Using Ontologies to Query Probabilistic Numerical Data

Franz Baader(✉), Patrick Koopmann(✉), and Anni-Yasmin Turhan(✉)

Institute of Theoretical Computer Science,
Technische Universität Dresden, Dresden, Germany
{franz.baader,patrick.koopmann,anni-yasmin.turhan}@tu-dresden.de

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 [2,9] to P for DLs of the $\mathcal{EL}$ family [28], all the way up to intractable data complexity for expressive DLs such as $\mathcal{ALC}$ and beyond [15].

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 [14] or Galen [29] 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 [21].

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

C. Dixon and M. Finger (Eds.): FroCoS 2017, LNCS 10483, pp. 77–94, 2017.
DOI: 10.1007/978-3-319-66167-4_5

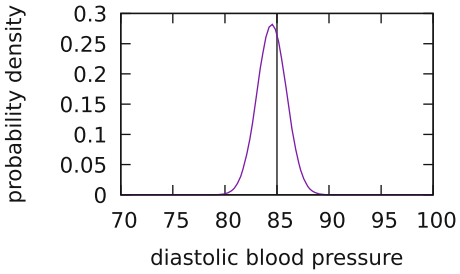

Fig. 1. Measured blood pressure as normal distribution.

What constitutes dangerously high blood pressure (HBP) depends on 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 [24], 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 [3,4,17,31], and develops query rewriting approaches. 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 Fig. 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 [34], tracking of object positions in video analytics [35], and mobile applications using probabilistic sensor data [12], to name a few. The interest in continuous probability distributions is also reflected in the development of database systems that support these [33].

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 [19], 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 $>_r$ for $r \in \mathbb{R}$. 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 [20], 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.

Detailed proofs for all results can be found in the long version of the paper [7].

2 Description Logics with Numerical Domains

We recall basic DLs with concrete domains, as introduced in [6], and give complexity results for classical query answering.

A *concrete domain* is a tuple $\mathcal{D} = (\Delta_\mathcal{D}, \Phi_\mathcal{D})$, where $\Delta_\mathcal{D}$ contains objects of the domain, and $\Phi_\mathcal{D}$ contains predicates P_n with associated arity n and extension $P_n^\mathcal{D} \subseteq \Delta_\mathcal{D}^n$. Let N_c, N_r, N_{cF} and N_i be pair-wise disjoint sets of *names* for *concepts*, *roles*, *concrete features* and *individuals*, respectively. Let $N_{aF} \subseteq N_r$ be a set of *abstract feature names*. Concrete features are partial functions that map individuals to a value in the concrete domain. Abstract features are functional roles and their use in *feature paths* does not harm decidability [23]. A *feature path* is an expression of the form $u = s_1 s_2 \dots s_n g$, where $s_i \in N_{aF}$, $1 \leq i \leq n$, and $g \in N_{cF}$. $\mathcal{ALC}(\mathcal{D})$ concepts are defined as follows, where $A \in N_c$, $s \in N_r$, u and u' are feature paths, $P_n \in \Phi_\mathcal{D}$ is a predicate of arity n, and C_1 and C_2 are $\mathcal{ALC}(\mathcal{D})$ concepts:

$$C \;:=\; \top \;\mid\; A \mid \neg C_1 \;\mid\; C_1 \sqcap C_2 \;\mid\; \exists s.C_1 \;\mid\; \exists(u_1, \dots, u_n).P_n \;\mid\; u{\uparrow}.$$

Additional concepts are defined as abbreviations: $C_1 \sqcup C_2 = \neg(\neg C_1 \sqcap \neg C_2)$, $\forall s.C = \neg\exists s.\neg C$, and $\bot = \neg\top$. If a concept uses only the constructors $\top$, A, $C_1 \sqcap C_2$, $\exists s.C_1$ and $\exists(u_1, \ldots, u_n).P_n$ and no abstract features, it is an $\mathcal{EL}(\mathcal{D})$ concept. The restrictions for $\mathcal{EL}(\mathcal{D})$ concepts ensure polynomial time complexity for standard reasoning tasks. Specifically, as done in [5], we disallow abstract features, since axiom entailment in $\mathcal{EL}$ with functional roles is ExpTime-hard [5].

