Semantics and Inference for Probabilistic Ontologies

Fabrizio Riguzzi, Elena Bellodi, Evelina Lamma, and Riccardo Zese

ENDIF – University of Ferrara, Italy,
email: {fabrizio.riguzzi, elena.bellodi, evelina.lamma}@unife.it, riccardo.zese@student.unife.it

Abstract. We present BUNDLE, a reasoner able to perform reasoning on probabilistic knowledge bases according to the semantics DISPONTE. In DISPONTE the axioms of a probabilistic ontology can be annotated with an epistemic or a statistical probability. The epistemic probability represents a degree of confidence in the axiom, while the statistical probability considers the populations to which the axiom is applied. BUNDLE exploits an underlying OWL DL reasoner, which is Pellet, that is able to return explanations for a query. However, it can work well with any reasoner able to return explanations for a query. The explanations are encoded in a Binary Decision Diagram from which the probability of the query is computed.

1 Introduction

Uncertainty has been recognized as an important feature for the Semantic Web [13]. In order to be able to represent and reason with probabilistic knowledge, various authors have advocated the use of probabilistic ontologies and many proposals have been put forward for allowing ontology languages, and OWL in particular, to represent uncertainty [10]. In the field of logic programming, the distribution semantics [11] has emerged as one of the most effective approaches.

In this paper we apply this approach to ontological languages and, in particular, to the OWL DL fragment, that is based on the description logic $\mathcal{SHOIN}(\mathbf{D})$. However, the approach is applicable in principle to any description logic. We called the approach DISPONTE for "DIstribution Semantics for Probabilistic ONTologiEs". The idea is to annotate axioms of a theory with a probability and assume that each axiom is independent of the others. As in Probabilistic Logic Programming, the probability of a query is computed from a covering set of explanations by solving a disjoint sum problem.

We also present the algorithm BUNDLE - for "Binary decision diagrams for Uncertain reasoNing on Description Logic thEories" - that performs inference over probabilistic ontologies.

The paper is organized as follows. Section 2 illustrates Description Logics. Section 3 presents DISPONTE. Section 4 describes BUNDLE and presents the results of the experimental comparison between the probabilistic reasoners BUNDLE and PRONTO [3].

2 Description Logics

Description Logics (DLs) are a family of formalisms used to represent knowledge. Studies on DLs are focused on finding ways to build intelligent applications, able to find the implicit consequences starting from the explicit knowledge. DLs are particularly useful for representing ontologies and have been adopted as the basis of the Semantic Web. For example, the OWL DL sublanguage of OWL is based on the $\mathcal{SHOIN}(\mathbf{D})$ DL. While DLs can be translated into predicate logic, they are usually represented using a syntax based on concepts and roles. A concept corresponds to a set of individuals of the domain while a role corresponds to a set of couples of individuals of the domain. In order to illustrate DLs, we now describe $\mathcal{SHOIN}$ following [6].

Let $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{I}$ be sets of *atomic concepts*, *roles* and *individuals*, respectively. A *role* is either an atomic role $R \in \mathbf{R}$ or the inverse R^- of an atomic role $R \in \mathbf{R}$. We use $\mathbf{R}^-$ to denote the set of all inverses of roles in $\mathbf{R}$. A 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 $R, S \in \mathbf{R} \cup \mathbf{R}^-$. *Concepts* are defined by induction as follows. Each $A \in \mathbf{A}$ is a concept, $\bot$ and $\top$ are concepts, and if $a \in \mathbf{I}$, then $\{a\}$ is a concept. If C, $C1$ and $C2$ are concepts and $R \in \mathbf{R} \cup \mathbf{R}^-$, then $(C_1 \sqcap C_2)$, $(C_1 \sqcup C_2)$, and $\neg C$ are concepts, as well as $\exists R.C$, $\forall R.C$, $n \geq R$ and $n \leq R$ for an integer $n \geq 0$. 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 $C \sqsubseteq D$ and $D \sqsubseteq C$. A *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 is a concept, $R \in \mathbf{R}$ and $a, b \in \mathbf{I}$. A *knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ consists of a TBox $\mathcal{T}$, an RBox $\mathcal{R}$ and an ABox $\mathcal{A}$.

