$$\exists y_1, y_2, z_1, z_2 : s_1(x, y_1) \wedge g_1(y_1, z_1) \wedge s_2(x, y_2) \wedge g_2(y_2, z_2) \wedge z_1 < z_2.$$

can be captured the query $\exists(s_1 g_1, s_2 g_2).{<}(x)$, but only given s_1, s_2 are abstract features, g_1, g_2 concrete features, and $<$ is a predicate of the concrete domain. As a foundation for our results on probabilistic knowledge bases, we analyse in the technical report the data and combined complexities of query entailment for the logics considered. Here, we assume all numbers to be represented in binary. Our complexity analysis only concerns knowledge bases that have a finite representation, which by this assumption are those in which each number can be represented with a finite number of bits.

An overview of the complexities is shown in Table 1. The results are based on and match the corresponding results on DLs without concrete domains [34, 32, 11, 28]. Since the corresponding lower bounds are the same for CQs as for UCQs, we do not include the results for CQs in the table.

3 Probabilistic Knowledge Bases with Continuous Probability Distributions

We want to represent both, discrete probabilities of assertions and continuous probability distributions of values of concrete features. As we can simply assign a probability of 1 to assertions that are certain, there is no need to handle certain assertions separately. A *discrete probability assertion* assigns a minimal probability to a classical assertion. This corresponds to the approach taken by *tuple-independent probabilistic database systems* [14], where probabilities are assigned to database and to *ipABoxes* introduced in [23]. For example, the fact that "Otto has a finding that is Diabetes with a probability of at least 0.7" is expressed by the two assertions $\mathsf{finding}(\mathsf{otto},\mathsf{f1}) : 1$ and $\mathsf{Diabetes}(\mathsf{f1}) : 0.7$.

Note that discrete probability assertions state a lower bound on the probability, rather than the actual probability, and that statistical independence is only assumed on this lower bound. This way, it is consistent to have the assertions $A(a) : 0.5$, $B(a) : 0.5$ together with the axiom $A \sqsubseteq B$ in the knowledge base. Under our semantics, the probability of $B(a)$ is then higher than 0.5, since this assertion can be entailed due to two different, statistically independent statements in the ABox. Namely, we would infer that the probability of $B(a)$ is at least 0.75 (compare also with [23]).

While for symbolic facts, assigning discrete probabilities is sufficient, for numerical values this is not necessarily the case. For example, if the blood pressure of a patient follows a continuous probability distribution, the probability of it to have any specific value is 0. For this reason, in a *continuous probability assertion*, we connect the value of a concrete feature with a probability density function. This way, the fact that "the diastolic blood pressure of Otto follows a normal distribution with an expected value of 84.5 and a variance of 2" can be expressed by the assertion $\mathsf{diastolicBloodPressure}(\mathsf{otto}) \sim \mathsf{norm}(84.5, 2)$. In addition to a concrete domain $\mathcal{D}$, the DLs introduced in this section are parametrised with a set $\mathcal{P}$ of *probability density functions* (*pdfs*), i.e, Lebesgue-integrable functions $f : A \to \mathbb{R}^+$, with $A \subseteq \mathbb{R}$ being Lebesgue-measurable, such that $\int_A f(x)\,\mathrm{d}x{=}1$ [1].

Example 1. As a typical set of probability density functions [1], we define the set $\mathcal{P}_{ex}$ that contains the following functions, which are parametrised with the numerical constants $\mu, \omega, \lambda, a, b \in \mathbb{Q}$, with $\lambda > 0$ and $a > b$:

normal distribution with mean μ and variance ω:
$\mathsf{norm}(\mu,\omega) : \mathbb{R} \to \mathbb{R}^+,\ x \mapsto \frac{1}{\sqrt{2\pi\omega}} e^{-(x-\mu)^2/2\omega}$,
exponential distribution with mean λ:
$\mathsf{exp}(\lambda) : \mathbb{R}^+ \to \mathbb{R}^+, x \mapsto \lambda e^{-\lambda x}$,
uniform distribution between a and b:
$\mathsf{uniform}(a,b) : [a,b] \to \mathbb{R}^+, x \mapsto \frac{1}{b-a}$.

Next, we define probabilistic KBs, which consist of a classical TBox and a set of probability assertions.