A *TBox* is a finite set of *general concept inclusion axioms* (GCIs), which are of the form $C \sqsubseteq D$, where C and D are concepts. A *classical ABox* is a finite set of *assertions*, which are of the forms $A(a)$, $s(a, b)$ and $g(a, d)$, where $a, b \in N_i$, $A \in N_c$, $s \in N_r$, $g \in N_{cF}$ and $d \in \Delta^{\mathcal{D}}$. We call GCIs and assertions collectively *axioms*. A *knowledge base* (KB) $\mathcal{K}$ is a pair $(\mathcal{T}, \mathcal{A})$ of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. Given a KB $\mathcal{K}$, we denote by $\mathrm{sub}(\mathcal{K})$ the *subconcepts* occurring in $\mathcal{K}$. Let $\mathcal{L}$ be a DL, then a TBox/KB that uses only $\mathcal{L}$ concepts is a $\mathcal{L}$ TBox/$\mathcal{L}$ KB.

The semantics of $\mathcal{EL}(\mathcal{D})$ and $\mathcal{ALC}(\mathcal{D})$ is defined in terms of interpretations. An *interpretation* is a tuple $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a *set of domain elements* $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$. The *interpretation function* $\cdot^{\mathcal{I}}$ maps individual names to elements of $\Delta^{\mathcal{I}}$, concept names to subsets of $\Delta^{\mathcal{I}}$, concrete features to partial functions $\Delta^{\mathcal{I}} \to \Delta^{\mathcal{D}}$, and role names to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ s.t. for all $s \in N_{aF}$, $s^{\mathcal{I}}$ is a partial function. The extension of $\cdot^{\mathcal{I}}$ to feature paths is $(s_1 \ldots s_n g)^{\mathcal{I}} = g^{\mathcal{I}} \circ s_n^{\mathcal{I}} \circ \ldots \circ s_1^{\mathcal{I}}$, and to (complex) concepts is:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \qquad (\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \qquad (C_1 \sqcap C_2)^{\mathcal{I}} = C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}}$$
$$(\exists s.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x, y) \in s^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$$
$$(\exists(u_1, \ldots, u_n).P)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid (u_1^{\mathcal{I}}(x), \ldots, u_n^{\mathcal{I}}(x)) \text{ is defined and in } P^{\mathcal{D}}\}$$
$$(u\uparrow)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid u^{\mathcal{I}}(x) \text{ is undefined}\}.$$

An axiom α is *true* in an interpretation $\mathcal{I}$, in symbols $\mathcal{I} \models \alpha$, if $\alpha = C \sqsubseteq D$ and $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, $\alpha = C(a)$ and $a^{\mathcal{I}} \in C^{\mathcal{I}}$, $\alpha = s(a, b)$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in s^{\mathcal{I}}$, or $\alpha = g(a, n)$ and $g^{\mathcal{I}}(a) = n$. An interpretation $\mathcal{I}$ is a *model* of a TBox (an ABox), if all GCIs (assertions) in it are true in $\mathcal{I}$. An interpretation is a model of a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, if it is a model of $\mathcal{T}$ and $\mathcal{A}$. A KB is satisfiable iff it has a model. Given a KB $\mathcal{K}$ and an axiom α, we say α *is entailed in* $\mathcal{K}$, in symbols $\mathcal{K} \models \alpha$, iff $\mathcal{I} \models \alpha$ in all models $\mathcal{I}$ of $\mathcal{K}$.

The particular concrete domain to be used needs to be selected carefully, in order to obtain a decidable logic with reasonable complexity bounds. Specifically, axiom entailment with TBoxes already becomes undecidable if $\Delta_{\mathcal{D}} = \mathbb{N}$ and $\Phi_{\mathcal{D}}$ can express incrementation, as well as equality between numbers and with 0 [25]. However, by restricting the predicates to basic comparison operators, decidability cannot only be retained, but an increase of complexity for common reasoning tasks can be avoided when adding such concrete domains to the logic. To pursue this as a goal, we concentrate on two concrete domains that allow for standard reasoning in P and ExpTime, respectively. The first concrete domain is $\mathsf{R} = \{\mathbb{R}, \Phi_{\mathsf{R}}\}$ investigated in [22], where Φ_{R} contains the binary predicates $\{<, =, >\}$ with the usual semantics, and the unary predicates $\{<_r, =_r, >_r \mid r \in \mathbb{R}\}$, where for $\oplus \in \{<, =, >\}$, the extension is defined as