The semantics of DLs can be given in a set-theoretic way or by transforming a DL knowledge base into a predicate logic theory and then using the model-theoretic semantics of the resulting theory.

$\mathcal{SHOIN}(\mathbf{D})$ adds to $\mathcal{SHOIN}$ datatype roles, i.e., roles that map an individual to an element of a datatype such as integers, floats, etc.

A query over a knowledge base is usually an axiom for which we want to test the entailment from the knowledge base.

3 The DISPONTE Semantics for Probabilistic Ontologies

A *probabilistic knowledge base* is a set of *certainly true axioms*, that take the form of DL axioms, of *epistemic probabilistic axioms* of the form $p ::_e E$ where p is a real number in $[0, 1]$ and E is a TBox, RBox or ABox axiom, and of *statistical probabilistic axioms* of the form $p ::_s E$ where p is a real number in $[0, 1]$ and E is a TBox or RBox axiom. In epistemic probabilistic axioms, p is interpreted as the degree of our belief in axiom E, while in statistical probabilistic axioms, p is interpreted as information regarding random individuals from certain populations. For example, an epistemic probabilistic concept inclusion axiom of the form $p ::_e C \sqsubseteq D$ represents the fact that we believe in the truth of $C \sqsubseteq D$ with probability p. A statistical probabilistic concept inclusion axiom of the form $p ::_s C \sqsubseteq D$ instead means that a random individual of class C has probability p of belonging to D, thus representing the statistical information that a fraction p of the individuals of C belong to D. In this way, the overlap between C and D is quantified. The difference between the two axioms is that, if two individuals belong to class C, the probability that they both belong to D according to the epistemic axiom is p, while according to the statistical axiom is $p \times p$.

In order to give a semantics to such probabilistic knowledge bases, we consider their translation into predicate logic. The idea of DISPONTE is to associate independent Boolean random variables to instantiations of the formula in predicate logic that is obtained from the translation of the axiom. An instantiation is a substitution θ for a logical formula F, consisting of pairs x/a, where x is a variable universally quantified by the outermost quantifier and a is an individual.

We now formally define the semantics in the following. An *atomic choice*, in this context, is a triple (F_i, θ_j, k), where F_i is the formula obtained by translating the ith axiom, θ_j is a substitution and $k \in \{0, 1\}$. A set of atomic choices κ is *consistent* if $(F_i, \theta_j, k) \in \kappa$ and $(F_i, \theta_j, m) \in \kappa$ implies $k = m$. A *composite choice* κ is a set of atomic choices and its probability is $P(\kappa) = \prod_{(F_i,\theta_j,1)\in\kappa} p_i \prod_{(F_i,\theta_j,0)\in\kappa}(1 - p_i)$. A *selection* σ is a total composite choice, i.e., an atomic choice for every instantiation of formulas of the theory. A selection σ identifies a theory w_σ called a *world* in this way: $w_\sigma = \{F_i\theta_j|(F_i, \theta_j, 1) \in \sigma\}$. A composite choice κ identifies a set of worlds $\omega_\kappa = \{w_\sigma|\sigma \in \mathcal{S}_T, \sigma \supseteq \kappa\}$, where $\mathcal{S}_T$ is the set of all selections. A composite choice κ is an *explanation* for a query Q if Q is entailed by every world of ω_κ. We define the set of worlds identified by a set of composite choices K as $\omega_K = \bigcup_{\kappa\in K} \omega_\kappa$. A set of composite choices K is *covering* with respect to Q if every world w_σ in which Q is entailed is such that $w_\sigma \in \omega_K$. Two composite choices κ_1 and κ_2 are *incompatible* if their union is inconsistent. A set K of composite choices is *mutually incompatible* if for all $\kappa_1 \in K, \kappa_2 \in K, \kappa_1 \neq \kappa_2 \Rightarrow \kappa_1$ and κ_2 are incompatible.

Now we can define a unique probability measure $\mu : \Omega \to [0, 1]$, where $\Omega = \{\omega_K|K \text{ is a finite set of finite composite choices}\}$. μ is defined by $\mu(\omega_K) = \sum_{\kappa\in K'} P(\kappa)$ where K' is a finite mutually incompatible set of finite composite choices such that $\omega_K = \omega_{K'}$.

