Providing an Uncertainty Reasoning Service for Semantic Web Application

Lei Li[1], Qiaoling Liu[1], Yunfeng Tao[2], Lei Zhang[1], Jian Zhou[1], and Yong Yu[1]

[1]APEX Data and Knowledge Management Lab
[2] Basics Lab,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, 200240, China
{lilei, lql, zhanglei, priest, yyu}@apex.sjtu.edu.cn, tyf@cs.sjtu.edu.cn

Abstract. In the semantic web context,the formal representation of knowledge is not resourceful while the informal one with uncertainty prevails. In order to provide an uncertainty reasoning service for semantic web applications, we propose a probabilistic extension of Description Logic, namely Probabilistic Description Logic Program (PDLP). In this paper, we introduce the syntax and intensional semantics of PDLP, and present a fast reasoning algorithm making use of Logic Programming techniques. This extension is expressive, lightweight, and intuitive. Based on this extension, we implement a PDLP reasoner, and apply it into practical use: Tourism Ontology Uncertainty Reasoning system (TOUR). The TOUR system uses PDLP reasoner to make favorite travel plans on top of an integrated tourism ontology, which describes travel cites and services with their evaluation.

1 Introduction

The Semantic Web (SW) [1] aims at transforming traditional text-based web and providing machine readable information for practical applications. Such information is based on three types of semantics: the implicit, the formal, and the powerful semantics[2]. The implicit semantics describes informal and uncertain information of the web, such as semantics in unstructured texts and document links. While the formal semantics often refers to well-defined and structured representation with definite meaning, such as ontologies upon Description Logic (DL). The last but most powerful semantics owns abilities of both of the previous ones, i.e. it describes both the informal (imprecise and probabilistic) and formal aspects of the web. However, it is not easy to obtain a powerful semantics due to the incompatibility between logic and probability. To resolve this incompatibility, we propose PDLP to tightly combine description logic with probability and provide a powerful semantics semantic web context.

As a formal representation, Description Logic[3] is a decidable fragment of First Order Logic (FO). With DL, Knowledge Bases (KB) describe concepts, roles (the relationship between concepts), axioms and assertions under a Tarski-like semantics. Efficient algorithms have been devised to solve reasoning tasks in

X. Zhou et al. (Eds.): APWeb 2006, LNCS 3841, pp. 628–639, 2006.

DL, such as highly optimized Tableau algorithms[4]. However, these algorithms do not deal with uncertainty.

In the recent decades, uncertainty starts to play an important role in order to supplement the expressivity of formal approaches. The changing web context and intrinsic properties of information require this type of uncertainty, as one example showed in our tourism ontology. However, pure DL does not lend a hand to uncertainty reasoning. Several extensions have been made to combine uncertainty and DL together[5,6,7]. In our attempt, we present Probabilistic Description Logic Programs (PDLP), a lightweight probabilistic extension to assertional knowledge in DL, and interpret probabilities in intentional semantics. Queries of PDLP are answered using a translational approach: reducing DL to logic programs in the same spirit as in [8].

The major differences between our PDLP and other related formalisms lie in the following aspects:

– PDLP only attaches probability to world assertions rather than terminological axioms;
– The syntax and semantics are carefully devised to meet both the DL and LP restriction;
– PDLP adopts a translational approach rather than a hybrid approach, such as [5,7].

As a result of these differences, PDLP achieves several highlights as follows:

– Expressive, PDLP owns the ability to deal with uncertainty, besides, the queries for PDLP could be both DL-like and LP-like thus PDLP provides a expressivity extension to queries;
– Lightweight, probability is only extended to world assertions rather than the whole knowledge base;
– Intuitive, the semantics of probability in PDLP is clear and natural;
– Speedy, its reasoning algorithm is very fast thanks to the efficient inferencing with logic programming techniques;
– Practical, we apply our implemented PDLP reasoner to Tourism Ontology Uncertainty Reasoning system to evaluate and rank travel plans for users, according to quality of service of travel cites.

