Probabilistic Hybrid Knowledge Bases under the Distribution Semantics

Marco Alberti[1], Evelina Lamma[2], Fabrizio Riguzzi[1], and Riccardo Zese[2]

[1] Dipartimento di Matematica e Informatica – University of Ferrara
Via Saragat 1, I-44122, Ferrara, Italy
[2] Dipartimento di Ingegneria – University of Ferrara
Via Saragat 1, I-44122, Ferrara, Italy
{marco.alberti,evelina.lamma,fabrizio.riguzzi,riccardo.zese}@unife.it

Abstract. Since Logic Programming (LP) and Description Logics (DLs) are based on different assumptions (the closed and the open world assumption, respectively), combining them provides higher expressiveness in applications that require both assumptions.

Several proposals have been made to combine LP and DLs. An especially successful line of research is the one based on Lifschitz's logic of Minimal Knowledge with Negation as Failure (MKNF). Motik and Rosati introduced Hybrid knowledge bases (KBs), composed of LP rules and DL axioms, gave them an MKNF semantics and studied their complexity. Knorr et al. proposed a well-founded semantics for Hybrid KBs where the LP clause heads are non-disjunctive, which keeps querying polynomial (provided the underlying DL is polynomial) even when the LP portion is non-stratified.

In this paper, we propose Probabilistic Hybrid Knowledge Bases (PHKBs), where the atom in the head of LP clauses and each DL axiom is annotated with a probability value. PHKBs are given a distribution semantics by defining a probability distribution over deterministic Hybrid KBs. The probability of a query being true is the sum of the probabilities of the deterministic KBs that entail the query. Both epistemic and statistical probability can be addressed, thanks to the integration of probabilistic LP and DLs.

1 Introduction

Complex domains are often modeled using Logic Programming (LP) or Description Logics (DLs). Both LP and DLs are based on first order logic so they share many similarities. The main and remarkable difference between them is the domain closure assumption: LP is based on the closed-world assumption (CWA) while DLs use the open-world assumption (OWA). Several authors proposed combinations of LP and DLs. Motik and Rosati [19] define Hybrid Knowledge Bases, composed of a logic program and a DL KB, with semantics based on the logic of Minimal Knowledge with Negation as Failure (MKNF) [15]; as shown by the authors, their proposal exhibits desirable properties (faithfulness, i.e. preservation of the semantics of both formalisms when the other is absent; tightness, i.e. no layering of LP and DL; flexibility, the possibility to view each predicate under both open and closed world assumption; decidability), that each of the other existing approaches to LP and DL integration lacks at least partly.

HKBs can manage domains where different information requires different closure assumptions, such as in legal reasoning; for instance, in [1] it is shown that modeling a real world penal code requires both assumptions.

Many domains, especially those that model the real world, are often characterized by uncertain information. In LP a large number of works have appeared for allowing probabilistic reasoning, leading to the dawn of the Probabilistic Logic Programming (PLP) field. One of the most widespread approaches is the distribution semantics [26]. According to this semantics, a program defines a probability distribution over normal Logic Programs called worlds from which the probability of a query is obtained by marginalization. The distribution semantics underlies many languages such as Logic Programs with Annotated Disjunctions (LPADs), CP-logic and ProbLog. All these languages have the same expressive power as a program in one language can be translated into each of others [27].

Similarly, DLs need as well to manage uncertainty to correctly model real world domains. Some proposals for combining DLs with probability theory exploit graphical models: [10] and [9] exploit Bayesian networks while [12] combine DLs with Markov networks. Differently, other approaches such as [11, 16, 17, 7] exploit Nilsson's probabilistic logic [20] to reason with intervals of probability values.

In [4] we applied the distribution semantics to DLs defining DISPONTE (for "DIstribution Semantics for Probabilistic ONTologiEs"). DISPONTE allows to associate probability values to axioms of a KB. The probability of queries is computed as for PLP languages.

In this paper we propose an approach for defining Probabilistic Hybrid KBs (PHKBs) under the distribution semantics. We combine LPADs with DLs under DISPONTE semantics, both following the distribution semantics. In a PHKB, a query is always either entailed or not entailed in the MKNF sense, so its probability can be computed as for LPADs and DISPONTE.

Halpern [13] distinguishes statistical statements from statements about degrees of belief and presents two examples: "the probability that a randomly chosen bird flies is 0.9" and "the probability that Tweety (a particular bird) flies is 0.9". The first statement captures statistical information about the world while the second captures a degree of belief. The first type of statement is called "Type 1" while the latter "Type 2". The first statement can be read as: given a randomly chosen x in the domain, if x is a bird, the probability that x flies is 0.9, or the conditional probability that x flies given that it is a bird is 0.9. DISPONTE allows to define only "Type 2" statements since the probability associated with an axiom represents the degree of belief in that axiom as a whole. Note that "Type 1" differs from statistical information on the domain such as partial concept overlapping of the form "90% of birds fly". In fact, this second statement means that for every bird we know with certainty whether it flies or not but, of all birds, only 90% fly. However, if each individual bird has probability 0.9 of flying, the expected number of birds that fly is 90% of all birds, so we can model partial overlapping with "Type 2" statements, i.e., with probabilistic statements about individuals. The integration of LP and DLs in PHKBs allows to express a form of statistical probabilistic knowledge that is not permitted by DISPONTE alone: in particular, with the LP part we can express "Type 1" statements.