Definition 1. *Let $\mathcal{L} \in \{\mathcal{EL}(\mathsf{R}_>), \mathcal{ALC}(\mathsf{R})\}$ and $\mathcal{P}$ be a set of pdfs. A* probabilistic $\mathcal{L}_\mathcal{P}$ ABox *is a finite set of expressions of the form $\alpha : p$ and $g(a) \sim f$, where α is*

an $\mathcal{L}$ assertion, $p \in [0,1] \cap \mathbb{D}$,[1] *$g \in N_{cF}$, $a \in N_i$, and $f \in \mathcal{P}$. A* probabilistic $\mathcal{L}_\mathcal{P}$ KB *is a tuple $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, where $\mathcal{T}$ is an $\mathcal{L}$ TBox and $\mathcal{A}$ is a probabilistic $\mathcal{L}_\mathcal{P}$ ABox. If $\mathcal{P} = \emptyset$, $\mathcal{K}$ and $\mathcal{A}$ are called* discrete*, and if $\mathcal{P} \neq \emptyset$, they are called* continuous*.*

3.1 Semantics of Probabilistic Knowledge Bases

As typical for probabilistic DLs and databases, we define the semantics using a *possible worlds semantics*. In probabilistic systems that only use discrete probabilities, the possible world semantics can be defined based on finite sets of non-probabilistic data sets, the possible worlds, each of which is assigned a probability [14, 23, 29]. The probability that a query q is entailed then corresponds to the sum of the probabilities of the possible worlds that entail q. If continuous probability distributions are used, this approach is insufficient. For example, if the KB contains the assertion $\mathsf{diastolicBP}(p) \sim \mathsf{norm}(84.5, 2)$, the probability of $\mathsf{diastolicBP}(p, x)$ should be 0 for every $x \in \mathbb{R}$. Therefore, we cannot obtain the probability of $\mathsf{diastolicBP}(p) > 85$ by just adding the probabilities of the possible worlds that entail $\mathsf{diastolicBP}(p, x)$ for some $x > 85$. To overcome this problem, we assign probabilities to (possibly uncountable) *sets* of possible worlds, rather than to single possible worlds. Specifically, we define the semantics using continuous probability measure spaces [1]. A *measure space* is a tuple $M = (\Omega, \Sigma, \mu)$ with $\Sigma \subseteq 2^\Omega$ and $\mu : \Sigma \to \mathbb{R}$ such that

1. $\Omega \in \Sigma$ and Σ is closed under complementation, countable unions and countable intersections,
2. $\mu(\emptyset) = 0$, and
3. $\mu(\bigcup_{E \in \Sigma'}) = \sum_{E \in \Sigma'} \mu(f)$ for every countable set $\Sigma' \subseteq \Sigma$ of pair-wise disjoint sets.

If additionally $\mu(\Omega) = 1$, M is a *probability measure space*.

We define a probability measure space $M_\mathcal{A} = (\Omega_\mathcal{A}, \Sigma_\mathcal{A}, \mu_\mathcal{A})$ that captures the relevant probabilities in a probabilistic ABox $\mathcal{A}$, similar to how it is done in [23] for discrete probabilistic ABoxes. For this, we introduce the three components $\Omega_\mathcal{A}$, $\Sigma_\mathcal{A}$ and $\mu_\mathcal{A}$ one after another. For simplicity, we assume all pdfs $f : A \to \mathbb{R} \in \mathcal{P}$ to be extended to the full real line by setting $f(x) = 0$ for all $x \in \mathbb{R} \setminus A$.

Given a probabilistic ABox $\mathcal{A}$, the set of *possible worlds for* $\mathcal{A}$, in symbols $\Omega_\mathcal{A}$, consists of all classical ABoxes w such that for every $g(a) \sim f \in \mathcal{A}$, w contains $g(a, x)$ for some $x \in \mathbb{R}$, and for every axiom $\alpha \in w$, either $\alpha : p \in \mathcal{A}$, or α is of the form $g(a, x)$ and $g(a) \sim f \in \mathcal{A}$. For $w \in \Omega_\mathcal{A}$, we write $w \models g(a) \oplus x$, $x \in \mathbb{R}$, $\oplus \in \{<, \leq, =, \geq, >\}$, iff $w \models g(a, y)$ and $y \oplus x$. We write $w \models g(a) \oplus h(b)$ iff $w \models g(a, y), h(b, z)$ and $y \oplus z$. We abbreviate $w \models g(a) \geq x, g(a) \leq y$ by $w \models g(a) \in [x, y]$. The *event space over* $\Omega_\mathcal{A}$, in symbols $\Sigma_\mathcal{A}$, is now the smallest subset $\Sigma_\mathcal{A} \subseteq 2^{\Omega_\mathcal{A}}$ that satisfies the following conditions:

1. $\Omega_\mathcal{A} \in \Sigma_\mathcal{A}$,

[1] Here, the set $\mathbb{D} \subseteq \mathbb{R}$ denotes the *dyadic rationals*, that is, the set of all real numbers that have a finite number of bits after the binary point.

2. for every $\alpha : p \in \mathcal{A}$, $\{w \in \Omega_\mathcal{A} \mid \alpha \in w\} \in \Sigma_\mathcal{A}$,
3. for every $g(a) \sim f \in \mathcal{A}$, $x \in \mathbb{R}$, $\{w \in \Omega_\mathcal{A} \mid w \models g(a) < x\} \in \Sigma_\mathcal{A}$,
4. for every $g_1(a_1) \sim f_1$, $g_2(b) \sim f_2 \in \mathcal{A}$, $\{w \in \Omega_\mathcal{A} \mid w \models g_1(a) < g_2(b)\} \in \Sigma_\mathcal{A}$, and
5. $\Sigma_\mathcal{A}$ is closed under complementation, countable unions and countable intersections.

The conditions ensure that for every query q and TBox $\mathcal{T}$, the set of possible worlds w such that $(\mathcal{T}, w) \models q$ is included in $\Sigma_\mathcal{A}$. To complete the definition of the measure space, we now assign probabilities to these sets via the measure function $\mu_\mathcal{A}$. This function has to respect the probabilities expressed by the discrete and continuous probability assertions in $\mathcal{A}$, as well as the assumption that these probabilities are statistically independent. We define $\mu_\mathcal{A}$ explicitly for sets of possible worlds that are selected by the assertions in them, and by upper bounds on the concrete features occurring in continuous probability assertions. By additionally requiring that Condition 3 in the definition of measure spaces is satisfied for $\mu_\mathcal{A}$, this is sufficient to fix the probability for any set in $\Sigma_\mathcal{A}$.

Given a probabilistic ABox $\mathcal{A}$, we denote by $\mathsf{cl\text{-}ass}(\mathcal{A}) = \{\alpha \mid \alpha : p \in \mathcal{A}\}$ the classical assertions occurring in $\mathcal{A}$. A *bound set for* $\mathcal{A}$ is a set $\mathbf{B}$ of inequations of the form $g(a) < x$, $x \in \mathbb{R}$, where $g(a) \sim f \in \mathcal{A}$ and every concrete feature $g(a)$ occurs at most once in $\mathbf{B}$. Given a set $\mathcal{E} \subseteq \mathsf{cl\text{-}ass}(\mathcal{A})$ of assertions from $\mathcal{A}$ and a bound set $\mathbf{B}$ for $\mathcal{A}$, we define the corresponding set $\Omega_\mathcal{A}^{\mathcal{E},\mathbf{B}}$ of possible worlds in $\Omega_\mathcal{A}$ as

$$\Omega_\mathcal{A}^{\mathcal{E},\mathbf{B}} = \{w \in \Omega_\mathcal{A} \mid w \cap \mathsf{cl\text{-}ass}(\mathcal{A}) = \mathcal{E}, w \models \mathbf{B}\}.$$

The probability measure space for $\mathcal{A}$ is now the probability measure space $M_\mathcal{A} = (\Omega_\mathcal{A}, \Sigma_\mathcal{A}, \mu_\mathcal{A})$, such that for every $\mathcal{E} \subseteq \mathsf{cl\text{-}ass}(\mathcal{A})$ and every bound set $\mathbf{B}$ for $\mathcal{A}$,

$$\mu_\mathcal{A}(\Omega_\mathcal{A}^{\mathcal{E},\mathbf{B}}) = \prod_{\substack{\alpha:p\in\mathcal{A}\\ \alpha\in\mathcal{E}}} p \cdot \prod_{\substack{\alpha:p\in\mathcal{A}\\ \alpha\notin\mathcal{E}}} (1-p) \cdot \prod_{\substack{g(a)\sim f\in\mathcal{A}\\ g(a)<x\in\mathbf{B}}} \int_{-\infty}^{x} f(y)\,\mathrm{d}y.$$

