Probabilistic RDF

Octavian Udrea[1] V.S. Subrahmanian[1]
Zoran Majkić[2]
[1]University of Maryland, College Park, MD 20742,{udrea,vs}@cs.umd.edu
[2]Universita di Roma "La Sapienza", Via Salaria, 113 I-00198 Rome, Italy, majkic@dis.uniroma1.it*

Abstract

RDF is rapidly being adopted around the world as a paradigm for knowledge representation. However, we are not aware of any version of RDF that can express probabilistic knowledge. In this paper, we develop a framework called Probabilistic RDF *(pRDF for short) – we provide a syntax as well as a model theoretic and fixpoint semantics for acyclic pRDF theories. We then provide algorithms to efficiently answer queries posed to pRDF ontologies. We have developed a prototype implementation that shows that our algorithms work very well in practice.*

1 Introduction

Over the last few years, the use of RDF as a paradigm for representing knowledge has grown dramatically. RDF and OWL ontologies exist on a wide variety of topics ranging from genetics to visual sensor data fusion. These are clearly domains that are chock full of uncertainty - image processing programs based on Bayesian analysis often return probabilistic identifications of objects, while relationships between a symptom or disease and the genetic markers a person may also be probabilistic.

In order to express such information, we introduce *Probabilistic RDF* (pRDF) for short. We define the concept of a pRDF schema and a pRDF instance in Section 2. A pRDF instance extends RDF triples by allowing *unconditioned* probability distributions over a set of possible values of an RDF triple. Section 3 provides a formal model theoretic semantics for pRDF based on the possible worlds probabilistic logics of Fagin et. al. [3]. Section 4 shows that we can associate a monotonic function with any pRDF theory — this function has a least fixpoint that compactly represents the set of all quadruples entailed by the pRDF theory. However, using the fixpoint to answer queries is not always desirable because the size of the fixpoint can be enormous. Section 5 provides algorithms to efficiently answer atomic queries with at most one variable. Section 6 describes our prototype implementation together with experiments showing that queries can be answered in very small amounts of time (a few seconds) for pRDF instances as large as 100,000 quadruples.

*Work performed while at the University of Maryland College Park

2 pRDF syntax

We now develop a formal syntax for pRDF. We assume the existence of the following sets: $\mathcal{U}$ is the set of URI references and $\mathcal{L}$ is the set of literals (primitive data values)[1]. We also assume the following are arbitrary but fixed:

(i) $\mathcal{C} \subseteq \mathcal{U}$ is the set of classes.
(ii) $\mathcal{P} \subseteq \mathcal{U}$ is the set of properties. We introduce the notion of *transitive properties* as a basic inference capability for RDF instance data. We assume that a set $\mathcal{P}^t \subseteq \mathcal{P}$ of *transitive* properties is specified, where $\mathcal{P}^t \cup \mathcal{P}^{nt} = \mathcal{P}$ and $\mathcal{P}^t \cap \mathcal{P}^{nt} = \emptyset$.
(iii) $\mathcal{I} \subseteq \mathcal{U}$ is a set of instances (individuals).

We now define a pRDF schema.

Definition 1 (pRDF schema) *A* probabilistic RDF schema *is a finite set S of elements in one of the following forms:*

(S1) Probabilistic quadruples *are of the form $(s, rdfs: subClassOf, O, \delta)$, where $s \in \mathcal{C}, O \subseteq \mathcal{C}$ and $\delta : O \to (0,1]$ a probability distribution over O. We require that:*

(S1.1) $\sum_{v \in O} \delta(v) \leq 1$.

[1]For reasons of space, we do not address features such as reification in this paper; blank nodes are therefore omitted from the discussion.

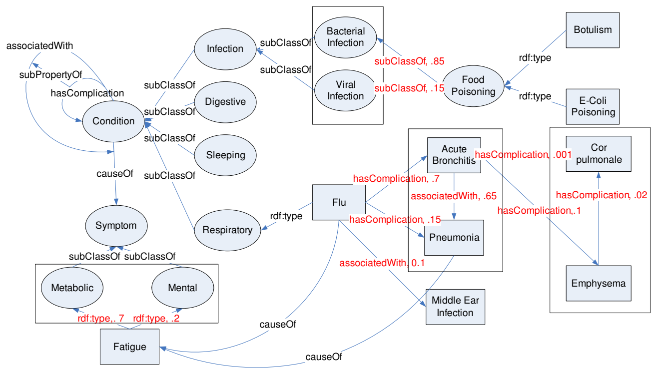


Figure 1. pRDF **example**

(S1.2) $\forall$ $((s, rdfs : subClassOf, O_1, \delta_1), (s, rdfs : subClassOf, O_2, \delta_2) \in S$ *such that* $O_1 \neq O_2)$ *then* $O_1 \cap O_2 = \emptyset$.