To understand PHKBs, one needs to first acquire background information about what they combine together. Thus in Section 2 we provide a description of these background notions and set the notation for the current work. In Section 3 we introduce our probabilistic extension to hybrid MKNF knowledge bases and we define their semantics. Section 4 discusses related work while Section 5 concludes the paper and presents remarks on future work.

2 Background and notation

This section is devoted to introducing the background notions required to understand PHKBs. Hybrid KBs, presented in Section 2.3, combine Description Logics (DLs) and Logic Programming (LP) following the Minimal Knowledge with Negation as Failure (MKNF) semantics. Therefore, we start with these blocks to achieve the goal of introducing HKBs. Then, we will discuss probabilistic extensions to LP and DLs and in particular about Logic Programs with Annotated Disjunctions (Sect. 2.4) and DISPONTE (Sect. 2.5), both following the distribution semantics [26].

2.1 Description Logics

DLs are fragments of First Order Logic (FOL) languages used for modeling ontologies [3]. These knowledge representation formalisms differ on which information they permit to define and are usually designed to assure computational properties such as decidability and/or low complexity.

Usually, DLs' syntax is based on concepts, corresponding to sets of individuals, and roles, sets of pairs of individuals of the domain. In order to illustrate DLs, we now describe $\mathcal{SHOIQ}$ [14] as a prototype of expressive DLs.

Let consider a set of *atomic concepts* $\mathbf{C}$, a set of *atomic roles* $\mathbf{R}$ and a set of individuals $\mathbf{I}$. *Concepts* are defined by induction as follows. Each $C \in \mathbf{C}$, $\bot$ and $\top$ are concepts. If C, C_1 and C_2 are concepts and $R \in \mathbf{R}$, then $(C_1 \sqcap C_2)$, $(C_1 \sqcup C_2)$ and $\neg C$ are concepts, as well as $\exists R.C$ and $\forall R.C$. Considering again C, C_1 and C_2, if $S \in \mathbf{R} \cup \mathbf{R}^-$, then $\geq nS.C$ and $\leq nS.C$ for an integer $n \geq 0$ are also concepts. Finally, if $a \in \mathbf{I}$, then $\{a\}$ is a concept called *nominal*.

Roles are either atomic roles $R \in \mathbf{R}$ or their inverse R^- where $R \in \mathbf{R}$. The set of all inverses of roles in $\mathbf{R}$ is denoted by $\mathbf{R}^-$.

A *TBox* $\mathcal{T}$ is a finite set of *concept inclusion axioms* $C \sqsubseteq D$, where C and D are concepts. We use $C \equiv D$ to abbreviate the conjunction of $C \sqsubseteq D$ and $D \sqsubseteq C$. An *RBox* $\mathcal{R}$ consists of a finite set of *transitivity axioms* $Trans(R)$, where $R \in \mathbf{R}$, and *role inclusion axioms* $R \sqsubseteq S$, where $R, S \in \mathbf{R} \cup \mathbf{R}^-$. An *ABox* $\mathcal{A}$ is a finite set of *concept membership axioms* $a : C$, *role membership axioms* $(a, b) : R$, *equality axioms* $a = b$ and *inequality axioms* $a \neq b$, where $C \in \mathbf{C}$, $R \in \mathbf{R}$ and $a, b \in \mathbf{I}$.

A $\mathcal{SHOIQ}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ consists of a TBox $\mathcal{T}$, an RBox $\mathcal{R}$ and an ABox $\mathcal{A}$. It is usually assigned a semantics in terms of interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a non-empty *domain* and $\cdot^{\mathcal{I}}$ is the *interpretation function*. This function assigns an element in $\Delta^{\mathcal{I}}$ to each $a \in \mathbf{I}$, a subset of $\Delta^{\mathcal{I}}$ to each $C \in \mathbf{C}$ and a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to each $R \in \mathbf{R}$.

The *satisfaction* of an axiom E in an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, denoted by $\mathcal{I} \models E$, is defined as follows: (1) $\mathcal{I} \models C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, (2) $\mathcal{I} \models a : C$ iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$, (3) $\mathcal{I} \models (a, b) : R$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$, (4) $\mathcal{I} \models a = b$ iff $a^{\mathcal{I}} = b^{\mathcal{I}}$, (5) $\mathcal{I} \models a \neq b$ iff $a^{\mathcal{I}} \neq b^{\mathcal{I}}$, (6) $\mathcal{I} \models Trans(R)$ iff $R^{\mathcal{I}}$ is transitive, i.e., $\forall X, Y, Z R(X, Y) \wedge R(Y, Z) \rightarrow R(X, Z)$, (7) $\mathcal{I} \models R \sqsubseteq S$ iff $R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$. $\mathcal{I}$ *satisfies* a set of axioms $\mathcal{E}$, denoted by $\mathcal{I} \models \mathcal{E}$, iff $\mathcal{I} \models E$ for all $E \in \mathcal{E}$. An interpretation $\mathcal{I}$ *satisfies* a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$, denoted $\mathcal{I} \models \mathcal{K}$, iff $\mathcal{I}$ satisfies $\mathcal{T}$, $\mathcal{R}$ and $\mathcal{A}$. In this case we say that $\mathcal{I}$ is a *model* of $\mathcal{K}$.

$\mathcal{SHOIQ}$ is decidable iff there are no number restrictions on roles which are transitive or have transitive subroles.

DLs can be directly translated into FOL by using function π_x that maps concept expressions to logical formulas. Table 1 shows the translation of each axiom of $\mathcal{SHOIQ}$ KBs:

Table 1. Translation of $\mathcal{SHOIQ}$ axioms into predicate logic.