As shown in the extended version of the paper, this definition uniquely determines $\mu_\mathcal{A}(W)$ for all $W \in \Sigma_\mathcal{A}$, including for sets such as $W = \{w \in \Omega_\mathcal{A} \mid w \models g_1(a) < g_2(b)\}$. The above product is a generalisation of the corresponding definition in [23] for discrete probabilistic KBs, where in addition to discrete probabilities, we take into consideration the continuous probability distribution of the concrete features in $\mathcal{A}$. Recall that if a concrete feature $g(a)$ follows the pdf f, the integral $\int_{-\infty}^{x} f(y)\,\mathrm{d}y$ gives us the probability that $g(a) < x$.

Since we have now finished the formal definition of the semantics of probabilistic ABoxes, we can now define the central reasoning task studied in this paper. As in Section 2, we concentrate on probabilistic query entailment rather than on probabilistic query answering. The latter is a ranked search problem that can be polynomially reduced to probabilistic query entailment as in [23]. Based on the measure space $M_\mathcal{A}$, we define the *probability of a Boolean query* q in a probabilistic KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ as $P_\mathcal{K}(q) = \mu_\mathcal{A}(\{w \in \Omega_\mathcal{A} \mid (\mathcal{T}, w) \models q\})$. Note that due to the open-world assumption, strictly speaking, $P_\mathcal{K}(q)$ corresponds to

a lower bound on the probability of q, since additional facts may increase the value of $P_{\mathcal{K}}(q)$.

Different to [23] and classical approaches in probabilistic query answering, because $\mathcal{P}$ contains real functions, $P_{\mathcal{K}}(q)$ is in general a real number, and as such not finitely representable. In practice, it is typical and usually sufficient to compute approximations of real numbers. To capture this adequately, we take the required precision of the probability $P_{\mathcal{K}}(q)$ as additional input to the probabilistic query entailment problem. For a real number $x \in \mathbb{R}$ and $n \in \mathbb{N}$, we use the notation $\langle x \rangle_n$ to refer to an *n-bit approximation of x*, that is, a real number such that $|\langle x \rangle_n - x| < 2^{-n}$. Note that, while we do not enforce it, generally n bits after the binary point are sufficient to identify $\langle x \rangle_n$. We can now state the main reasoning problem studied in this paper.

Definition 2. *The* probabilistic query entailment problem *is the problem of computing, given a probabilistic KB $\mathcal{K}$, a Boolean query q and a natural number n in unary encoding, a number x s.t. $x = \langle P_{\mathcal{K}}(q) \rangle_n$.*

Since the precision parameter n determines the size of the result, we assume it in unary encoding. If we would represent it in binary, it would already take exponential time just to write the result down.

4 Feasibility Conditions for PDFs

Up to now, we did not put any restrictions on the set $\mathcal{P}$ of pdfs, so that a given set $\mathcal{P}$ could easily render probabilistic query entailment uncomputable. In this section, we define a set of feasibility conditions on pdfs that ensure that probabilistic query entailment is not computationally harder than when no continuous probability distributions are used. We know from results in probabilistic databases [14], that query entailment over probabilistic data is $\#\cdot$P-hard. Note that integration of pdfs over bounded intervals can be reduced to probabilistic query answering. Namely, if $g(a) \sim f \in \mathcal{A}$, we have $P_{(\emptyset,\mathcal{A})}((\exists g.{>}_r)(a)) = \int_r^\infty f(x)\,\mathrm{d}y$ for all $r \in \mathbb{R}$. Our feasibility conditions ensure that the complexity of approximating integrals does not dominate the overall complexity of probabilistic query entailment.