The remaining part of this paper is organized as follows: in next section, we will formalize the syntax and semantics of PDLP. In section 3, we describe the reasoning tasks and the corresponding algorithm. In section 4, we present our PDLP implementation and its application in tourism planning. Related work is discussed in section 5. Finally, section 6 concludes the paper.

2 Syntax and Semantics

2.1 Syntax of PDLP

The language of PDLP is obtained by tailoring DL in the same essence as DHL[8]. The tailoring is justifiable due to the different expressivity and complexity between DL and LP, thus it is necessary to build up a model to capture common abilities of the two. Primarily, our extension enhances the expressivity of uncertain knowledge.

A probabilistic knowledge base of PDLP consists of two components: $pKB := \langle T, pA \rangle$, where TBox T contains *axioms* about concepts (in other words, the relationship between concepts):

$$C_h^1 \equiv C_h^2 \quad (concept\ equivalence) \qquad R \sqsubseteq S \quad (role\ hierarchy))$$
$$C_b \sqsubseteq C_h \quad (concept\ inclusion) \qquad R \in \mathcal{R}_+ \quad (transitive\ role)$$
$$R \equiv S^- \quad (inverse\ role)$$

where C_h, C_b are concepts and R, S are roles, defined as follows:

$$C_h \rightarrow A \mid C_h \sqcap C_h \mid \forall R.C_h \qquad C_b \rightarrow A \mid \neg A \mid C_b \sqcup C_b \mid C_b \sqcap C_b \mid \exists R.C_b$$

where A is an *atomic concept* and R is a *role*. Here, negation is allowed on *primitive concepts* rather than arbitrary ones to meet the translating restriction from DL to LP because LP cannot represent arbitrary negation.

The ABox of PDLP, pA, differs from that of normal DLs in the uncertainty it asserts. An probabilistic assertion $p \sim \varphi$ could assert uncertainty beyond the ability of ordinary assertion φ, where φ is $a{:}C$ or $\langle a, b \rangle{:}R$. A probabilistic ABox contains following assertions:

$$a{:}C_h, \quad \langle a, b \rangle{:}S, \quad p \sim a{:}A, \quad p \sim \langle a, b \rangle{:}R,$$

where a, b are individuals, A a *primitive concept*, S a role and R a *primitive role*[1], and p, called asserted probability (AP), is a real number in the range of $[0, 1]$. The first two assertions are deterministic, while the latter two are attached with uncertainty on primitive concepts or roles (thus it is lightweight).

2.2 Semantics

In PDLP, an assertion is attached with a probability to indicate how much likely it is consistent with respect to probabilistic knowledge base. In order to assign probability to an assertion, we first recall some notations from Probability Theory.

An interpretation of TBox is called a model of T, written as $I \models T$, where I assigns to every concept C a set $C^I \subseteq \triangle^I$ and to every role R a binary relation $R^I \subseteq \triangle^I \times \triangle^I$, where $\triangle^I$ is the domain of the interpretation (Table 1). The partial ordering $\preccurlyeq$ on interpretations is defined as $I_1 \preccurlyeq I_2$ if $C^{I_1} \subseteq C^{I_2}$ for every concept C and $R^{I_1} \subseteq R^{I_2}$ for every role R. $I_1 \prec I_2$ if $I_1 \preccurlyeq I_2$ and $I_1 \neq I_2$.

[1] *Primitive concepts* refer to atomic concepts with none occurrence in heads C_h of TBox axioms, primitive role is similar.

Table 1. Semantics of concept constructors

Constructor name	Syntax	Semantics
Atomic Concept	A	$A^I \subseteq \triangle^I$
Atomic Negation	$\neg A$	$\triangle^I \setminus A^I$
Role	R	$R^I \subseteq \triangle^I \times \triangle^I$
Conjunction	$C \sqcap D$	$C^I \cap D^I$
Disjunction	$C \sqcup D$	$C^I \cup D^I$
Exists restriction	$\exists R.C$	$\{x \mid \exists y.\langle x, y\rangle \in R^I \wedge y \in C^I\}$
Value restriction	$\forall R.C$	$x \mid \forall y.\langle x, y\rangle \in R^I \rightarrow y \in C^I$