Axiom	Translation
$C \sqsubseteq D$	$\forall x.\pi_x(C) \rightarrow \pi_x(D)$
$R \sqsubseteq S$	$\forall x, y.R(x, y) \rightarrow S(x, y)$
$Trans(R)$	$\forall x, y, z.R(x, y) \wedge R(y, z) \rightarrow R(x, z)$
$a : C$	$\pi_a(C)$
$(a, b) : R$	$R(a, b)$
$a = b$	$a = b$
$a \neq b$	$a \neq b$

Example 1. In a social network scenario, the following axioms

$$\begin{array}{ll}
\exists rejectedBy.\top & \sqsubseteq spammer \\
\geq_3 reported.\top & \sqsubseteq hasReports \\
\exists reported.trustedUser & \sqsubseteq hasReports \\
hasReports & \sqsubseteq spammer
\end{array}$$

model the fact that a user is considered a spammer if she or he has had at least a friend request rejected, or if she or he has reports, which means she or he has been reported at least three times, or at least once by a trusted user.

2.2 MKNF

The logic of Minimal Knowledge with Negation as Failure (MKNF) was introduced in [15]. We briefly recall its syntax and semantics, following [19]. The syntax of MKNF is the syntax of first order logic augmented with modal operators **K** and **not** . In the following, Δ is the Herbrand universe of the signature at hand.

An MKNF *structure* is a triple (I, M, N) where I as a first-order interpretation over Δ and M and N are sets of first order interpretations over Δ. Satisfaction of a closed formula by an MKNF structure is defined as follows (where p is an atom and ψ is a formula):

$$\begin{array}{lll}
(I,M,N) \models p & \text{iff} & p \in I \\
(I,M,N) \models \neg\psi & \text{iff} & (I,M,N) \not\models \psi \\
(I,M,N) \models \psi_1 \wedge \psi_2 & \text{iff} & (I,M,N) \models \psi_1 \text{ and } (I,M,N) \models \psi_2 \\
(I,M,N) \models \exists x : \psi & \text{iff} & (I,M,N) \models \psi[\alpha/x] \text{ for some } \alpha \in \Delta \\
(I,M,N) \models \mathbf{K}\,\psi & \text{iff} & (J,M,N) \models \psi \text{ for all } J \in M \\
(I,M,N) \models \mathbf{not}\,\psi & \text{iff} & (J,M,N) \not\models \psi \text{ for some } J \in N
\end{array}$$

An MKNF *interpretation* is a set M of interpretations over Δ. An interpretation M is an MKNF *model* of a closed formula ψ iff

- $(I, M, M) \models \psi$ for all $I \in M$
- for all $M' \supset M$, for some $I' \in M'(I', M', M) \not\models \psi$

A formula ψ *entails* a formula ϕ, written $\psi \models_{\text{MKNF}} \phi$, iff for all MKNF models M of ψ and for all $I \in M$ $(I, M, M) \models \phi$.

2.3 Hybrid Knowledge Bases

Let $\mathcal{DL}$ be a description logic, i.e., a fragment of first order logic such that

- a transformation π (such as the one in Table 1) exists from each knowledge base $\mathcal{O}$ of $\mathcal{DL}$ to a formula of function-free first order logic with equality;
- it supports *ABoxes* (assertions of the form $C(a_1)$ and of the form $R(a_1, a_2)$, where C is an unary predicate -a class-, P is a binary predicate -a role- and the a_i's are $\mathcal{DL}$ constants);
- satisfiability checking and instance checking are decidable.

A Hybrid Knowledge Base (HKB, [19]) is a pair $\mathcal{K} = \langle \mathcal{O}, \mathcal{P} \rangle$ where $\mathcal{O}$ is a $\mathcal{DL}$ knowledge base and $\mathcal{P}$ is a set of LP rules of the form $h \leftarrow a_1, \dots, a_n, \sim b_1, \dots, \sim b_m$, where a_i and b_i are atoms. Note that [19] allow disjunctions in rule heads, but we do not introduce them because they are not required for our definition of *PHKB*s (see Sect. 3).

[19] define HKB's semantics by transforming it into an MKNF formula. More precisely, the transformation π defined for $\mathcal{DL}$ is extended as follows to support LP rules:

- if C is a rule of the form $h \leftarrow a_1, \ldots, a_n, \sim b_1, \ldots, \sim b_m$ and $\mathbf{X}$ is the vector of all variables in C, $\pi(C) = \forall \mathbf{X}(\mathbf{K}\, a_1 \wedge \ldots \wedge \mathbf{K}\, a_n \wedge \mathbf{not}\, b_1 \wedge \ldots \wedge \ldots \mathbf{not}\, b_m \rightarrow \mathbf{K}\, h)$
- $\pi(\mathcal{P}) = \bigwedge_{C \in \mathcal{P}} \pi(C)$
- $\pi(\langle \mathcal{O}, \mathcal{P} \rangle) = \mathbf{K}\, \pi(\mathcal{O}) \wedge \pi(\mathcal{P})$

In Sect. 3, we employ grounding in order to define the semantics of Probabilistic Hybrid Knowledge Bases (PHKBs); for this purpose, it is important for a HKB to have the same MKNF models as its grounding. As shown in [19], a sufficient condition is *DL-safety.* A rule in a HKB $\mathcal{K} = \langle \mathcal{O}, \mathcal{P} \rangle$ is DL-safe if all variables in it occur in a positive atom in its body, whose predicate does not appear in $\mathcal{O}$. A HKB is DL-safe if all the rules in $\mathcal{P}$ are DL-safe. In [19], the authors also argue that non DL-safe knowledge bases can be made DL-safe by a syntactic transformation that does not affect their semantics, so in practice it can be assumed that all knowledge bases are DL-safe.