We first recall some notions from the complexity theory of real functions by Ker-I Ko [24], which identifies computability of real numbers $x \in \mathbb{R}$ and functions $f : A \to \mathbb{R}$, $A \subseteq \mathbb{R}$, with the computability of n-bit approximations $\langle x \rangle_n$ and $\langle f(x) \rangle_n$, where n is given in unary encoding. Since real function arguments have no finite representation in general, computable real functions are modelled as function oracle Turing machines $T^{\phi(x)}$, where the oracle $\phi(x)$ represents the function argument x and can be queried for n-bit approximations $\langle x \rangle_n$ in time linear in $c + n$, where c is the number of bits in x before the binary point. Given a precision n in unary encoding on the input tape, $T^{\phi(x)}$ then writes a number $\langle f(x) \rangle_n$ on the output tape. This formalism leads to a natural definition of computability and complexity of real numbers and real functions. Namely, a real number $x \in \mathbb{R}$ is P-*computable* iff there is a polynomial time Turing machine

that computes a function $\phi : \mathbb{N} \mapsto \mathbb{D}$ s.t. $\phi(n) = \langle x \rangle_n$. A function $f : A \to \mathbb{R}$, $A \subseteq \mathbb{R}$, is P-computable iff there is a function oracle Turing machine $T^{\phi(x)}$ as above that computes for all $x \in A$ a function $\psi : \mathbb{N} \mapsto \mathbb{D}$ with $\psi(n) = \langle f(x) \rangle_n$ in time polynomial in n and the number of bits in x before the binary point.

An important property of P-computable functions f that we use in the next section is that they have a monotone and polynomial *modulus of continuity* (*modulus*), that is, a monotone, polynomial function $\omega_f : \mathbb{N} \to \mathbb{N}$ s.t. for all $n \in \mathbb{N}$ and $x, y \in [2^{-n}, 2^n]$, $|x - y| < 2^{-\omega_f(n)}$ implies $|f(x) - f(y)| < 2^{-n}$ [22, 24, Chapter 3].

Approximating integrals $\int_0^1 f(x)\,\mathrm{d}x$ of P-computable functions $f : [0, 1] \to \mathbb{R}$ is $\#\cdot$P-complete [24, Chapter 5]. To be able to integrate over unbounded integrals in $\#\cdot$P, we introduce an additional condition.

Definition 3. *A probability density function f is* $\#\cdot$P-admissible *iff it satisfies the following conditions:*

1. *f is* P*-computable, and*
2. *there is a monotone polynomial function $\delta_f : \mathbb{N} \to \mathbb{N}$ such that for all $n \in \mathbb{N}$:*

$$1 - \int_{-2^{\delta_f(n)}}^{2^{\delta_f(n)}} f(x)\,\mathrm{d}x < 2^{-n}.$$

Condition 2 allows us to reduce integration over *un*bounded integrals to integration over bounded integrals: to obtain a precision of n bits, it is sufficient to integrate inside the interval $[-2^{\delta_f(n)}, 2^{\delta_f(n)}]$. Note that as a consequence of Condition 1, there is also a polynomial $\rho_f : \mathbb{N} \to \mathbb{N}$ s.t. for all $x \in [-2^{\delta_f(n)}, 2^{\delta_f(n)}]$, $f(x) < 2^{\rho_f(n)}$. Otherwise, approximations of $f(x)$ would require a number of bits that is not polynomially bounded by the number of bits in x before the binary point, and could thus not be computed in polynomial time. We call δ_f and ρ_f in the definition above respectively *bounding function* and *range function* of f. In the following, we assume that for any set $\mathcal{P}$ of $\#\cdot$P-admissible pdfs, their moduli, bounding functions and range functions are known.

The above properties are general enough to be satisfied by most common pdfs. Specifically, we have the following lemma for the set $\mathcal{P}_{\mathsf{ex}}$ defined in Example 1:

Lemma 1. *Every function in $\mathcal{P}_{\mathsf{ex}}$ is* $\#\cdot$P*-admissible.*

5 Complexity of Probabilistic Query Answering

We study the complexity of probabilistic query answering for KBs with $\#\cdot$P-admissible pdfs. As often in probabilistic reasoning, counting complexity classes play a central role in our study. However, strictly speaking, these are defined for computation problems for *natural numbers*. To get a characterisation for probabilistic query answering, which computes real numbers, we consider corresponding counting problems. Their solutions are obtained by, intuitively, shifting the binary point of the query probability to the right to obtain a natural number. We first recall counting complexity classes following [20].