Definition 2.1. *(Least Fixed Point Semantics) Let I_0 be a* base interpretation *only on primitive concepts and primitive roles, an extension I on a* base interpretation I_0 *is called a* least fixed point model *with respect to a TBox T if: (1) $I_0 \preccurlyeq I, I \models T$; (2) $\forall I'.I_0 \preccurlyeq I' \wedge I' \models T \Longrightarrow I \preccurlyeq I'$.*

Hence, we define the sample space in PDLP as the model class $\mathfrak{C}$: the collection of least fixed point models with respect to TBox. We assume that μ is a probability distribution on $\mathfrak{C}$ with restriction $\mu(\mathfrak{C}) = 1$.

Definition 2.2. *A pair $\langle \mathfrak{C}, \mu\rangle$ is called a* probabilistic world *of the knowledge base pKB, where $\mathfrak{C}$ and μ follow conditions mentioned above.*

Intensional Semantics. An ABox assertion is in fact a logical formula. A formula has its satisfied models: $Mod(\varphi) := \{I : I \in \mathfrak{C} \wedge I \models \varphi\}$.

Definition 2.3. *The calculated φ probability (CP) of a classical deterministic assertion is a function defined by:*

$$\nu(\varphi) := \mu(Mod(\varphi)) = \sum_{I \in \mathfrak{C}, I \models \varphi} \mu(I)$$

where φ is a deterministic assertion (a:C or $\langle a, b\rangle$:R), $\langle \mathfrak{C}, \mu\rangle$ is a probabilistic world. $\sum$ is well defined because $\mathfrak{C}$ is enumerable.

Lemma 2.1 *The CP of an assertion has following properties:*

- $\nu(a{:}\neg C) = 1 - \nu(a{:}C)$
- *Inclusion-exclusion principle:* $\nu(a{:}C \sqcup D) = \nu(a{:}C) + \nu(a{:}D) - \nu(a{:}C \sqcap D)$

where C, D are concepts.

Definition 2.4. *A probabilistic world $\langle \mathfrak{C}, \mu\rangle$ satisfies a probabilistic assertion $p \sim \varphi$, written as $\langle \mathfrak{C}, \mu\rangle \models\!\!\approx p \sim \varphi$, if $\nu(\varphi) = p$ (CP equals AP).*

A probabilistic world satisfies ABox pA with respect to a TBox T, $\langle \mathfrak{C}, \mu\rangle \models\!\!\approx pA$, if it satisfies all assertions in pA. In this sense, it is also written as $\langle \mathfrak{C}, \mu\rangle \models\!\!\approx pKB$, for a probabilistic Knowledge Base pKB constituted by T and pA.

Definition 2.5. *A pKB entails an assertion $p \sim \varphi$, $pKB \approx p \sim \varphi$, if all probabilistic worlds of pKB, satisfy the assertion.*

$$pKB \approx p \sim \varphi \quad \textit{iff} \quad \forall \langle \mathfrak{C}, \mu \rangle \approx pKB \implies \langle \mathfrak{C}, \mu \rangle \approx p \sim \varphi$$

Lemma 2.2 *Probability Indicator*

- *if $pKB \approx p \sim a{:}C$, then $pKB \approx (1-p) \sim a{:}\neg C$;*
- *inclusion-exclusion principle: if $pKB \approx p \sim a{:}C$, $pKB \approx q \sim a{:}D$ and $pKB \approx r \sim a{:}C \sqcap D$, then $pKB \approx (p+q-r) \sim a{:}C \sqcup D$;*
- *$a{:}\exists R.C \equiv \bigvee_{b \in HU} (\langle a, b\rangle{:}R \wedge b{:}C)$, where HU denotes the Herbrand Universe. This provides an approach to calculate existential assertions.*

The lemma above is vitally important because it is the basis of our calculation of the probability of a given assertion. And this also explains why we restrict the syntax of probabilistic assertion in ABox: we would like to compute the probability of complex assertions from some basic and simple facts. This is rather useful in practical application as shown in our example: the ontology knows basic facts while the reasoner can figure out complex events.