Example 2. Consider the HKB $\mathcal{K} = \langle \mathcal{O}, \mathcal{P} \rangle$, where $\mathcal{O}$ is the set of axioms defined in Example 1, except for the last one, and $\mathcal{P} =$

$$\begin{aligned} &\mathit{spammer}(X) &&\leftarrow \mathit{hasReports}(X), \sim \mathit{trustedUser}(X). \\ &\mathit{rejectedBy}(X,Y) &&\leftarrow \mathit{invited}(X,Y), \sim \mathit{accepted}(Y,X). \end{aligned}$$

The LP rules define the role *rejectedBy* which occurs in the DL axioms in terms of missing acceptance of a friend request, employing the closed world assumption, and specify that a user with reports is a spammer, but only if she or he is not a trusted user, again using default negation.

The corresponding MKNF formula is

$$\begin{aligned} \pi(\mathcal{K}) = \;& \forall X(\mathbf{K}\, \mathit{hasReports}(X) \wedge \mathbf{not}\, \mathit{trustedUser}(X) \rightarrow \mathbf{K}\, \mathit{spammer}(X)) \\ \wedge \;& \forall X \forall Y(\mathbf{K}\, \mathit{invited}(X,Y) \wedge \mathbf{not}\, \mathit{invited}(Y,X) \rightarrow \mathbf{K}\, \mathit{rejectedBy}(X,Y)) \\ \wedge \;& \mathbf{K}\, (\forall X(\exists Y\, \mathit{rejectedBy}(X,Y) \rightarrow \mathit{spammer}(X)) \\ \wedge \;& \forall X(\exists^{\geq 3} Y\, \mathit{reported}(X,Y) \rightarrow \mathit{hasReports}(X)) \\ \wedge \;& \forall X(\exists Y(\mathit{reported}(X,Y) \wedge \mathit{trustedUser}(Y)) \rightarrow \mathit{hasReports}(X))) \end{aligned}$$

2.4 Probabilistic Logic Programs

We consider Logic Programs with Annotated Disjunctions (LPADs) and we do not allow for function symbols; for the treatment of function symbols, see [22].

LPADs [28] consist of a finite set of annotated disjunctive clauses C_i of the form $h_{i1} : \Pi_{i1}; \ldots; h_{in_i} : \Pi_{in_i} \leftarrow b_{i1}, \ldots, b_{im_i}$. Here, $b_{i1}, \ldots, b_{im_i}$ are logical literals which form the *body* of C_i, denoted by *body*(C_i), while $h_{i1}, \ldots h_{in_i}$ are logical atoms and $\{\Pi_{i1}, \ldots, \Pi_{in_i}\}$ are real numbers in the interval $[0,1]$ such that $\sum_{k=1}^{n_i} \Pi_{ik} \leq 1$. Note that if $n_i = 1$ and $\Pi_{i1} = 1$ the clause corresponds to a non-disjunctive clause. Otherwise, if $\sum_{k=1}^{n_i} \Pi_{ik} < 1$, the head of the annotated disjunctive clause implicitly contains an extra atom *null* that does not appear in the body of any clause and whose annotation is $1 - \sum_{k=1}^{n_i} \Pi_{ik}$. The grounding of an LPAD $\mathcal{P}$ is denoted by *ground*$(\mathcal{P})$.

ground(P) is still an LPAD from which we can obtain a normal logic program by selecting a head atom for each ground clause. In this way we obtain a so-called "world" to which we can assign a probability by multiplying the probabilities of all the head atoms chosen. In this way we get a probability distribution over worlds. We consider only *sound* LPADs, where each possible world w has a total well-founded model, so for a query Q (a ground clause) either $w \models Q$ (Q is true in the well-founded model of w) or $w \not\models Q$, i.e. the well-founded model is two-valued. The probability of a query Q given a world w can be thus defined as $P(Q|w) = 1$ if $w \models Q$ and 0 otherwise. The probability of Q is then:

$$P(Q) = \sum_{w \in W_{\mathcal{P}}} P(Q, w) = \sum_{w \in W_{\mathcal{P}}} P(Q|w)P(w) = \sum_{w \in W_{\mathcal{P}} : w \models Q} P(w) \qquad (1)$$

Example 3. In the same setting of Example 2, the program

$$\begin{array}{l} spammer(john) : 0.3 \leftarrow hasReports(john), {\sim}trustedUser(john). \\ spammer(john) : 0.4 \leftarrow rejectedBy(john, jack). \\ hasReports(john). \qquad rejectedBy(john, jack). \end{array}$$

is ground and has two probabilistic clauses, so there are four worlds. The query *spammer*$(john)$ is true in three of them, i.e., those containing at least one probabilistic clause, and false in the world that does not contain any probabilistic clause. The probability of Q is $0.3 \times 0.4 + 0.3 \times 0.6 + 0.7 \times 0.4 = 0.58$.

2.5 Probabilistic Description Logics

DISPONTE [4] applies the distribution semantics to probabilistic ontologies [26]. In DISPONTE a *probabilistic knowledge base* $\mathbb{O}$ is a set of certain and probabilistic axioms. *Certain axioms* are regular DL axioms. *Probabilistic axioms* take the form $p :: E$, where p is a real number in $[0, 1]$ and E is a DL axiom. Probability p can be interpreted as an *epistemic probability*, i.e., as the degree of our belief in axiom E. For example, a probabilistic concept membership axiom $p :: a : c$ means that we have degree of belief p in $c(a)$. The statement that Tweety flies with probability 0.9 can be expressed as $0.9 :: tweety : flies$.