$\oplus_r^{\mathsf{R}} = \{r' \in \mathbb{R} \mid r' \oplus r\}$. This concrete domain allows for axiom entailment in ExpTime, while even small extensions lead to undecidability [22]. The second concrete domain is $\mathsf{R}_> = \{\mathbb{R}, \Phi_{\mathsf{R}_>}\}$, where $\Phi_{\mathsf{R}_>} = \{>_r \mid r \in \mathbb{R}\}$. Since polynomial time reasoning requires the concrete domain to be *convex* [5], we consider this convex concrete domain.

Example 1. The axioms in the introduction only use predicates from $\mathsf{R}_>$ and are in the logic $\mathcal{EL}(\mathsf{R}_>)$. Feature paths and the more expressive concrete domain R allow to compare different values referred to by concrete features. The following more flexible definition of HBP patients compares their diastolic blood pressure (BP) with the maximal diastolic blood pressure assigned to their age group:

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

2.1 Queries

We recall atomic, conjunctive and unions of conjunctive queries. Let N_v be a set of variables disjoint from N_c, N_r, N_{cF} and N_i. An *atom* is of the form $C(x)$ or $s(x, y)$, where C is a concept, $s \in N_r$, $x, y \in N_v \cup N_i$. A *conjunctive query (CQ)* q is an expression of the form $\exists x_1, \ldots, x_n : a_1 \wedge \ldots \wedge a_m$, where $x_1, \ldots, x_n \in N_v$ and $a_1, \ldots, a_m$ are atoms. The variables $x_1, \ldots, x_n$ are the *existentially quantified variables in* q, the remaining variables in q are the *free variables in* q. If a CQ contains only one atom, it is an *atomic query* (AQ). A *union of conjunctive queries (UCQ)* is an expression of the form $q_1 \vee \ldots \vee q_n$, where $q_1, \ldots, q_n$ are CQs with pairwise-disjoint sets of variables. The existentially quantified/free variables of a UCQ are the existentially quantified/free variables of its disjuncts. We call AQs, CQs and UCQs collectively *queries*. A query is *Boolean* if it has no free variables.

Given an interpretation $\mathcal{I}$ and a Boolean CQ q, q is *true in* $\mathcal{I}$, in symbols $\mathcal{I} \models q$, iff there is a mapping π that maps variables in q to domain elements in $\mathcal{I}$ and each $a \in N_i$ to $a^{\mathcal{I}}$ such that for every atom $A(x)$ in q, $\pi(x) \in A^{\mathcal{I}}$, and for every atom $s(x, y)$ in q, $(\pi(x), \pi(y)) \in s^{\mathcal{I}}$. A Boolean UCQ is true in $\mathcal{I}$ iff one of its disjuncts is true in $\mathcal{I}$. Finally, given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and a Boolean query q, q *is entailed by* $\mathcal{K}$, in symbols $\mathcal{K} \models q$, if $\mathcal{I} \models q$ in every model of $\mathcal{K}$. The *query entailment problem* is to decide whether a given Boolean query is entailed by a given KB.

The *query answering* problem is to find a substitution from the free variables in the query to individual names such that the resulting Boolean query is entailed by the KB. Because this problem can be polynomially reduced to query entailment, it is typical to focus on the query entailment problem, which is a decision problem, when analysing computational complexity. We follow the same route in this paper.

Note that according to our definition, concrete features cannot be used outside of concepts in a query. Therefore, 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 $s_1, s_2 \in N_{aF}$, $g_1, g_2 \in N_{cF}$, and $<$ is a predicate of the concrete domain.