Our extension enhances the expressivity of uncertain knowledge. Meanwhile it is its lightweight extension that enables an easy semantic model of probabilistic knowledge base. The overall semantics is rather intuitive once we set up probabilities for models of the knowledge base, which inspires our fast reasoning scheme in the following section.

3 Reasoning

3.1 Reasoning Tasks

A DL reasoning system typically supports several kinds of reasoning tasks: membership, subsumption, satisfiability and hierarchy, all of which can be reduced into retrieval problems [3]. While LP engines could typically answer two kinds of queries: instance retrieval and membership check[8]. Since PDLP adopts the translational approach, it supports similar queries as DHL[8], which enables us to express information need either DL-like with concept constructors or LP-like with variables. For example, we can represent our query "retrieve any instance of $\exists R.C$" as:

- DL-like: $\exists R.C$
- LP-like: $Query(x) \leftarrow R(x, y), C(y)$

The DL-like queries can be easily translated into LP-like queries, while LP queries can express more than DL ones. Therefore PDLP is expressive in query ability.

The query is answered in the following scheme:

1. Fast retrieval of any possible instances by making use of the below translation;
2. Calculate the probability corresponding to each instance (pair).

3.2 Translation

In order to retrieve all possible results, a probabilistic knowledge base pKB can be partially translated into a logic program[9] while preserving the semantics. We follow the approach of [8] and define a mapping from PDLP to LP in the same way as DHL except for atomic negations and probability assertions as follows:

$\Gamma(A, x)$	$\longrightarrow$	$A(x)$
$\Gamma(\neg A, x)$	$\longrightarrow$	$\sim A(x)$
$\Gamma(C^1 \sqcap C^2, x)$	$\longrightarrow$	$\Gamma(C^1, x) \wedge \Gamma(C^2, x)$
$\Gamma(C_b^1 \sqcup C_b^2, x)$	$\longrightarrow$	$\Gamma(C_b^1, x) \vee \Gamma(C_b^2, x)$
$\Gamma(\forall R.C_h, x)$	$\longrightarrow$	$\Gamma(C_h, y) \leftarrow R(x, y)$
$\Gamma(\exists R.C_b, x)$	$\longrightarrow$	$R(x, y) \wedge \Gamma(C_b, y)$
$\Gamma(C_b \sqsubseteq C_h)$	$\longrightarrow$	$\Gamma(C_h, x) \leftarrow \Gamma(C_b, x)$
$\Gamma(C_h^1 \equiv C_h^2)$	$\longrightarrow$	$\begin{cases} \Gamma(C_h^1 \sqsubseteq C_h^2) \\ \Gamma(C_h^2 \sqsubseteq C_h^1) \end{cases}$
$\Gamma(R \sqsubseteq S)$	$\longrightarrow$	$S(x, y) \leftarrow R(x, y)$
$\Gamma(R \in R_+)$	$\longrightarrow$	$R(x, y) \leftarrow R(x, z) \wedge R(z, y)$
$\Gamma(R \equiv S^-)$	$\longrightarrow$	$\begin{cases} R(x, y) \leftarrow S(y, x) \\ S(x, y) \leftarrow R(y, x) \end{cases}$
$\Gamma(a : C_h)$	$\longrightarrow$	$\Gamma(C_h, a)$
$\Gamma(\langle a, b\rangle : R)$	$\longrightarrow$	$R(a, b)$
$\Gamma(p \sim a : A)$	$\longrightarrow$	$A(a)$
$\Gamma(p \sim \langle a, b\rangle : R)$	$\longrightarrow$	$R(a, b)$

A (retrieval) query could also be translated into LP conventions as the body part of a rule without head.

This translation phase does not concern probability, with its primary target on all possible result. The preservation of semantics relies primarily on translation of deterministic part of pKB, which is ensured and by the common Least Fixed Point Semantics these two formal frameworks share[10]. Therefore the semantics preserve, for the probability distribution is on the models of TBox, which concerns no uncertainty.