The idea of DISPONTE is to associate independent Boolean random variables with the probabilistic axioms. By assigning values to every random variable we obtain a *world*, i.e. the set of probabilistic axioms whose random variable takes on value 1 together with the set of certain axioms. DISPONTE defines a probability distribution over worlds as in PLP.

We can now assign probabilities to queries. Given a world w, the probability of a query Q is defined as $P(Q|w) = 1$ if $w \models Q$ and 0 otherwise. The probability of a query can be defined by marginalizing the joint probability of the query and the worlds, as for PLP.

Example 4. Consider the following KBs, a probabilistic version of the one in Example 1:

$$\begin{array}{ll} 0.3 :: \exists rejectedBy.\top & \sqsubseteq spammer \\ \geq_3 reported.\top & \sqsubseteq hasReports \\ \exists reported.trustedUser & \sqsubseteq hasReports \\ 0.5 :: hasReports & \sqsubseteq spammer \end{array}$$

Given that $john : \exists rejectedBy.\top$, the query $john : spammer$ has probability 0.3. With the additional assumption $john : \exists reported.trustedUser$, the probability of $john : spammer$ is given by the sum of the probabilities of the three worlds where either of the probabilistic axioms occurs, that is $0.3 \times 0.5 + 0.3 \times 0.5 + 0.7 \times 0.5 = 0.65$.

3 Probabilistic Hybrid Knowledge Bases

In this section we formally define Probabilistic Hybrid Knowledge Bases (PHKBs), which combine a probabilistic DL knowledge base with a proabbilistic logic program, and their semantics.

Definition 1. *A PHKB is a pair $\mathbb{K} = \langle \mathbb{O}, \mathbb{P} \rangle$ where $\mathbb{O}$ is a DISPONTE knowledge base and $\mathbb{P}$ is an LPAD.*

The semantics of a PHKB, as usually in the distribution semantics approach, is given by considering possible worlds.

Definition 2. *A possible world of a PHKB $\mathbb{K} = \langle \mathbb{O}, \mathbb{P} \rangle$ is a non probabilistic HKB $w = \langle \mathcal{O}, \mathcal{P} \rangle$ where $\mathcal{O}$ is a possible world of $\mathbb{O}$ and $\mathcal{P}$ is a possible world of $\mathbb{P}$, where $\mathbb{P}$'s grounding is performed over $\mathbb{O}$'s individuals together with $\mathbb{P}$'s Herbrand universe.*

The probability distribution over worlds in the PHKB is induced by the probability distributions of worlds in its DL and LP components.

Definition 3. *If $P(\mathcal{P})$ is the probability of $\mathcal{P}$ and $P(\mathcal{O})$ is the probability of $\mathcal{O}$, the probability of $w = \langle \mathcal{O}, \mathcal{P} \rangle$ is $P(w) = P(\mathcal{P})P(\mathcal{O})$.*

It is easy to see that this is a probability distribution over the worlds of $\mathbb{K}$.

We can assign probabilities to queries as for LPAD and DISPONTE, by defining a joint probability distribution over worlds and queries, where a query's conditional probability given a world is 1 if the world entails the query in the MKNF sense (see Sect. 2.2), and 0 otherwise. The probability of a query is again defined by marginalizing the joint probability of the query and the worlds.

Definition 4. *Given a world w, the probability of a query Q is defined as $P(Q|w) = 1$ if $w \models_{MKNF} \mathbf{K}\, Q$ and 0 otherwise.*

The probability of the query is its marginal probability:

$$P(Q) = \sum_{w} P(w) * P(Q|w) \tag{2}$$

Note that Eq. (2) is the sum of the probabilities of the worlds that entail, in the MKNF sense, the query.

A nice result of PHKBs is that they allow coping with both types of probabilities defined by Halpern in [13]. In fact, "Type 1" probabilistic statements about individuals can be expressed using the LP part of PHKB because each LP clause stands for the set of its ground instantiations and there is a different random variable for each instantiation.

Example 5. The knowledge base $\mathbb{K} = \langle \mathbb{O}, \mathbb{P} \rangle$ with

$$\begin{aligned}
\mathbb{P} = & \mathit{soldier}(X) : 0.8 \leftarrow \mathit{person}(X), \mathit{guard}(X). \qquad (3)\\
& \mathit{person}(\mathit{pete}).\\
& \mathit{person}(\mathit{al}).\\
& \mathit{person}(\mathit{john}).\\
\mathbb{O} = & \forall \mathit{commands}.\mathit{soldier} \sqsubseteq \mathit{commander}\\
& \mathit{pete} : \mathit{guard}\\
& \mathit{al} : \mathit{guard}\\
& (\mathit{john}, \mathit{pete}) : \mathit{commands}\\
& (\mathit{john}, \mathit{al}) : \mathit{commands}\\
& \mathit{john} : \forall \mathit{commands}.\mathit{guard}
\end{aligned}$$

expresses that if X is a person and a guard, then X is a soldier with probability 80%. Moreover, we know that those who command only soldiers are commanders, that *pete* and *al* are guards and that *john* commands only guards. Note that this KB is DL-safe because the predicate $\mathit{person}/1$ does not appear in the DL portion. What is the probability of $\mathit{commander}(\mathit{john})$?