(S2) Non-probabilistic triples *are in one of the following forms:*

(S2.1) $(p_1, rdfs : subPropertyOf, p_2)$, *where* $p_1, p_2 \in \mathcal{P}$.
(S2.2) $(p, rdfs : range, c)$, *where* $p \in \mathcal{P}, c \in \mathcal{C}$.
(S2.2) $(p, rdfs : domain, c)$, *where* $p \in \mathcal{P}, c \in \mathcal{C}$.

Example 1 *Figure 1 contains a graph representation*[2] *of a pRDF schema (ignore the rectangular nodes for now). (FoodPoisoning, rdfs:subClassOf, {BacterialInfection, ViralInfection,}, {.85,.15}) is a pRDF schema quadruple illustrating item (S1) in Definition 1; the quadruple states that a FoodPoisoning condition is a BacterialInfection with 85% probability and a ViralInfection with 15% probability. (hasComplication,rdfs:subPropertyOf, associatedWith) is a non-probabilistic triple*[3] *illustrating item (S2.1) of the same definition.*

The rest of the paper will primarily focus on the semantics and query processing at the pRDF instance level; the problem of pRDF schema queries will be addressed in an extended version of this paper.

Definition 2 (pRDF instance) *A* probabilistic RDF instance R *is a finite set of probabilistic quadruples in one of the following forms:*

(I1) (s, p, V, δ), *where* $s \in \mathcal{I}$, $p \in \mathcal{P}$, $V \subseteq \mathcal{I} \cup \mathcal{L}$, $\delta : V \rightarrow (0, 1]$ *such that:*

(I1.1) $\sum_{v \in V} \delta(v) \leq 1$.
(I1.2) $\forall$ $(s, p, V_1, \delta_1), (s, p, V_2, \delta_2) \in R$ *such that* $V_1 \neq V_2$, $V_1 \cap V_2 = \emptyset$.

(I2) $(s, rdf : type, V, \delta)$, *where* $s \in \mathcal{I}$, $V \subseteq \mathcal{C}$ *such that:*

(I2.1) $\sum_{v \in V} \delta(v) \leq 1$.
(I2.2) $(s, rdf : type, V_1, \delta_1), (s, rdf : type, V_2, \delta_2) \in R \Rightarrow (V_1 = V_2) \wedge (\delta_1 = \delta_2)$.

Example 2 *The quadruple (Flu, hasComplication, {AcuteBronchitis,Pneumonia},{.7, .15}) in Figure 1 can be intuitively read as: whenever the Flu condition has complications from {AcuteBronchitis, Pneumonia}, there is a 70% probability that the complication is AcuteBronchitis and a 15% probability that the complication is Pneumonia. Whenever probabilities are not specified in the example, they are assumed to be 1.*

We will often abuse notation and talk about quadruples in a pRDF instance having the form (s, p, V, δ) where p is either a property or $rdf : type$, with the restrictions imposed. This allows us to talk about both kinds of quadruples in a pRDF instance in a unified way. We also use $rdfs : subPropertyOf^*$ to denote the reflexive, transitive closure of the $rdfs : subPropertyOf$ relation.

Definition 3 (pRDF theory) *A pRDF theory is a pair* (S, R), *where* S *is a pRDF schema and* R *is a pRDF instance.*

We note that there are a number of ways in which pRDF theories can be represented in practice. One possible way is to use quadruples[4]; another possibility is the use of reification. Since pRDF semantics and query processing are the focus of this paper, we omit a lengthy discussion on representation issues.

We now define the important concept of a p-path.