3.3 Probabilistic Inferencing

Definition 3.1. *A set of primitive assertions $\mathcal{E}$ is called a (basic)* evidence *of an assertion $p\ \varphi$ if*

- $\langle T, \mathcal{D}(\mathcal{E})\rangle \models \mathcal{D}(p \sim \varphi)$, *where* $\langle T, \mathcal{D}(\mathcal{E})\rangle$ *is a temporal knowledge base.*
- $\forall \mathcal{E}' \subsetneq \mathcal{E}$, $\langle T, \mathcal{D}(\mathcal{E}')\rangle \not\models \mathcal{D}(p \sim \varphi)$

where $\mathcal{D}(\mathcal{E}) = \{\varphi : p \sim \varphi \in \mathcal{E}\}$. *$\mathcal{E}$ is a minimum set of primitive facts to support φ.*

Assuming the consistency of pKB, an answer of a query under the least fixed point semantics is a result of bottom-up calculation while the evidence here in essence is a result of top-down search of supporting facts. Hence, the procedure to calculate the probability of an assertion can be summarized as follows:

1. Inference all basic evidences $\mathcal{E}_1 \cdots \mathcal{E}_k$ by LP engine
2. Calculate the probability by $CP = CP(\bigvee_{i=1}^{k} \wedge \mathcal{E}_i)$ by inclusion-exclusion principle introduced in lemma 2.2.

In order to calculate probability, one vital property we assume here is the independence of assertions in a basic evidence, that is, $CP(p \sim \varphi \wedge q \sim \psi) = AP(p \sim \varphi) \cdot AP(q \sim \psi) = p \cdot q$, where φ, ψ are $a : A$ or $\langle a, b \rangle : R$, and A,R are primitive.

4 Implementation and Application

In this section, we present an example application of our implemented PDLP reasoner, the Tourism Ontology Uncertainty Reasoning system(TOUR), to provide a travel planning service for clients.

4.1 Implementation

We implement a PDLP reasoner based on intensional semantics. In order to speed up its reasoning, PDLP reasoner adopts following optimizing techniques: sideway information passing, magic set and semi-naive evaluation strategies[11]. The PDLP reasoning speed is really fast in the following practical tourism application.

4.2 Scenario and Architecture

The TOUR system is built upon the tourism ontology (adapted from Protege[2] ontology library). The system aims at making a tourism plan most conforming to customer's expectation. This system contains three components (Figure 1):

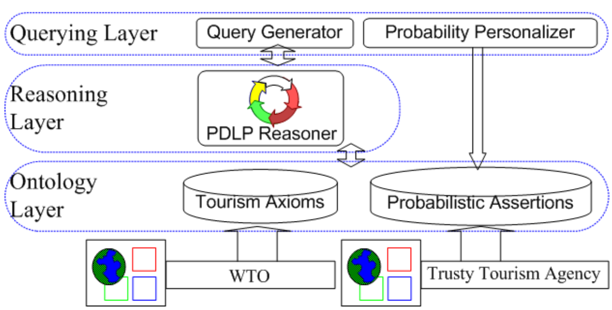

Fig. 1. Architecture of TOUR

– Ontology layer: The tourism ontology formally describes a set of available destinations, accommodation, and activities. Tourism axioms (see Table 2)of

[2] http://protege.stanford.edu/

this ontology, are devised according to the WTO thesaurus[3]. The ontology grades each instance with a score. These grades of travel cites and services come from two sources: trusty tourism agencies like National Tourism Administration (CNTA)[4], and customer personalized favorite setup;
- Reasoning layer: make use of our implemented PDLP reasoner;
- Querying layer: transform user queries to DL-style or LP-style forms, and put customer personalized assertions about travel cites and services into ontology.