There are four pairs of possible worlds (in each pair the worlds are identical, except for the presence of the clause $\mathit{soldier}(\mathit{john}) \leftarrow \mathit{person}(\mathit{john}), \mathit{guard}(\mathit{john})$), which share $\mathcal{O}$ and differ for the LP clauses:

1. both instantiations of (3): $\mathit{soldier}(\mathit{pete}) \leftarrow \mathit{person}(\mathit{pete}), \mathit{guard}(\mathit{pete})$ and $\mathit{soldier}(\mathit{al}) \leftarrow \mathit{person}(\mathit{al}), \mathit{guard}(\mathit{al})$ (probability $0.8 \times 0.8 = 0.64$);
2. $\mathit{soldier}(\mathit{pete}) \leftarrow \mathit{person}(\mathit{pete}), \mathit{guard}(\mathit{pete})$ (probability $0.8 \times 0.2 = 0.16$);
3. $\mathit{soldier}(\mathit{al}) \leftarrow \mathit{person}(\mathit{al}), \mathit{guard}(\mathit{al})$ (probability $0.2 \times 0.8 = 0.16$);
4. no clause(probability $0.2 \times 0.2 = 0.04$).

All MKNF models of world 1 entail $\mathit{soldier}(\mathit{pete})$ and $\mathit{soldier}(\mathit{al})$, so $\mathit{john} : \forall \mathit{commands}.\mathit{soldier}$, and therefore $\mathit{commander}(\mathit{john})$; this means that world 1 entails $\mathit{commander}(\mathit{john})$. $\mathit{soldier}(\mathit{al})$ is not entailed by world 2, so neither is $\mathit{commander}(\mathit{john})$. Likewise, worlds 3 and 4 do not entail $\mathit{commander}(\mathit{john})$. The probability of $\mathit{commander}(\mathit{john})$ is thus the probability of world 1, i.e., 0.64.

This example shows that PHKB allows "Type 1" statements: the only LP rule (3) models the fact that an individual guard has 80% of being a soldier. In this way PHKB highly extends the expressive power of DISPONTE.

Please note however that rule (3) is not equivalent to saying that 80% of guards are soldiers. In this case in fact the query would be false with probability 1, as there exist guards that are not soldiers so *john* does not command only soldiers. However, the expected number of soldiers, given that there are two guards, is $2 \times 0.64 + 1 \times 0.16 + 1 \times 0.16 + 0 \times 0.04 = 1.6$, which is 80% of 2.

Example 6. Consider the knowledge base $\mathbb{K} = \langle \mathbb{O}, \mathbb{P} \rangle$, a probabilistic version of the knowledge base in Example 2, with

$$\begin{aligned} \mathbb{P} =& \mathit{spammer}(X) : 0.3 \leftarrow \mathit{hasReports}(X), {\sim}\mathit{trustedUser}(X). & (4) \\ & \mathit{rejectedBy}(X, Y) \leftarrow \mathit{invited}(X, Y), {\sim}\mathit{accepted}(Y, X). & \\ \mathbb{O} =& 0.4 \ :: \ \exists \mathit{rejectedBy}.\top \sqsubseteq \mathit{spammer} & (5) \\ & \geq_3 \mathit{reported}.\top \sqsubseteq \mathit{hasReports} & \\ & \exists \mathit{reported}.\mathit{trustedUser} \sqsubseteq \mathit{hasReports} & \end{aligned}$$

Here, the probability of a randomly chosen user reported for spamming activity that is not trusted is considered a spammer is 0.3, so the expected number of spammers is 30% that of untrusted users reported for spamming.

With the assertion $\langle \mathit{john}, \mathit{mary} \rangle : \mathit{invited}$, the KB has eight worlds and the query $Q = \mathit{john} : \mathit{spammer}$ is true in four of them, those containing axiom (5). The probability of the query is $0.7 \times 0.7 \times 0.4 + 0.3 \times 0.7 \times 0.4 + 0.7 \times 0.3 \times 0.4 + 0.3 \times 0.3 \times 0.4 = 0.4$.

Adding the assertions $\mathit{john} :\geq_3 \mathit{reported}.\top$, Q is true in six worlds: those that contain axiom (5) and (4) with X instantiated to *john*. The probability of Q is thus $0.3 \times 0.7 \times 0.6 + 0.7 \times 0.7 \times 0.4 + 0.3 \times 0.3 \times 0.6 + 0.3 \times 0.7 \times 0.4 + 0.7 \times 0.3 \times 0.4 + 0.3 \times 0.3 \times 0.4 = 0.58$.

Finally, if $\mathit{john} : \mathit{trustedUser}$ holds, the rule (4) can no longer be used to entail Q, so the probability is 0.4.

Example 7. We now consider the example in Section 4 of [18] and make one of the DL axioms probabilistic thus obtaining $\mathbb{K} = \langle \mathbb{O}, \mathbb{P} \rangle$ with:

$$\begin{aligned} \mathbb{P} =& \mathit{notMarried}(X) \leftarrow \mathit{person}(X), {\sim}\mathit{married}(X). & \\ & \mathit{discount}(X) \leftarrow \mathit{spouse}(X, Y), \mathit{person}(X), \mathit{person}(Y). & \\ \mathbb{O} =& \mathit{notMarried} \equiv \neg \mathit{married} & \\ & 0.4 \ :: \ \mathit{notMarried} \sqsubseteq \mathit{highRisk} & (6) \\ & \exists \mathit{spouse}.\top \sqsubseteq \mathit{married} & \\ & \mathit{john} : \mathit{person} & \end{aligned}$$

In [18], $\mathit{highRisk}(\mathit{john})$ is entailed by the deterministic knowledge base. In the probabilistic version, there are two possible worlds: one where axiom (6) occurs, whose probability is 0.4, and one where it does not. The query $\mathit{highRisk}(\mathit{john})$ is true only in the world where axiom (6) occurs, so its probability is 0.4.