Definition 4 (p-path) *Let* (S, R) *be a pRDF theory and let* $p \in \mathcal{P}^t$. *Let* P *be a sequence of quadruples* $(s_i, p_i, v_i, \gamma_i)$, *for* $i = 1, \ldots, n$, *where:*

(i) $\forall\ i \in [1, n], \exists\ (s_i, p_i, V, \delta) \in R$ *s.t.* $v_i \in V \wedge \delta(v_i) = \gamma_i$.
(ii) $\forall\ i \in [1, n], p_i\ rdfs : subPropertyOf^*\ p$.
(iii) $\forall\ i \in [1, n-1], v_i = s_{i+1}$.

[2] The example was compiled using information available at www.wrongdiagnosis.com Probabilities were computed from a small subset (1980-1985) of statistical data publicly available from the National Health Interview Survey — www.cdc.gov/nchs/nhis.htm.

[3] While conditions may be *associated* in terms of diagnosis methods, treatment or symptoms, *hasComplication* directly specifies a cause-effect relationship.

[4] A quadruple-based approach is currently discussed for representing contexts/data provenance in RDF — see http://www.w3.org/2001/12/attributions/.

*In this case, P is a p-*path *of length n with origin s_1 and destination v_n: we denote P by $\langle s_i, p_i, v_i, \gamma_i \rangle_{i\in[1,n]}$. We use $org(P)$ and $dest(P)$ to denote s_1 and v_n respectively.*

Example 3 *Consider the pRDF instance in Figure 1. There are two distinct associatedWith-paths between Flu and Pneumonia — one resulting from the hasComplication edge between the two objects and the second containing the AcuteBronchitis node (hasComplication is a subproperty of associatedWith).*

A pRDF instance is *acyclic* iff for all properties $p \in \mathcal{P}^t$, there are no cyclic p-paths in it.

Assumption. Throughout this paper, we assume that all pRDF theories are acyclic[5].

We recall the notion of a triangular norm (first introduced in the context of fuzzy logic)[3] that is often used to compute the probability of a conjunction of events.

Definition 5 (t-norm) *A* triangular norm *(t-norm for short) is any binary function $\otimes$ from $[0,1] \times [0,1]$ to $[0,1]$ such that:*

(i) $\otimes$ *is associative and commutative;*
(ii) $\forall\ x, y, z, w \in [0,1]$ *s.t.* $x \leq y$ *and* $z \leq w$, $x \otimes z \leq y \otimes w$.
(iii) $0 \otimes x = 0$;
(iv) $1 \otimes x = x$.

We will also denote by $min_{\otimes}(p, p_t) = min(\{q | p \otimes q \geq p_t\})$. As a convention, if $\{q | p \otimes q \geq p_t\} = \emptyset$, then we write $min_{\otimes} = 1$. If can be easily proved that $\forall\ p, \forall\ p_t\ p_t \leq min_{\otimes}(p, p_t)$.

3 pRDF semantics

We now introduce a declarative model theoretic semantics for pRDF. Given a pRDF quadruple (s, p, V, δ), we know that each RDF-triple of the form (s, p, v) where $v \in V$ has a probability $\delta(v)$ of being true. For any set of such triples (or "world") obtained in this manner from a pRDF theory, we would like to associate a probability. We now define these terms.

Definition 6 (World) *A world W is a set of triples (s, p, v), such that $(s \in \mathcal{I} \wedge p \in \mathcal{P} \wedge v \in \mathcal{I} \cup \mathcal{L}) \vee (s \in \mathcal{I} \wedge p = rdf : type \wedge v \in \mathcal{C})$. We denote the set of possible worlds by $\mathcal{W}$.*

[5]This is a reasonable assumption – the main argument is that the expressiveness of RDF is lower than that of most description logic frameworks (which can deal with cyclic ABoxes). Space constraints prevent us from going through these details.

Definition 7 (pRDF interpretation) *A pRDF interpretation is a mapping $I : \mathcal{W} \to [0,1]$, such that $\sum_{W\in\mathcal{W}} I(W) = 1$.*

A pRDF interpretation postulates that exactly one possible world is the "real" world — uncertainty arises because we don't know which of the possible worlds is the real one. Our possible worlds approach builds on probabilistic logics due to Halpern[6].