Example 2. In a KB with patient records, the following query can be used to retrieve a list of doctors who diagnosed their patients with high blood pressure.

$$\exists y, z : \mathsf{hasPatient}(x, y) \wedge \mathsf{finding}(y, z) \wedge \mathsf{observed}(x, z) \wedge \mathsf{HighBloodPressure}(z)$$

2.2 Complexity of Classical Query Entailment

We give tight complexity bounds for query entailment for the introduced DLs. To the best of our knowledge, the complexity of query answering for the logics studied here has not been considered in the literature before. We focus on the DLs $\mathcal{EL}(\mathsf{R}_>)$ and $\mathcal{ALC}(\mathsf{R})$, since $\mathcal{EL}(\mathsf{R})$ has the same expressive power as $\mathcal{ALC}(\mathsf{R})$ [5], and $\mathcal{ALC}(\mathsf{R}_>)$ already has matching lower bounds from $\mathcal{ALC}$ to our upper bounds for $\mathcal{ALC}(\mathsf{R})$. We further assume values from the concrete domain 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. When analysing complexity of query entailment, we distinguish between *combined* and *data complexity*, where in combined complexity, the size of the complete input is taken into consideration, while for data complexity, everything but the ABox is fixed.

Table 1. Complexity of classical query entailment.

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

An overview of the complexities is shown in Table 1. Since the corresponding lower bounds are the same for CQs as for UCQs, we do not include CQs. Matching lower bounds are already known for the DLs $\mathcal{EL}$ and $\mathcal{ALC}$ [10,30,32], so that adding the respective concrete domains does not increase the complexity of query answering for these logics. We show in the extended version of the paper how to reduce query entailment in $\mathcal{EL}(\mathsf{R}_>)$ to query entailment of $\mathcal{EL}$ KBs, following a technique from [23, Sect. 2.4]. For $\mathcal{ALC}(\mathsf{R})$, the results are based on and match results from [22], [23, Sect. 6.2], and [26], which concern the combined complexities of $\mathcal{SHIQ}(\mathsf{R})$ TBox satisfiability and $\mathcal{ALC}(\mathsf{R})$ KB satisfiability, as well as the combined complexity of query entailment in $\mathcal{SHQ}^{\cap}$.

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* [11], where probabilities are assigned to database and to *ipABoxes* introduced in [19]. 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 [19]).

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 3. 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 $\mathcal{L}_\mathcal{P}$

[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.

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 [11,19,27]. 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 [19] 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. $\Omega_\mathcal{A} \in \Sigma_\mathcal{A}$,
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 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 [19] 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 Sect. 2.1, 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 [19]. 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 [19] 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 [11], that query-entailment over probabilistic data is #·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 [20], 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 $|f(x) - f(y)| < 2^{-n}$ [18,20, Chap. 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 [20, Chap. 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 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 3:

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, we consider corresponding counting problems. Their solutions are obtained by, intuitively, shifting the binary point of an approximated query probability to the right to obtain a natural number. We first recall counting complexity classes following [16].

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*

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

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

Relevant to this section are the counting complexity classes $\#\cdot$P, $\#\cdot$NP and $\#\cdot$coNP. The class $\#\cdot$P is also called #P. The following inclusions are known: $\#\cdot P \subseteq \#\cdot\text{NP} \subseteq \#\cdot\text{coNP} \subseteq \text{FPSpace}$ [16].

In order to characterise the complexity of probabilistic query answering using counting classes, we consider corresponding counting problems, inspired by [20, Chap. 5] and [11]. 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, the above definition allows us to give a complexity upper bound for a counting problem corresponding to probabilistic query entailment in a quite direct way. Without loss of generality, we assume that queries contain only concept names as concepts. If $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ is discrete, the probability measure space $M_{\mathcal{A}}$ has only a finite set $\Omega_{\mathcal{A}}$ of possible worlds, and each possible world $w \in \Omega_{\mathcal{A}}$ has a probability $\mu_{\mathcal{A}}(\{w\})$ that can be represented with a number of bits polynomial in the size of the input. We use this to define a relation R as used in Definition 4. Let $b_{\mathcal{K}}$ be the maximal number of bits used by any probability $\mu_{\mathcal{A}}(\{w\})$, $w \in \Omega_{\mathcal{A}}$. Define the relation R by setting $R((\mathcal{K}, q, n), (w, d))$ for all $w \in \Omega_{\mathcal{A}}$, $d \in \mathbb{N}$ s.t. $(\mathcal{T}, w) \models q$ and $d < 2^{b_{\mathcal{K}}} \cdot \mu_{\mathcal{A}}(\{w\})$, where $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. One easily establishes that $\langle P_{\mathcal{K}}(q)\rangle_n = 2^{-b_{\mathcal{K}}} \cdot \|\{y \mid R((\mathcal{K}, q, n), y)\}\|$ for any $n \in \mathbb{N}$. (Note that our "approximation" is always the precise answer in this case.) For discrete KBs, we thus obtain a complexity upper bound of $\#\cdot\mathcal{C}$ for the corresponding counting problem defined by $g(\mathcal{K}, q, n) = 2^{b_{\mathcal{K}}} \cdot P_{\mathcal{K}}(q)$, where $\mathcal{C}$ is the complexity of classical query entailment.