Definition 4. *Let $\mathcal{C}$ be a class of decision problems. Then, $\#\cdot\mathcal{C}$ describes the class of functions $f : A \to \mathbb{N}$ such that*

$$f(x) = \left\|\left\{y \mid R(x,y) \wedge |y| < p(|x|)\right\}\right\|$$

for some $\mathcal{C}$-decidable relation R and polynomial function p.

Relevant for us are the counting complexity classes $\#\cdot$P, $\#\cdot$NP and $\#\cdot$coNP. The class $\#\cdot$P is also known as #P. The following inclusions are known: $\#\cdot P \subseteq \#\cdot\text{NP} \subseteq \#\cdot\text{coNP} \subseteq \text{FPSpace}$ [20]. In order to characterise the complexity of probabilistic query answering using counting classes, we consider corresponding counting problems, inspired by [24, Chapter 5] and [14]. For a function $f : A \to \mathbb{D}$, we call $g : A \to \mathbb{N}$ a *corresponding counting problem* if $g(x) = 2^{p(x)} f(x)$ for all $x \in A$, where $p : A \to \mathbb{N}$ and p can be computed in unary in polynomial time.[2]

For discrete probabilistic KBs $(\mathcal{T}, \mathcal{A})$, we can define a counting problem corresponding to probabilistic query entailment that is solved by counting possible worlds in the probability measure space for $\mathcal{A}$. To obtain the probability of a query q, we count those possible worlds $w \in \Omega_{\mathcal{A}}$ for which $(\mathcal{T}, w) \models q$, and we count each possible world several times depending on its probability. If $\mathcal{C}$ is the respective complexity of classical query entailment, this counting problem can be easily established as being in $\#\cdot\mathcal{C}$.

For continuous probabilistic KBs, this approach cannot be directly employed, because here the set of possible worlds is uncountable. Namely, for every continuous probability assertion $g(a) \sim f \in \mathcal{A}$ and every real number $r \in \mathbb{R}$, there is a possible world $w \in \Omega_{\mathcal{A}}$ s.t. $g(a, r) \in w$. To overcome this, we define an *approximated measure space*, based on the precision parameter n, in which every assertion $g(a, r)$ uses at most a polynomial number of bits. As a result, the number of possible worlds becomes finite, and each possible world can be assigned a positive probability. We can now compute the approximated probability of a query by summing up the probabilities of each possible world that entails this query. As shown in more detail in [7, 8], by carefully taking into account the precision parameter n and properties of $\#\cdot$P-admissible pdfs, it is possible to define such an approximated measure space so that the overall approximation error on every query is bounded by 2^{-n}, and that the probability of each possible world can be computed in polynomial time. Using this, one can define a corresponding counting problem for probabilistic query answering that is in $\#\cdot\mathcal{C}$, where $\mathcal{C}$ is the respective complexity of classical query answering, and obtain the complexity upper bounds shown in Table 2. The ExpTime-bounds follow from the fact that the number of possible worlds in the approximated measure space is exponentially bounded.

Hardness for all complexities already holds for discrete probabilistic KBs, so that continuous, $\#\cdot$P-admissible probability distributions do not increase the complexity of probabilistic query answering. A general $\#\cdot$P-lower bound follows

[2] Note that the counting complexity classes considered here are all closed under this operation. To see this, consider f and g characterized by the relations R and R' s.t. $R' = \{(x, y\#z) \mid R(x, y), z \in \{0,1\}^*, |z| = p(x)\}$. Clearly, $g(x) = 2^{p(x)} f(x)$.

	$\mathcal{EL}(\mathsf{R}_{>})_{\mathcal{P}}$		$\mathcal{ALC}(\mathsf{R})_{\mathcal{P}}$	
	AQs	UCQs	AQs	UCQs
Data complexity	#·P	#·P	#·coNP	#·coNP
Combined Complexity	#·P	#·NP	ExpTime	ExpTime

Table 2. Complexities of counting problems corresponding to prob. query entailment.

from the corresponding complexity of probabilistic query entailment in probabilistic databases [14], while for the combined complexities in $\mathcal{ALC}(\mathsf{R})_{\mathcal{P}}$, the lower bound follows from the non-probabilistic case. For the remaining complexities, we provide matching lower bounds in the technical report using appropriate reductions. Specifically, we show #·NP-hardness under *subtractive reductions* for $\mathcal{EL}$ w.r.t. combined complexity, and #·coNP-hardness under *parsimonious reductions* for $\mathcal{ALC}$ w.r.t data complexity [17].