Example 4 *Any subgraph of the graph in Figure 1 is a possible world. These are not all the possible worlds. Intuitively, we would assign a very low probability to a world such as W={(MiddleEarInfection,hasComplication,Emphysema)}, since there is no evidence for the information in W.*

Definition 8 (pRDF satisfaction) *An interpretation I* satisfies *a pRDF quadruple (s, p, V, δ) iff $\forall\ v \in V, \delta(v) \leq \sum_{\{W\in\mathcal{W} | (s,p,v)\in W\}} I(W)$. I satisfies a pRDF theory (S, R) w.r.t. t-norm $\otimes$ iff:*

(i) I satisfies every quadruple in R.
(ii) $\forall\ p-paths\ \langle s_i, p_i, v_i, \gamma_i \rangle_{i\in[1,n]}$ *in* (S, R), $\bigotimes_i \gamma_i \leq \sum_{\{W\in | (s_1,p,v_n)\in W\}} I(W)$.

We say that (S, R) is consistent *iff it has a satisfying interpretation.*

The second condition above uses $\otimes$ to compute the probability of a given p-path[6]. Entailment is defined in the usual way.

Definition 9 (Entailment) *Let $\otimes$ be an arbitrary but fixed t-norm. (S, R) entails (s, p, V, δ), written $(S, R) \models_{\otimes} (s, p, V, \delta)$ iff every satisfying interpretation of (S, R) also satisfies (s, p, V, δ). Furthermore, $(S_1, R_1) \models_{\otimes} (S_2, R_2)$ iff every satisfying interpretation of (S_1, R_1) is a satisfying interpretation of (S_2, R_2). (S_1, R_1) is equivalent to (S_2, R_2), written $(S_1, R_1) \equiv_{\otimes} (S_2, R_2)$ iff $(S_1, R_1) \models_{\otimes} (S_2, R_2)$ and $(S_2, R_2) \models_{\otimes} (S_1, R_1)$.*

Example 5 *The pRDF theory in Figure 1 entails (Flu, associatedWith, {Pneumonia},{.65}) w.r.t. the $min(x, y)$ t-norm. It also trivially entails (Flu, associatedWith, {AcuteBronchitis,Pneumonia},{0.6,0.15}).*

[6]We note that analogous concepts of interpretation and satisfaction can be defined for pRDF schemas (in the same way as RDF model theory defines RDF and RDFS interpretations). However, due to space constraints, we primarily focus on pRDF instances.

The fact that all pRDF theories are consistent can be easily shown by assigning $I(W_{max}) = 1$, where W_{max} is the "largest" possible world.

Theorem 1 *Every pRDF theory (S, R) is consistent w.r.t. any t-norm $\otimes$.*

4 pRDF: Fixpoint Semantics

In this section, we build upon Theorem 1 and show how to associate an operator with each pRDF theory that maps pRDF theories with pRDF theories. This operator has a least fixpoint that compactly represents all quadruples entailed by the theory.

Suppose (S, R) is a pRDF theory and n is a non-negative integer. Let $\mathcal{T}_n(s, p, v)$ be the set of p-paths between s and v of length n or less.

Definition 10 (Δ Operator) *Let (S, R) be a pRDF theory. Let $\epsilon(S, R) = \{(s, p', V, \delta) | \exists (s, p, V, \delta) \in R \wedge (p, rdfs : subPropertyOf, p') \in S\}$ and $\mu(S, R) = \{(s, p, \{v\}, \delta) | (|\mathcal{T}_2(s, p, v)| \geq 2) \wedge \delta(v) = \max_{P_j \in \mathcal{T}_2(s,p,v)} (\bigotimes_i \gamma_i^j)\}$. We define $\Delta(S, R) = (S, R \cup \epsilon(S, R) \cup \mu(S, R))$.*

Similarly to relational databases, with a fixed pRDF schema, we show that Δ is monotonic w.r.t. the pRDF instance.

Proposition 1 *Δ is monotonic in its second argument, i.e. if $R_1 \subseteq R_2$, then $\Delta(S, R_1) \subseteq \Delta(S, R_2)$. Hence, for a given (S, R), Δ has a least fixpoint (S, R') with $R \subseteq R'$. This least fixpoint is denoted lfp(S,R) and is also called the* closure *of (S, R).*

Example 6 *Consider the pRDF theory in Figure 1 and let $x \otimes y = xy$. The closure of this theory includes (AcuteBronchitis, hasComplication, {CorPulmonale},{0.002}); the probability was computed as the maximum on the two paths between $AcuteBronchitis$ and $CorPulmonale$.*

Clearly, (S, R) entails $\Delta(S, R)$ according to Definitions 8 and 9, thus $(S, R) \models_{\otimes}$ lfp(S,R). Soundness of the inference system defined by the closure is thus guaranteed. The following proposition establishes the completeness of the closure, showing that all quadruples entailed by (S, R) are either in lfp(S,R) or trivially entailed by a single quadruple in it.