In order to transfer this approach to continuous probabilistic KBs, we define a discretisation of continuous probability measure spaces based on the precision parameter n and the TBox $\mathcal{T}$. Namely, given a probabilistic KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and a desired precision n, we step-wise modify the measure space $M_{\mathcal{A}}$ into an approximated measure space $M^a_{\mathcal{K},n} = (\Omega^a_{\mathcal{K},n}, \Sigma^a_{\mathcal{K},n}, \mu^a_{\mathcal{K},n})$ such that (i) the size of each possible world $w \in \Omega^a_{\mathcal{K},n}$ is polynomially bounded by $|\mathcal{K}| + n$, (ii) for each $w \in \Sigma^a_{\mathcal{K},n}$, $\mu^a_{\mathcal{K},n}(\{w\})$ can be computed precisely and in time polynomial in $|\mathcal{K}| + n$, and (iii) it holds $\mu^a_{\mathcal{K},n}(\{w \in \Omega^a_{\mathcal{K},n} \mid (\mathcal{T}, w) \models q\}) = \langle P_{\mathcal{K}}(q)\rangle_n$ for every query q. Real numbers occur in $M_{\mathcal{A}}$ in concrete feature values and in the range of $\mu_{\mathcal{A}}$, and have to be replaced by numbers with a polynomially bounded number of bits. We proceed in three steps: (1) we first reduce the number of bits that occur *before* the binary point in any concrete feature value, (2) we then reduce the number of bits that occur *after* the binary point in any concrete feature value, and (3) we finally reduce the number of bits in the range of $\mu_{\mathcal{A}}$.

[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, $g(x) = 2^{p(x)} f(x)$.

We define $\mathbf{C} = \{g_i(a_i) \sim f_i \in \mathcal{A}\}$ as the set of continuous probability assertions in $\mathcal{K}$ and $\mathcal{F} = \{f_i \mid g_i(a_i) \sim f_i \in \mathbf{C}\}$ as the relevant pdfs in $\mathcal{K}$. We also set $n_v = \|\mathbf{C}\|$ and n_c as the number of unary concrete domain predicates in $\mathcal{K}$.

Step 1: Reduce the number of bits before the binary point. Because every function $f \in \mathcal{F}$ has a monotone polynomial bounding function, we can obtain a function $\delta : \mathbb{N} \to \mathbb{N}$ s.t. for every pdf $f \in \mathcal{F}$ and every $n' \in \mathbb{N}$, we have

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

The first step is to remove all possible worlds w in which for some $g(a) \sim f \in \mathbf{C}$, we have $w \not\models g(a) \in [-2^{\delta(n_v+n)}, 2^{\delta(n_v+n)}]$. Note that for each $g(a) \sim f \in \mathcal{A}$, the probability of $g(a)$ to lay outside this interval is 2^{-n_v-n}. Based on this, one can show that for the resulting measure space $M_1 = (\Omega_1, \Sigma_1, \mu_1)$, we have $|\mu_{\mathcal{A}}(\Omega_{\mathcal{A}}) - \mu_1(\Omega_1)| < 2^{-n-1}$. This restricts also the overall error on the probability of any query. Therefore, we have a remaining error of 2^{-n-1} that we can make in subsequent steps. Note that the number of bits before the binary point in any concrete feature value is now polynomially bounded by the input.