Table 2. A fragment of the translated LP of axioms from tourism ontology

Axioms	Translated rules
RuralArea $\sqsubseteq$PreferredDest	PreferredDest(X) $\leftarrow$RuralArea(X).
UrbanArea $\sqsubseteq$PreferredDest	PreferredDest(X) $\leftarrow$UrbanArea(X).
PreferredPark $\sqsubseteq$RuralArea	RuralArea(X) $\leftarrow$PreferredPark(X).
PreferredFarm $\sqsubseteq$RuralArea	RuralArea(X) $\leftarrow$PreferredFarm(X).
PreferredTown $\sqsubseteq$UrbanArea	UrbanArea(X) $\leftarrow$PreferredTown(X).
PreferredCity $\sqsubseteq$UrbanArea	UrbanArea(X) $\leftarrow$PreferredCity(X).
hasPart $\in \mathcal{R}_+$	hasPart(X,Z) $\leftarrow$hasPart(X,Y),hasPart(Y,Z).
Sports $\sqsubseteq$Activity	Activity(X) $\leftarrow$Sports(X).
Adventure $\sqsubseteq$Activity	Activity(X) $\leftarrow$Adventure(X).
Sightseeing $\sqsubseteq$Activity	Activity(X) $\leftarrow$Sightseeing(X).
Hotel $\sqsubseteq$Accommodation	Accommodation(X) $\leftarrow$Hotel(X).
LuxuryHotel $\sqsubseteq$Hotel	Hotel(X) $\leftarrow$LuxuryHotel(X).
offerActivity $\equiv$isOfferedAt$^-$	offerActivity(X,Y) $\leftarrow$isOfferedAt(Y,X).
	isOfferedAt(X,Y) $\leftarrow$offerActivity(Y,X).

Probabilities in the tourism ontology have practical meanings about quality of a tourism cites and services:

- $p \sim a$:Accommodation denotes accommodation rating given by CNTA;
- $p \sim \langle d, a\rangle$:hasAccommodation indicates the convenience of accommodation a in destination d, e.g. the environment and traffic conditions in neighborhood;
- $p \sim \langle d, c\rangle$:hasActivity describes the service probability of activity c in destination d, e.g. activities such as watching sun rising should be in clean days.

4.3 Querying on the Ontology

The TOUR system takes the following procedures to evaluate a travel plan for a customer:

1. Setup probabilities for basic facts (assertions) in ontology (Probability Personalizer);
2. Specify the factors and rules to retrieve travel plans (Query Generator);

[3] The World Tourism Organization, http://www.world-tourism.org
[4] http://www.cnta.gov.cn/

3. Make use of PDLP reasoner to infer possible travel plans with their probability (PDLP reasoner).

The following example concerning travelling in Beijing illustrates the whole procedure.

In the first phase, suppose a customer grades travel cites and services in Beijing (Figure 2) as follows (partially):

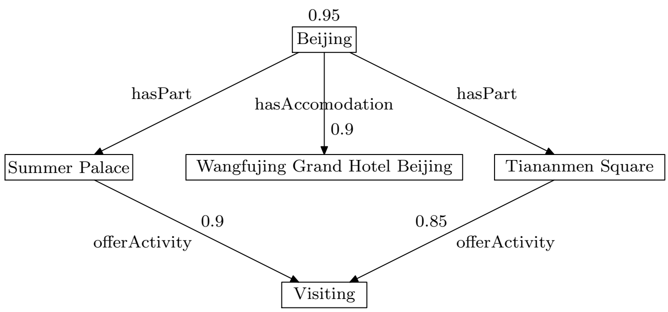

Fig. 2. Assertions about Beijing

$$\begin{array}{lll} 0.95 & \sim \text{Beijing:PreferredCity} & (1)\\ 0.9 & \sim \langle\text{Beijing,Wangfujing\_Grand\_Hotel}\rangle\text{:hasAccommodation} & (2)\\ 0.9 & \sim \langle\text{Summer\_Palace,Visiting}\rangle\text{:offerActivity} & (3)\\ 0.85 & \sim \langle\text{Tiananmen\_Square,Visiting}\rangle\text{:offerActivity} & (4)\\ & \langle\text{Beijing,Summer\_Palace}\rangle\text{:hasPart} & (5)\\ & \langle\text{Beijing,Tiananmen\_Square}\rangle\text{:hasPart} & (6) \end{array}$$

Second, the optimal travel plan is specified as a triple of destination, accommodation and activity, judged by a combination of their service quality and convenience. Thus the query rule is:

Q1: Query(X,Y,Z)← PreferredDest(X),hasAccommodation(X,Y),
hasPart(X,X1),offerActivity(X1,Z).