Theorem 2 *Let (S, R) be a pRDF theory and let (S, R_{fp}) = lfp(S,R). Then for any (s, p, V, δ) s.t. $(S, R) \models_{\otimes} (s, p, V, \delta)$, $\exists\ (s, p, V', \delta') \in R_{fp}$ such that $V \subseteq V' \wedge \forall\ v \in V, \delta(v) \leq \delta'(v)$.*

Justification. Let us assume that $(S, R) \models_{\otimes} (s, p, V, \delta)$ and the proposition does not hold. Then there are two possible cases.

(1) For each $(s, p, V', \delta') \in R_{fp}$, $\exists\ v \in V - V'$. It is then straightforward that $\not\exists (s, p, V'', \delta'') \in R$ s.t. $v \in V''$. Hence, from the justification of Proposition 1, the worlds containing quadruples with s, p, v are not in $\mathcal{W}_R$, thus there exists a satisfying interpretation such that $\sum_{\{W \in \mathcal{W} | (s,p,v) \in W\}} I(W) = 0$. Since $\delta(v) > 0$, I cannot satisfy (s, p, V, δ), leading to contradiction.

(2) For each $(s, p, V', \delta') \in R_{fp}$ such that $V \subseteq V'$, $\exists v \in V$ s.t. $\delta(v) > \delta'(v')$. By induction on the structure $\mu(S, R)$, the probability value assigned to each p-path is the same as the value used in the system of equations in the justification of Proposition 1. Since there are an infinite number of satisfying interpretations that meet (stricter) equality conditions than the inequalities in Definition 8, there exists a satisfying interpretation that does not satisfy (s, p, V, δ).

5 Atomic pRDF queries

An atomic pRDF query has the form (s, p, v, λ) where at most one of the members of the quadruple is allowed to be a variable. We prefix variables with a question mark. In this section, we show how to answer queries in the cases when the subject s, the property p and the probability λ are variables (the case when the value v is a variable is analogous to the case for s and hence is not repeated).

Theorem 2 provides a simple algorithm for answering atomic queries. The algorithm would: (1) compute lfp(S,R); (2) Compute A, the set of potential answers by performing a linear search for possible substitutions of q to quadruples in lfp(S,R); (3) Remove from A any redundant quadruples (w.r.t. entailment).

The simple method shown above is inefficient since it computes the entire lfp(S,R). We will now present efficient algorithms for answering atomic queries that prune the search space significantly.

Definition 11 (Answer) *Let (S, R) be a pRDF theory and let $q = (s, p, v, \lambda)$ be a simple query. The answer to q is $Ans_q(S, R) = \{(s', p', v', \lambda') | ((S, R) \models_{\otimes} (s', p', v', \lambda') \wedge (\exists\ \theta$ substitution s.t. $(s\theta = s') \wedge (p\theta = p') \wedge (v\theta = v') \wedge (\lambda\theta \leq \lambda')) \wedge (\not\exists (s', p', v', \lambda'') \in Ans_q(S, R)$ s.t. $\lambda'' > \lambda'))\}$.*

Intuitively, the answer to (s, p, v, λ) consists of all instances of this query that are entailed by (S, R) subject to the restriction that if (S, R) entails (s', p', v', λ') and $(s'', p'', v'', \lambda'')$ and $\lambda'' < \lambda'$ then $(s'', p'', v'', \lambda'')$ is not