Step 2: Reduce the number of bits after the binary point. Intuitively, in this step we "replace" each possible world $w \in \Omega_1$ by a possible world w' that is obtained by "cutting off" in all concrete feature values all digits after a certain position after the binary point, preserving its probability. First, we specify the maximum number m of digits after the binary point we keep. Similar as for the bounding function δ, we can obtain a polynomial function ω that is a modulus of all functions $f \in \mathcal{F}$, and a polynomial function ρ that is a range function of all functions $f \in \mathcal{F}$. Let $k = \rho(n_v + n)$ be the highest number of bits before the binary point in the range of any pdf in the remaining interval $[-2^{\delta(n+n_v)}, 2^{\delta(n+n_v)}]$, and set $l = n_v + \delta(n_v + n) + 2 + n$. Based on k, l and ω, we define the maximal precision m by

$$m = \lceil \log_2(n_v(n_v + n_c)) + k + n + 3 + \omega(l) \rceil .$$

The motivation behind this definition will become clear in the following. For now, just notice that m is polynomially bounded by $|\mathcal{K}| + n$.

In the approximated measure space $M_2 = (\Omega_2, \Sigma_2, \mu_2)$, Ω_2 contains all worlds from Ω_1 in which each concrete feature value has at most m bits after the binary point. To preserve the probabilities, we define a function $\Omega_{2\to1} : \Omega_2 \to 2^{\Omega_1}$ that maps each possible world $w \in \Omega_2$ to the possible worlds in Ω_1 that have been "replaced" by w. $\Omega_{1\to2}$ is defined as

$$\Omega_{2\to1}(w) = \{w' \in \Omega_1 \mid w \cap \mathsf{cl\text{-}ass}(\mathcal{A}) = w' \cap \mathsf{cl\text{-}ass}(\mathcal{A}), \\ \forall g(a,x) \in w, g(a) \sim f \in \mathbf{C} : w' \models g(a) \in [x, x + 2^{-m}]\}.$$

The measure function μ_2 is now defined by

$$\mu_2(\{w\}) = \mu_1(\Omega_{2\to1}(w)).$$

This transformation affects the probability of concepts such as $\exists(g_1, g_2).>$ and $\exists g.>_r$, because the probability that two concrete features have the same value, or that a concrete feature has a value occurring in some unary domain predicate, increases. One can show that this probability is bounded by $n_v(n_v+n_c)\cdot 2^{-m+k+1}$. By definition, $m > \log_2(n_v(n_v + n_c)) + k + n + 3$, so that the error created in this step is bounded by 2^{-n-2}.

Step 3: Reduce the number of bits in the probabilities. Each possible world M_2 can be finitely represented and has a size that is polynomially bounded in the size of the input. However, the probabilities for each possible world are still real numbers. We first explain how we approximate the probabilities for a single concrete feature. For an assertion $g_i(a_i) \sim f_i \in \mathbf{C}$, and a number $x \in \mathbb{R}$ with m bits after the binary point, we have $\mu_2(\{w \in \Omega_2 \mid w \models g(a) = x\}) = \int_x^{x+2^{-m}} f_i(y)\,\mathrm{d}y$. To discretise this probability, we make use of the modulus ω of the pdfs used in $\mathcal{K}$. Recall that, by the definition of a modulus, for any precision $n' \in \mathbb{N}$ and two real numbers $x, y \in [2^{-n'}, 2^{n'}]$, $|x - y| < 2^{-\omega(n')}$ implies $|f_i(x) - f_i(y)| < 2^{-n'}$. By construction, we have $m > \omega(l)$, and hence, for $x \in [2^{-l}, 2^l]$ and $y \in [x, x+2^{-m}]$, we have $|f_i(x)-f_i(y)| < 2^{-l}$. Consequently, the integral $\int_x^{x+2^{-m}} f_i(y)\,\mathrm{d}y$ can be approximated by the product $2^{-m} \cdot \langle f_i(x)\rangle_l$, and we have

$$\left| \int_x^{x+2^{-m}} f_i(y)\,\mathrm{d}y - 2^{-m} \cdot \langle f_i(x)\rangle_l \right| \;<\; 2^{-m-l}.$$

There are $2^{\delta(n_v+n)+1+m}$ different values per concrete feature in our measure space, so that an error of 2^{-m-l} per approximated interval introduces a maximal error of 2^{-n-n_v-1} for each concrete feature value (recall $l = n_v+\delta(n_v+n)+2+n$). If we approximate all pdfs this way, for similar reasons as in Step 1, we obtain a maximal additional error of 2^{-n-2} for any query.