4 Related Work

Despite the large number of proposals regarding the combination of probability and LP or DLs, the field of hybrid KBs is still in fast development. Moreover, to the best of our knowledge the definition of probabilistic hybrid KBs represents a completely new field.

FOProbLog [6] is an extension of ProbLog where a program contains a set of *probabilistic facts*, i.e. facts annotated with probabilities, and a set of general clauses which can have positive and negative probabilistic facts in their body. Each fact is assumed to be probabilistically independent. FOProbLog follows the distribution semantics and exploits Binary Decision Diagrams to compute the probability of queries. Differently from our approach, it follows only the open world assumption. Moreover, it permits to associate probability values only to facts. In our case a probability value can be associated also with implications in the LP part and with TBox axioms in the DL part of the hybrid KB. FOProbLog is a reasoner for FOL that is not tailored to DLs, so the algorithm could be suboptimal for them.

In [8] the authors extend the definition of ontologies to allow the management of information under the open world assumption. Axioms not included in the KB are called *open* and are associated with a probability interval between 0 and a fixed threshold. In this way a query can return a probability interval in the case open information is used for answering a query. Moreover, a background knowledge can be specified to restrict the worlds defined by the open KB. Similar to [6], only assertional probabilistic data is allowed.

A combination between DLs and logic programs was presented in [7] in order to integrate ontologies and rules. They use a tightly coupled approach to (probabilistic) disjunctive description logic programs. They define a description logic program as a pair (L, P), where L is a DL KB and P is a disjunctive logic program which contains rules on concepts and roles of L. P may contain probabilistic alternatives in the style of ICL [21]. The integration follows Nilsson's probabilistic logic [20] approach.

Nilsson's logic allows weaker conclusions than the distribution semantics: consider a probabilistic DISPONTE ontology composed of the axioms $0.4 :: a : C$ and $0.5 :: b : C$ and a probabilistic Nilsson KB composed of $C(a) \geq 0.4$ and $C(b) \geq 0.5$. The distribution semantics permits the derivation of $P(a : C \vee b : C) = 0.4 \times (1 - 0.5) + (1 - 0.4) \times 0.5 + 0.4 \times 0.5 = 0.7$. Differently, Nilsson's logic returns the lowest p such that Pr satisfies all the $F \geq p$ in the KB. Since in Nilsson's logic Pr satisfies $F \geq p$ iff $Pr(F) \geq p$, in this example the lowest p such that $Pr(C(a) \vee C(b)) \geq p$ holds is 0.5. This is due to the fact that in the distribution semantics the probabilistic axioms are considered as independent, which allows to make stronger conclusions, without limiting the expressive power as any probabilistic dependence can be modeled.

5 Conclusions and future work

In this paper we introduced Probabilistic Hybdrid Knowledge Bases, an extension of Hybrid MKNF Knowledge Bases to support probabilistic reasoning, and gave them a distribution semantics.

The next step is to provide a reasoner for PHKBs. The SLG($\mathcal{O}$) procedure [2] for hybrid knowledge bases under the well founded semantics integrates a reasoner for the DL at hand with the SLG procedure in the form of an oracle: the DL reasoner returns the LP atoms that need to be true for the query to succeed. We are following a similar approach for PHKBs, integrating the TRILL probabilistic DL reasoner [23, 29] with the PITA algorithm [25] for PLP reasoning. We also plan to develop a web application for using the system, similarly to what we have done for TRILL[3] [5] and PITA[4] [24].

Acknowledgement This work was supported by the "GNCS-INdAM".

References

1. Alberti, M., Gomes, A.S., Gonçalves, R., Leite, J., Slota, M.: Normative systems represented as hybrid knowledge bases. In: Leite, J., Torroni, P., Ågotnes, T., Boella, G., van der Torre, L. (eds.) Computational Logic in Multi-Agent Systems - 12th International Workshop, CLIMA XII, Barcelona, Spain, July 17-18, 2011. Proceedings. Lecture Notes in Artificial Intelligence, vol. 6814, pp. 330–346. Springer Verlag (2011)
2. Alferes, J.J., Knorr, M., Swift, T.: Query-driven procedures for hybrid MKNF knowledge bases. ACM Trans. Comput. Logic 14(2), 16:1–16:43 (2013)
3. Baader, F., Horrocks, I., Sattler, U.: Description Logics, chap. 3, pp. 135–179. Elsevier, Amsterdam (2008)
4. Bellodi, E., Lamma, E., Riguzzi, F., Albani, S.: A distribution semantics for probabilistic ontologies. In: 7th International Workshop on Uncertainty Reasoning for the Semantic Web. CEUR Workshop Proceedings, vol. 778, pp. 75–86. Sun SITE Central Europe, Aachen, Germany (2011)
5. Bellodi, E., Lamma, E., Riguzzi, F., Zese, R., Cota, G.: A web system for reasoning with probabilistic OWL. Software: Practice and Experience (2016), to appear
6. Bruynooghe, M., Mantadelis, T., Kimmig, A., Gutmann, B., Vennekens, J., Janssens, G., De Raedt, L.: Problog technology for inference in a probabilistic first order logic. In: ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings. Frontiers in Artificial Intelligence and Applications, vol. 215, pp. 719–724. IOS Press (2010)
7. Calì, A., Lukasiewicz, T., Predoiu, L., Stuckenschmidt, H.: Tightly coupled probabilistic description logic programs for the semantic web. In: Journal on Data Semantics XII, pp. 95–130. Springer (2009)
8. Ceylan, İ.İ., Darwiche, A., den Broeck, G.V.: Open-world probabilistic databases. In: Baral, C., Delgrande, J.P., Wolter, F. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25-29, 2016. pp. 339–348. AAAI Press (2016)