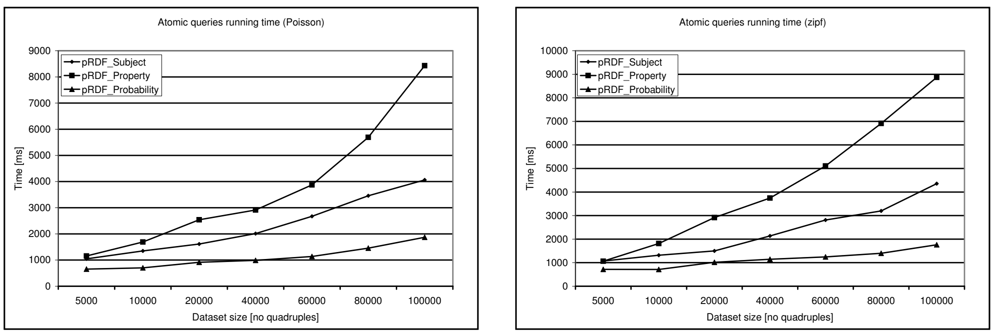
Algorithm pRDF_Subject$(S, R, \otimes, q = (?s, p_q, v_q, \lambda_q))$
Input: pRDF theory (S, R), t-norm $\otimes$, query q.
Output: $Ans_q(S, R)$.
Notation: $SP(p) = \{p' \in \mathcal{P} | (p', rdfs: subPropertyOf^*, p)\}$.

1. $R' \leftarrow \{(s, p', V, \delta) \in R | p' \in SP(p_q)\}$;
2. **if** $p_q \in \mathcal{P}^t$ **then**
3. $Q \leftarrow \{(s, \lambda_s) | \exists (s, p, V, \delta) \in R' \ s.t. \ (v_q \in V) \wedge (\lambda_s = \delta(v) \geq \lambda_q)\}$;
4. $Q' \leftarrow Q$;
5. **while** $Q' \neq \emptyset$ **do**
6. $(s, \lambda_s) \leftarrow$ remove from Q' element with highest λ_s;
7. $Q'' \leftarrow \{(s', \lambda_{s'} \otimes \lambda_s) | \exists (s, p, V, \delta) \in R' \ s.t. \ (s \in V) \wedge (\lambda_{s'} = \delta(s) \geq min_{\otimes}(\lambda_s, \lambda_q))\}$;
8. $Q' \leftarrow Q' \cup Q''$;
9. $Q \leftarrow Q \cup Q''$;
10. $Q, Q' \leftarrow$ replace $(s, \lambda_s), (s, \lambda'_s) \in Q, Q'$ with $(s, max(\lambda_s, \lambda'_s))$;
11. **end**
12. $Ans \leftarrow \{(s, p_q, v_q, \lambda) | \exists (s, \lambda) \in Q\}$;
13. **else**
    $Ans \leftarrow \{(s, p_q, v_q, \lambda) | (\exists (s, p, V, \delta) \in R \ s.t. \ (v_q \in V) \wedge (\delta(v_q) = \lambda \geq \lambda_q)) \wedge (\nexists (s, p_q, v_q, \lambda') \in Ans \ s.t. \lambda' > \lambda)\}$;
14. **end**
15. **return** Ans;

Algorithm pRDF_Property$(S, R, \otimes, q = (s_q, ?p, v_q, \lambda_q))$
Input: pRDF theory (S, R), t-norm $\otimes$, query q.
Output: $Ans_q(S, R)$.

1. $Q \leftarrow \{(v, p, \lambda_v) | \exists (s_q, p, V, \delta) \in R \ s.t. \ (v \in V) \wedge (\lambda_v = \delta(v) \geq \lambda_q) \wedge (p \in \mathcal{P}^t)\}$;
2. $Ans \leftarrow \emptyset$;
3. **while** $Q \neq \emptyset$ **do**
4. $(v, p, \lambda_v) \leftarrow$ remove from Q element with highest λ_v;
5. $Q' \leftarrow \{(v', p', \lambda_{v'} \otimes \lambda_v) | \exists (v, p', V, \delta) \in R \ s.t. \ (v' \in V) \wedge ((p, rdfs: subPropertyOf, p') \in S) \wedge (\lambda_{v'} = \delta(v') \geq min_{\otimes}(\lambda_v, \lambda_q))\}$;
6. $Q' \leftarrow Q' \cup \{(v', p, \lambda_{v'} \otimes \lambda_v) | \exists (v, p', V, \delta) \in R \ s.t. \ (v' \in V) \wedge ((p', rdfs: subPropertyOf, p) \in S) \wedge (\lambda_{v'} = \delta(v') \geq min_{\otimes}(\lambda_v, \lambda_q))\}$;
7. $Ans \leftarrow Ans \cup \{(s_q, p, v_q, \lambda) | \exists (v_q, p, \lambda) \in Q'\}$;
8. $Q' \leftarrow$ remove elements containing v_q from Q';
9. $Q \leftarrow Q \cup Q'$;
10. $Q \leftarrow$ replace $(v, p, \lambda_v), (v, p, \lambda'_v) \in Q$ with $(v, p, max(\lambda_v, \lambda'_v))$;
11. **end**
12. $Ans \leftarrow Ans \cup \{(s_q, p, v_q, \lambda) | \exists \ (s, p, V, \delta) \in R \ s.t. \ (p \in \mathcal{P}^{nt}) \wedge (v_q \in V) \wedge (\delta(v) = \lambda \geq \lambda_q)\}$;
13. $Ans \leftarrow Ans - \{$redundant quadruples in $Ans\}$;
14. **return** Ans;