Example 1. Let us consider the following simple ontology, inspired by the `people+pets` ontology proposed in [8]:

$$\begin{aligned}
&\exists hasAnimal.Pet \sqsubseteq PetOwner\\
&(kevin, fluffy) : hasAnimal\\
&(kevin, tom) : hasAnimal\\
&fluffy : Cat\\
&tom : Cat\\
0.6 ::_e\ &Cat \sqsubseteq Pet
\end{aligned}$$

The predicate logic formula (without external quantifiers) equivalent to the only probabilistic axiom above is: $F_1 = Cat(x) \to Pet(x)$. A covering set of explanations for the query axiom $Q = kevin : PetOwner$ is $K = \{\kappa_1\}$ with $\kappa_1 = \{(F_1, \emptyset, 1)\}$. In fact, there is only one probabilistic axiom and its presence is necessary to entail the query. This is also a mutually exclusive set of explanations since it contains a single composite choice so $P(Q) = 0.6$.

Example 2. If the axiom $0.6 ::_e Cat \sqsubseteq Pet$ in Example 1 is replaced by $0.6 ::_s Cat \sqsubseteq Pet$ then the query would have the explanations $K = \{\kappa_1, \kappa_2\}$, where $\kappa_1 = \{(F_1, \{x/fluffy\}, 1)\}$ and $\kappa_2 = \{(F_1, \{x/tom\}, 1)\}$. The set $K' = \{\kappa'_1, \kappa'_2\}$, where

$\kappa'_1 = \{(F_1, \{x/\mathit{fluffy}\}, 1)\}$ and $\kappa'_2 = \{(F_1, \{x/\mathit{fluffy}\}, 0), (F_1, \{x/tom\}, 1)\}$, is such that $\omega_K = \omega_{K'}$ and K' is mutually incompatible, so $P(Q) = 0.6 + 0.6 \cdot 0.4 = 0.84$. K' can be found by applying the splitting algorithm of [9].

4 Query Answering and Experiments

The BUNDLE algorithm computes the probability of queries from a probabilistic ontology that follows the DISPONTE semantics. BUNDLE needs an underlying DL reasoner, such as Pellet [12], that is able to return explanations for queries. BUNDLE builds a Binary Decision Diagram (BDD) from the set of explanations. The BDD is then used to compute the probability using the dynamic programming algorithm of [1].

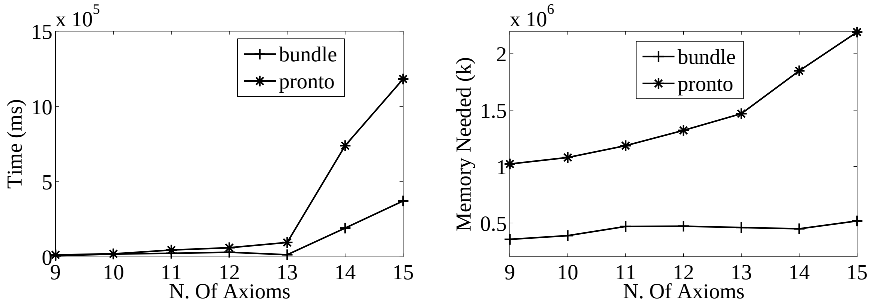
If the knowledge base contains only epistemic probabilistic axioms, Pellet can be used directly as the underlying ontology reasoner. If the knowledge base contains also statistical probabilistic axioms, Pellet needs to be modified so that it records, besides the axioms that have been used to answer the query, also the individuals to which they are applied. Each tableau expansion rule used by Pellet returns a set of uninstantiated axioms. Therefore we have modified Pellet's expansion rules in order to return a set of couples $(axiom, substitution)$ instead of simple axioms.