Based on these observations, we define the final discretised measure space. Specifically, we define the measure space $M^a_{\mathcal{K},n} = (\Omega^a_{\mathcal{K},n}, \Sigma^a_{\mathcal{K},n}, \mu^a_{\mathcal{K},n})$, where $\Omega^a_{\mathcal{K},n} = \Omega_2$ and $\mu^a_{\mathcal{K},n}$ is specified by

$$\mu^a_{\mathcal{K},n}(\{w\}) \;=\; \prod_{\substack{\alpha:p\in\mathcal{A}\\ \alpha\in w}} p \cdot \prod_{\substack{\alpha:p\in\mathcal{A}\\ \alpha\notin w}} (1-p) \cdot \prod_{\substack{g(a)\sim f\in\mathcal{A}\\ g(a,x)\in w}} 2^{-m}\langle f(x)\rangle_l.$$

Note that $\mu^a_{\mathcal{K},n}(\{w\})$ can be evaluated in polynomial time, and can be represented with at most $2 + n_a \cdot n_b + n_v \cdot (m + l)$ bits, where n_a is the number of discrete probability assertions and n_b the maximal number of bits in a discrete probability assertion.

Given a probabilistic KB $\mathcal{K}$ and a precision $n \in \mathbb{N}$, we call the measure space $M^a_{\mathcal{K},n}$ constructed above the *n-approximated probability measure space* for $\mathcal{K}$. We have the following lemma.

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

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

Lemma 2. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a probabilistic KB, q a query, $n \in \mathbb{N}$ and $M^a_{\mathcal{K},n}$ the n-approximated probability measure space for $\mathcal{K}$. Then,*

$$\mu^a_{\mathcal{K},n}(\{w \in \Omega^a_{\mathcal{K},n} \mid (\mathcal{T}, w) \models q\}) = \langle P_\mathcal{K}(q) \rangle_n.$$

Note that one can test in polynomial time whether a given possible world is in $\Omega^a_{\mathcal{K},n}$, and compute its probability in polynomial time. Using the observations from the beginning of this section, together with the complexity results in Table 1, we can establish the upper bounds for data and combined complexity shown in Table 2 on counting problems corresponding to probabilistic query answering, which already hold for discrete probabilistic KBs without concrete domain. To the best of our knowledge, only the data complexity for query answering in probabilistic $\mathcal{EL}$ has been considered in the literature before [19], while the other results are new. For the ExpTime upper bounds, note that the approximated measure space has at most exponentially many elements, and can thus be constructed and checked in exponential time.

Hardness for all complexities already holds for discrete probabilistic KBs, so that continuous, #·P-admissible probability distributions do not increase the complexity of probabilistic query answering. A general #·P-lower bound follows from the corresponding complexity of probabilistic query entailment in probabilistic databases [11], 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 for the corresponding counting problems in the extended version of the paper using appropriate reductions. Specifically, we show #·NP-hardness w.r.t. combined complexity under *subtractive reductions* in the case of UCQ entailment in $\mathcal{EL}$, and #·coNP-hardness w.r.t data complexity under *parsimonious reductions* in the case of AQ entailment in $\mathcal{ALC}$ [13].

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 $\mathcal{EL}$ [19], 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 [20, Chap. 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 [33], 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 [19], 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 [8].

References

1. Adams, M.R., Guillemin, V.: Measure Theory and Probability. Springer, Boston (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: Proceedings ECAI 2012, pp. 61–66. IOS Press (2012)
4. Baader, F., Borgwardt, S., Lippmann, M.: Query rewriting for DL-Lite with n-ary concrete domains. In: Proceedings IJCAI 2017 (2017, to appear)
5. Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ envelope. In: Proceedings of IJCAI 2005, pp. 364–369. Professional Book Center (2005)
6. Baader, F., Hanschke, P.: A scheme for integrating concrete domains into concept languages. In: Proceedings of IJCAI 1991, pp. 452–457 (1991)
7. 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). https://lat.inf.tu-dresden.de/research/reports.html
8. Belle, V., Van den Broeck, G., Passerini, A.: Hashing-based approximate probabilistic inference in hybrid domains: an abridged report. In: Proceedings of IJCAI 2016, pp. 4115–4119 (2016)
9. 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)
10. Calvanese, D., Giacomo, G.D., Lembo, D., Lenzerini, M., Rosati, R.: Data complexity of query answering in description logics. Artif. Intell. **195**, 335–360 (2013)