In the third phase, the PDLP engine infers a result $\text{RES}_1 = \langle$Beijing, Wangfujing_Grand_Hotel, Visiting$\rangle$ with two evidences: $\mathcal{E}_1$ ={(1),(2),(3),(5)} and $\mathcal{E}_2$ = {(1),(2),(4),(6)}, thus the convenience (probability) of this plan is $\text{CP}(\text{RES}_1)$= $\text{CP}(\mathcal{E}_1)\text{V}+\text{CP}(\mathcal{E}_2)-\text{CP}(\mathcal{E}_1 \cdot \mathcal{E}_2)$=0.842175.

Figure 3 shows results of Q1.

4.4 Performance

We have tested performance of the TOUR system using two sorts of queries. One is a mixed query combining both probability and formal inquiry mentioned above. Another is intended to test capacity of the system on simple queries:
Q2: Query(X,Y)←offerActivity(X,Y).

Fig. 3. Partial result of Q1

	Ontology	Tour1	Tour2	Tour3	Tour4	Tour5
	Ontology Size	489	1002	1524	1998	3099
Q1	Result size	296	638	986	1302	2036
	Time Cost(s)	0.94	2.92	8.30	18.64	54.66
Q2	Result size	15	15	15	15	15
	Time Cost(s)	0.19	0.14	0.16	0.17	0.19

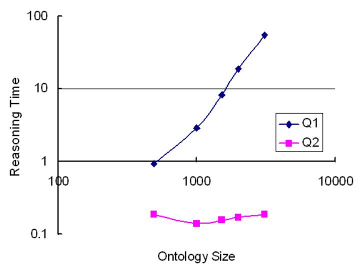

Fig. 4. Performance of TOUR

Figure 4 shows the performance of TOUR system of two queries on two ontologies. The size of ontology is measured in instance number(both concept instances and role instances). In the logarithmic graph, the reasoning time is in scale to the ontology size, illustrating the attractive computability of PDLP. Theoretically, PDLP compute query answers in fast thanks to the tractable complexity of LP[9]. Because PDLP reasoner is not dedicated to TOUR system, we could also expect high performance of PDLP's reasoning in general applications.

5 Related Work

Previously there are several related approaches to probabilistic description logics which can be classified according to: (1) what component the uncertainty be attached to (1a) TBox[12,13,14] (1b) ABox[7,15] (1c) both[16,6,5,17,18,19,20]; (2) what approach is applied to reasoning. For the latter aspect, (2a) fuzzy logic inferencing[16,17], (2b) Bayesian Network[12,18], (2c) lattice-based approach[20], (2d) combination of probabilistic DL with LP[6,7,5].

[16,17] extend DL (both TBox and ABox) with uncertainty interval by fuzzy set theory, and devise a set of reasoning rules to inference uncertainty. [12,18] translate probabilistic extension of DL into Bayesian Network approach with different expressivities. The former [12] works based on extension to ALC TBox while the latter [18] makes a probabilistic extension to OWL ontology. [20] manages uncertainty in DL with lattice-based approach mapping an assertion to a uncertain value in a lattice and reasoning in a tableaux-like calculus. Other related work [13] concentrates on probabilities on terminological axioms, [15] on world assertions while [14] on concept subsumption and role quantification. [19] extends SHOQ(D) using probabilistic lexicography entailment and supports assertional knowledge.