6 Conclusion

When numerical data are of an uncertain nature, such as data obtained by sensor readings or video tracking, they can often be more precisely represented using continuous probability distributions than using discrete distributions. While there is work on OBQA for discrete probabilistic KBs in DL-Lite and $\mathcal{EL}$ [23], this is the first work that considers KBs with concrete domains and continuous probability distributions. For our complexity analysis, we devised a set of feasibility conditions for probability distributions based on the complexity theory of real functions, which captures most typical distributions one might encounter in realistic applications. We show that under these conditions, continuous probability distributions do not increase the complexity of probabilistic query entailment. Using a similar technique as in [24, Chapter 5], our results can likely be extended to a wider class of probability distributions, where the requirement of P-computability is weakened to *polynomial approximability*.

For light-weight description logics, it is often possible to rewrite queries w.r.t the ontology, so that they can be answered directly by a corresponding database system. As there are probabilistic database systems like Orion 2.0 that support continuous probability distributions [35], query rewriting techniques for continuous probabilistic KBs could be employed in our setting as well. For more expressive DLs, a practical implementation could be based on a less fine-grained representation of measure spaces, for which relevant intervals for each concrete feature value are determined based on the concrete domain predicates in the TBox. Probabilities could then be computed using standard algorithms for numerical integration. It might also be worth investigating whether Monte-Carlo approximations can be used for practical implementations. However, as observed in [23], this might be hard to accomplish already for discrete probabilistic $\mathcal{EL}$ KBs. Another basis for practical implementations could be approximation techniques developed for other logical frameworks involving continuous probability distributions, such as the one presented in [9].

References

1. Adams, M.R., Guillemin, V.: Measure theory and probability. Springer (1996)
2. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The *DL-Lite* family and relations. J. Artif. Intell. Res. 36, 1–69 (2009)
3. Artale, A., Ryzhikov, V., Kontchakov, R.: *DL-Lite* with attributes and datatypes. In: Proc. ECAI'12. pp. 61–66. IOS Press (2012)
4. Baader, F., Borgwardt, S., Lippmann, M.: Query rewriting for *DL-Lite* with n-ary concrete domains (2017), to appear in Proc. IJCAI'17.
5. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proc. IJCAI'05. pp. 364–369. Professional Book Center (2005)
6. Baader, F., Hanschke, P.: A scheme for integrating concrete domains into concept languages. In: Proc. IJCAI'91. pp. 452–457 (1991)
7. Baader, F., Koopmann, P., Turhan, A.Y.: Using ontologies to query probabilistic numerical data. In: Proc. FroCoS'17. Springer (2017), to appear
8. Baader, F., Koopmann, P., Turhan, A.Y.: Using ontologies to query probabilistic numerical data (extended version). LTCS-Report 17-05, Chair for Automata Theory, Technische Universität Dresden, Germany (2017), see https://lat.inf.tu-dresden.de/research/reports.html
9. Belle, V., Van den Broeck, G., Passerini, A.: Hashing-based approximate probabilistic inference in hybrid domains: An abridged report. In: Proc. IJCAI'16. pp. 4115–4119 (2016)
10. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family. J. Autom. Reas. 39(3), 385–429 (2007)
11. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Data complexity of query answering in description logics. Artificial Intelligence 195, 335 – 360 (2013)
12. Ceylan, İ.İ., Lukasiewicz, T., Peñaloza, R.: Complexity results for probabilistic datalog±. In: Proc. ECAI'16. pp. 1414–1422. IOS Press (2016)
13. Ceylan, İ.İ., Peñaloza, R.: Probabilistic query answering in the Bayesian description logic $\mathcal{BEL}$. In: Proc. SUM'15. pp. 21–35. Springer (2015)
14. Dalvi, N., Suciu, D.: Management of probabilistic data: foundations and challenges. In: Proc. SIGMOD'07. pp. 1–12. ACM (2007)
15. D'Amato, C., Fanizzi, N., Lukasiewicz, T.: Tractable reasoning with Bayesian description logics. In: Proc. SUM'08. pp. 146–159. Springer (2008)
16. Dargie, W.: The role of probabilistic schemes in multisensor context-awareness. In: Proc. PerCom'07. pp. 27–32. IEEE (2007)
17. Durand, A., Hermann, M., Kolaitis, P.G.: Subtractive reductions and complete problems for counting complexity classes. Theoretical Computer Science 340(3), 496–513 (2005)
18. Elkin, P.L., Brown, S.H., Husser, C.S., Bauer, B.A., Wahner-Roedler, D., Rosenbloom, S.T., Speroff, T.: Evaluation of the content coverage of SNOMED CT: ability of SNOMED clinical terms to represent clinical problem lists. Mayo Clin. Proc. 81(6), 741–748 (2006)
19. Glimm, B., Lutz, C., Horrocks, I., Sattler, U.: Conjunctive query answering for the description logic $\mathcal{SHIQ}$. J. Artif. Intell. Res. (JAIR) 31, 157–204 (2008)
20. Hemaspaandra, L.A., Vollmer, H.: The satanic notations: counting classes beyond #P and other definitional adventures. ACM SIGACT News 26(1), 2–13 (1995)