[3] http://trill.lamping.unife.it

[4] http://cplint.lamping.unife.it

9. Ceylan, İ.İ., Peñaloza, R.: Bayesian description logics. In: Bienvenu, M., Ortiz, M., Rosati, R., Simkus, M. (eds.) Informal Proceedings of the 27th International Workshop on Description Logics, Vienna, Austria, July 17-20, 2014. CEUR Workshop Proceedings, vol. 1193, pp. 447–458. CEUR-WS.org (2014)
10. d'Amato, C., Fanizzi, N., Lukasiewicz, T.: Tractable reasoning with bayesian description logics. In: Greco, S., Lukasiewicz, T. (eds.) Scalable Uncertainty Management, Second International Conference, SUM 2008, Naples, Italy, October 1-3, 2008. Proceedings. LNCS, vol. 5291, pp. 146–159. Springer (2008)
11. Giugno, R., Lukasiewicz, T.: P-SHOQ(D): A probabilistic extension of SHOQ(D) for probabilistic ontologies in the semantic web. In: Flesca, S., Greco, S., Leone, N., Ianni, G. (eds.) Logics in Artificial Intelligence, European Conference, JELIA 2002, Cosenza, Italy, Proceedings. Lecture Notes in Computer Science, vol. 2424, pp. 86–97. Springer (2002)
12. Gottlob, G., Lukasiewicz, T., Simari, G.I.: Conjunctive query answering in probabilistic datalog+/- ontologies. In: Rudolph, S., Gutierrez, C. (eds.) 5th International Conference on Web Reasoning and Rule Systems (RR 2011), Galway, Ireland. Lecture Notes in Computer Science, vol. 6902, pp. 77–92. Springer (2011)
13. Halpern, J.Y.: An analysis of first-order logics of probability. Artif. Intell. 46(3), 311–350 (1990)
14. Horrocks, I., Sattler, U.: A tableau decision procedure for SHOIQ. J. Autom. Reasoning 39(3), 249–276 (2007)
15. Lifschitz, V.: Nonmonotonic databases and epistemic queries. In: Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1. pp. 381–386. IJCAI'91, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1991)
16. Lukasiewicz, T.: Probabilistic default reasoning with conditional constraints. Ann. Math. Artif. Int. 34(1-3), 35–88 (2002)
17. Lukasiewicz, T.: Expressive probabilistic description logics. Artif. Int. 172(6-7), 852–883 (2008)
18. Motik, B., Rosati, R.: A faithful integration of description logics with logic programming. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence. pp. 477–482. IJCAI'07, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2007)
19. Motik, B., Rosati, R.: Reconciling description logics and rules. J. ACM 57(5), 30:1–30:62 (Jun 2010)
20. Nilsson, N.J.: Probabilistic logic. Artif. Intell. 28(1), 71–87 (1986)
21. Poole, D.: The Independent Choice Logic for modelling multiple agents under uncertainty. Artif. Intell. 94, 7–56 (1997)
22. Riguzzi, F.: The distribution semantics for normal programs with function symbols. International Journal of Approximate Reasoning 77, 1 – 19 (October 2016)
23. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R., Cota, G.: Learning probabilistic description logics. In: Bobillo, F., Carvalho, R.N., Costa, P.C., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) Uncertainty Reasoning for the Semantic Web III, pp. 63–78. Lecture Notes in Computer Science, Springer International Publishing, Berlng, Heidelberg (2014)
24. Riguzzi, F., Bellodi, E., Lamma, E., Zese, R., Cota, G.: Probabilistic logic programming on the web. Software Pract. and Exper. (2015)
25. Riguzzi, F., Swift, T.: The PITA system: Tabling and answer subsumption for reasoning under uncertainty. Theor. Pract. Log. Prog. 11(4–5), 433–449 (2011)

26. Sato, T.: A statistical learning method for logic programs with distribution semantics. In: Sterling, L. (ed.) 12th International Conference on Logic Programming, Tokyo, Japan. pp. 715–729. MIT Press, Cambridge, Massachusetts (1995)
27. Vennekens, J., Verbaeten, S.: Logic programs with annotated disjunctions. Tech. Rep. CW386, KU Leuven (2003)
28. Vennekens, J., Verbaeten, S., Bruynooghe, M.: Logic programs with annotated disjunctions. In: 20th International Conference on Logic Programming. LNCS, vol. 3131, pp. 195–209. Springer, Berlin Heidelberg, Germany (2004)
29. Zese, R., Bellodi, E., Riguzzi, F., Lamma, E.: Tableau reasoners for probabilistic ontologies exploiting logic programming techniques. In: Bellodi, E., Bonfietti, A. (eds.) Doctoral Consortium (DC) co-located with the 14th Conference of the Italian Association for Artificial Intelligence (AI*IA 2015). CEUR Workshop Proceedings, vol. 1485, pp. 1–6. Sun SITE Central Europe, Aachen, Germany (2015)