Concerning extension of uncertainty to combination of DL and LP, the works most related to ours can be divided into: (i) hybrid approaches tightly combining DL with LP and adding uncertainty in order to extend expressivity [5][7]; (ii) translational approaches reducing DL with uncertainty to probabilistic inferencing in LP in order to take advantage of powerful logic programming technology for inference [6]. [5] adds an uncertainty interval to an assertion in a combination of DL and LP under Answer Set Semantics. [7] presents combination of description logic programs (or dl-programs) and adds probability to assertions under the answer set semantics and the well-founded semantics, and reduces computation of probability to solving linear optimization systems. [6] generalizes DAML+OIL with probability (in essence both on TBox and ABox), and maps it to four-valued probabilistic datalog. Besides difference in four-valued extension, other difference between [6] and our PDLP lies in that [6] attaches probability to both axioms and assertions in DAML+OIL while we restrict probability only on ABox assertions in order to achieve our three highlights, especially intuitive in semantical aspect.

6 Conclusion and Future Work

In this paper, we have extended DL with probability on assertional knowledge, namely PDLP, and interpreted probabilistic ABox assertions under intensional semantics. The syntax and semantics of PDLP are very lightweight, intuitive and expressive to deal with uncertainty and practical applications in Semantic Web, and its reasoning is very fast through LP techniques. We have implemented a PDLP reasoner and apply it into a practical application TOUR system to make optimal travel plans for users. The performance of the TOUR system is encouraging for future use of PDLP in other applications.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic Web. Scientific American **284** (2001) 34–43
2. Sheth, A.P., Ramakrishnan, C., Thomas, C.: Semantics for the semantic web: The implicit, the formal and the powerful. Int. J. Semantic Web Inf. Syst. **1** (2005) 1–18

3. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press (2003)
4. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for very expressive description logics. Logic J. of the IGPL **8** (2000)
5. Straccia, U.: Uncertainty and description logic programs: A proposal for expressing rules and uncertainty on top of ontologies. (2004)
6. Nottelmann, H., Fuhr, N.: pDAML+OIL: A probabilistic extension to DAML+OIL based on probabilistic Datalog. In: IPMU-04. (2004)
7. Lukasiewicz, T.: Probabilistic description logic programs. In: Proc. of the 8th Euro. Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Springer (2005) 737–749
8. Grosof, B.N., Horrocks, I., Volz, R., Decker, S.: Description logic programs: combining logic programs with description logic. In: WWW '03, ACM Press (2003) 48–57
9. Dantsin, E., Eiter, T., Gottlob, G., Voronkov, A.: Complexity and expressive power of logic programming. ACM Comput. Surv. **33** (2001) 374–425
10. Grädel, E.: Finite model theory and descriptive complexity. In: Finite Model Theory and Its Applications. Springer-Verlag (2003)
11. Ramakrishnan, R., Srivastava, D., Sudarshan, S.: Efficient bottom-up evaluation of logic programs. In Vandewalle, J., ed.: The State of the Art in Computer Systems and Software Engineering. Kluwer Academic Publishers (1992)
12. Koller, D., Levy, A.Y., Pfeffer, A.: P-classic: A tractable probablistic description logic. In: AAAI/IAAI. (1997) 390–397
13. Heinsohn, J.: Probabilistic description logics. In: Proc. of UAI-94. (1994) 311–318
14. Jaeger, M.: Probabilistic role models and the guarded fragment. In: IPMU-04. (2004)
15. Duerig, M., Studer, T.: Probabilistic abox reasoning: Preliminary results. In: DL-05, Edinburgh, Scotland (2005)
16. Straccia, U.: Towards a fuzzy description logic for the semantic web (preliminary report). In: ESWC-05, Springer Verlag (2005) 167–181
17. Straccia, U.: Reasoning within fuzzy description logics. JAIR **14** (2001) 137–166
18. Ding, Z., Peng, Y., Pan, R.: A Bayesian Approach to Uncertainty Modeling in OWL Ontology. In: Proc. of Int. Conf. on Advances in Intelligent Systems - Theory and Applications. (2004)
19. Giugno, R., Lukasiewicz, T.: P-shoq(d): A probabilistic extension of shoq(d) for probabilistic ontologies in the semantic web. In: Logics in Artificial Intelligence, European Conference,, Springer (2002) 86–97
20. Straccia, U.: Uncertainty in description logics: a lattice-based approach. In: IPMU-04. (2004) 251–258