11. Dalvi, N., Suciu, D.: Management of probabilistic data: foundations and challenges. In: Proceedings of SIGMOD 2007, pp. 1–12. ACM (2007)
12. Dargie, W.: The role of probabilistic schemes in multisensor context-awareness. In: Proceedings of PerCom 2007, pp. 27–32. IEEE (2007)
13. Durand, A., Hermann, M., Kolaitis, P.G.: Subtractive reductions and complete problems for counting complexity classes. Theoret. Comput. Sci. **340**(3), 496–513 (2005)
14. 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)
15. 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)
16. Hemaspaandra, L.A., Vollmer, H.: The satanic notations: counting classes beyond $\#P$ and other definitional adventures. ACM SIGACT News **26**(1), 2–13 (1995)
17. Hernich, A., Lemos, J., Wolter, F.: Query answering in DL-Lite with datatypes: a non-uniform approach. In: Proceedings of AAAI 2017 (2017)
18. Hoover, H.J.: Feasible real functions and arithmetic circuits. SIAM J. Comput. **19**(1), 182–204 (1990)
19. Jung, J.C., Lutz, C.: Ontology-based access to probabilistic data with OWL QL. In: Cudré-Mauroux, P., et al. (eds.) ISWC 2012. LNCS, vol. 7649, pp. 182–197. Springer, Heidelberg (2012). doi:10.1007/978-3-642-35176-1_12
20. Ko, K.I.: Complexity Theory of Real Functions. Birkhäuser, Boston (1991)
21. Kumar, N., Khunger, M., Gupta, A., Garg, N.: A content analysis of smartphone-based applications for hypertension management. J. Am. Soc. Hypertens. **9**(2), 130–136 (2015)
22. Lutz, C.: Adding numbers to the $\mathcal{SHIQ}$ description logic–first results. In: Proceedings KR 2001, pp. 191–202. Citeseer (2001)
23. Lutz, C.: The complexity of description logics with concrete domains. Ph.D. thesis, RWTH Aachen (2002)
24. Lutz, C.: Description logics with concrete domains–a survey. In: Advances in Modal Logic 4, pp. 265–296. King's College Publications (2002)
25. Lutz, C.: NExpTime-complete description logics with concrete domains. ACM Trans. Comput. Logic (TOCL) **5**(4), 669–705 (2004)
26. Lutz, C.: The complexity of conjunctive query answering in expressive description logics. In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS (LNAI), vol. 5195, pp. 179–193. Springer, Heidelberg (2008). doi:10.1007/978-3-540-71070-7_16
27. Lutz, C., Schröder, L.: Probabilistic description logics for subjective uncertainty. In: Proceedings of KR 2010, pp. 393–403. AAAI Press (2010)
28. Lutz, C., Toman, D., Wolter, F.: Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In: Proceedings of IJCAI 2009, pp. 2070–2075. IJCAI/AAAI (2009)
29. 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: Proceedings of MIE 1994, pp. 229–233 (1994)
30. Rosati, R.: On conjunctive query answering in $\mathcal{EL}$. In: Proceedings of DL 2007, pp. 451–458. CEUR-WS.org (2007)
31. Savković, O., Calvanese, D.: Introducing datatypes in DL-Lite. In: Proceedings of ECAI 2012, pp. 720–725 (2012)

32. Schild, K.: A correspondence theory for terminological logics: preliminary report. In: Mylopoulos, J., Reiter, R. (eds.) Proceedings of IJCAI 1991, pp. 466–471. Morgan Kaufmann (1991)
33. Singh, S., Mayfield, C., Mittal, S., Prabhakar, S., Hambrusch, S., Shah, R.: Orion 2.0: native support for uncertain data. In: Proceedings of SIGMOD 2008, pp. 1239–1242. ACM (2008)
34. Thrun, S., Burgard, W., Fox, D.: A probabilistic approach to concurrent mapping and localization for mobile robots. Auton. Robots **5**(3–4), 253–271 (1998)
35. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Comput. Surv. (CSUR) **38**(4), 13 (2006)