21. Hernich, A., Lemos, J., Wolter, F.: Query answering in DL-Lite with datatypes: A non-uniform approach. In: Proc. AAAI'17 (2017)
22. Hoover, H.J.: Feasible real functions and arithmetic circuits. SIAM Journal on Computing 19(1), 182–204 (1990)
23. Jung, J.C., Lutz, C.: Ontology-based access to probabilistic data with OWL QL. In: Proc. ISWC'12. pp. 182–197. Springer (2012)
24. Ko, K.I.: Complexity Theory of Real Functions. Birkhäuser (1991)
25. Kumar, N., Khunger, M., Gupta, A., Garg, N.: A content analysis of smartphone–based applications for hypertension management. Journal of the American Society of Hypertension 9(2), 130–136 (2015)
26. Lutz, C.: Adding numbers to the $\mathcal{SHIQ}$ description logic—first results. In: Proc. KR'01. pp. 191–202. Citeseer (2001)
27. Lutz, C.: Description logics with concrete domains—a survey. In: Advances in Modal Logic 4. pp. 265–296. King's College Publications (2002)
28. Lutz, C.: The complexity of conjunctive query answering in expressive description logics. In: Proc. IJCAR'08. pp. 179–193. Springer (2008)
29. Lutz, C., Schröder, L.: Probabilistic description logics for subjective uncertainty. In: Proc. KR'10. pp. 393–403. AAAI Press (2010)
30. Lutz, C., Toman, D., Wolter, F.: Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In: Proc. IJCAI'09. pp. 2070–2075. IJCAI/AAAI (2009)
31. Rector, A., Gangemi, A., Galeazzi, E., Glowinski, A., Rossi-Mori, A.: The GALEN CORE model schemata for anatomy: Towards a re-usable application-independent model of medical concepts. In: Proc. MIE'94. pp. 229–233 (1994)
32. Rosati, R.: On conjunctive query answering in $\mathcal{EL}$. In: Proc. DL'07. pp. 451–458. CEUR-WS.org (2007)
33. Savković, O., Calvanese, D.: Introducing datatypes in *DL-Lite*. In: Proc. ECAI'12. pp. 720–725 (2012)
34. Schild, K.: A correspondence theory for terminological logics: Preliminary report. In: Mylopoulos, J., Reiter, R. (eds.) Proc. IJCAI'91. pp. 466–471. Morgan Kaufmann (1991)
35. Singh, S., Mayfield, C., Mittal, S., Prabhakar, S., Hambrusch, S., Shah, R.: Orion 2.0: native support for uncertain data. In: Proc. SIGMOD'08. pp. 1239–1242. ACM (2008)
36. Thrun, S., Burgard, W., Fox, D.: A probabilistic approach to concurrent mapping and localization for mobile robots. Autonomous Robots 5(3-4), 253–271 (1998)
37. Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM computing surveys (CSUR) 38(4), 13 (2006)