In order to evaluate the performances of BUNDLE, we followed the methodology of [4] where the system PRONTO [3] is used to answer queries to increasingly complex ontologies, obtained by randomly sampling axioms from a large probabilistic ontology for breast cancer risk assesment (BRCA). This ontology is divided into two parts: a classical and a probabilistic part. The probabilistic part contains conditional constraints [2] of the form $(D|C)[l, u]$ that informally mean "generally, if an object belongs to C, then it belongs to D with a probability in $[l, u]$". For instance, the statement that an average woman has up to 12.3% of developing breast cancer in her lifetime is expressed by

$$(WomanUnderAbsoluteBRCRisk|Woman)[0, 0.123]$$

Tests have been defined by randomly sampling a subset of conditional constraints from the probabilistic part and adding these constraints to the classical part to form a random ontology. We varied the number of constraints from 9 to 15, and, for each number, we repeatedly sampled ontologies and tested them for consistency. We stopped sampling when we obtained 100 consistent ontologies for each number of constraints. The ontologies have then been translated into DISPONTE by replacing the constraint $(D|C)[l, u]$ with the axiom $l ::_s C \sqsubseteq D$. For each ontology we performed the query $a : C$, where a is a new individual for which a number of class assertions are added to the ontology: a was randomly assigned to each class appearing in the sampled conditional constraints with probability 0.6. Class C of the query was randomly selected among those representing women under increased and lifetime risk.

We then applied both BUNDLE and PRONTO to each generated test and we measured the execution time and the memory used. Figure 1(a) shows the average execution time as a function of the number of axioms and, similarly, Figure 1(b) shows the average amount of memory used. These graphs show that BUNDLE is faster and uses less memory than PRONTO. The comparison is not meant to be interpreted in terms of a superiority of BUNDLE, since the two systems treat ontologies with different semantics, rather, it should provide a qualitative evaluation of the complexity of the DISPONTE semantics with respect to the one of [2] that is based on lexicographic entailment [5] and Nilsson's probabilistic logic [7]. In the future we plan to investigate the application of BUNDLE to other real life ontologies.



(a) Execution times (ms).

(b) Memory used (Kb).

Fig. 1. Comparison between BUNDLE and PRONTO.

References

1. De Raedt, L., Kimmig, A., Toivonen, H.: ProbLog: A probabilistic Prolog and its application in link discovery. In: International Joint Conference on Artificial Intelligence. pp. 2462–2467 (2007)
2. Giugno, R., Lukasiewicz, T.: P-SHOQ(D): A probabilistic extension of SHOQ(D) for probabilistic ontologies in the semantic web. In: European Conference on Logics in Artificial Intelligence. LNCS, vol. 2424, pp. 86–97. Springer (2002)
3. Klinov, P.: Pronto: A non-monotonic probabilistic description logic reasoner. In: European Semantic Web Conference. LNCS, vol. 5021, pp. 822–826. Springer (2008)
4. Klinov, P., Parsia, B.: Optimization and evaluation of reasoning in probabilistic description logic: Towards a systematic approach. In: International Semantic Web Conference. LNCS, vol. 5318, pp. 213–228. Springer (2008)
5. Lehmann, D.J.: Another perspective on default reasoning. Ann. Math. Artif. Intell. 15(1), 61–82 (1995)
6. Lukasiewicz, T., Straccia, U.: Managing uncertainty and vagueness in description logics for the semantic web. Journal of Web Semantics 6(4), 291–308 (2008)
7. Nilsson, N.J.: Probabilistic logic. Artificial Intelligence 28(1), 71–87 (1986)
8. Patel-Schneider, P, F., Horrocks, I., Bechhofer, S.: Tutorial on OWL (2003), http://www.cs.man.ac.uk/ horrocks/ISWC2003/Tutorial/
9. Poole, D.: Abducing through negation as failure: stable models within the independent choice logic. Journal of Logic Programming 44(1-3), 5–35 (2000)
10. Predoiu, L., Stuckenschmidt, H.: Probabilistic models for the semantic web: A survey. In: The Semantic Web for Knowledge and Data Management: Technologies and Practices. IGI Global (2008)

11. Sato, T.: A statistical learning method for logic programs with distribution semantics. In: International Conference on Logic Programming. pp. 715–729. MIT Press (1995)
12. Sirin, E., Parsia, B., Cuenca-Grau, B., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Journal of Web Semantics 5(2), 51–53 (2007)
13. URW3-XG: Uncertainty reasoning for the World Wide Web, final report (2005)