Figure 2. Algorithms pRDF_Subject and pRDF_Property



(a) Poisson distribution

(b) Zipf distribution

Figure 3. Atomic query running time

included as it is clearly a redundant quadruple that adds nothing.

Example 7 *Consider the pRDF theory in Figure 1. The following are simple queries and their answers w.r.t. the $min(x, y)$ t-norm:*

What condition is associated with Pneumonia with probability above .6? q=(?s,associatedWith,Pneumonia,.6). The answer is {(Flu, associatedWith, Pneumonia, .65),(AcuteBronchitis, associatedWith, Pneumonia, .65)}.

What is the relation between Flu and Pneumonia with probability above .5? q=(Flu,?p,Pneumonia,.5). The answer is {(Flu, associatedWith, Pneumonia, .65)}.

Algorithm *pRDF_Subject* shown in Figure 2 computes the answer to atomic queries with an unknown subject. The algorithm prunes the search space at every inference step by excluding quadruples that are below the query probability λ_q on line 7; due to the aforementioned properties of $min_{\otimes}$, the pruned quadruples cannot be in $Ans_q(S, R)$. The probability assigned to a triple (s, p, v) (in cases where there are several p-paths between s and v) in Definition 10 is computed in line 10. *pRDF_Subject* also performs an initial pruning step on line 1, by limiting the search to the portion of the pRDF theory that is related to p_q. Algorithm *pRDF_Property* takes similar advantage of the properties of $min_{\otimes}$ to minimize the search space. For reasons of space, we omit two (similar) algorithms that compute the answers to simple queries with unknown object (pRDF_Object) and unknown probability (pRDF_Probability), which were implemented as part of the experimental evaluation.

Theorem 3 *Algorithms pRDF_Subject and pRDF_Property return $Ans_q(S, R)$ for each query q to a pRDF theory (S, R).*

6 Experimental results

We have developed an experimental prototype of the pRDF system consisting of about 1700 lines of Java code. Experiments were conducted on a Pentium 4

3.2 GHz machine with 512 MB or RAM running SuSE Linux 9.1. The experiments were performed on synthetically generated datasets.

In the first set of experiments, $|\mathcal{P}| = 50$ and $|\mathcal{P}^t| = 15$. The average width of the pRDF theory (the width of a theory is $\max_{(s,p,V,\delta)}(|V|)$) was 15. We varied $|R|$ between 5,000 and 100,000 quadruples and measured the running time, including any disk I/O overhead. The experiment was performed by using the Poisson and Zipf distribution respectively for quadruple generation. The results averaged over 30 runs are shown in Figure 3(a) and (b). As expected, the *pRDF_Probability* is the fastest algorithm; this is due to (i) knowing s_q, p_q, v_q and (ii) the reduction of the space needed to store intermediate values of *Ans*. *pRDF_Subject* has a linear trend, whereas *pRDF_Property* is the least efficient due to the large search space. In a separate experiment we have determined that by using he naive method (computing lfp(S,R) and performing a linear search), the running time increases by a factor ranging between 6 and 15. Clearly, the probability distribution used for the quadruples has very little influence on the running time of the algorithms.

In a second set of experiments, with $|R|$=100,000, $|\mathcal{P}| = 50$ and $|\mathcal{P}^t| = 15$, we varied the average width of the pRDF theory from 5 to 35, with a standard deviation of 1.5; the results were averaged over 30 independent executions. We found that the number of iterations for all three algorithms presented in Figure 3 was linearly increasing up to a width of 20, after which the probabilities in the quadruples decrease enough so that more and more paths are pruned, leading to a linear decrease in the running time.

7 Related work and conclusion

Our work builds on a larger body of research about uncertainty in logic and Web languages. In terms of representation, Fukushige [4] provides a comprehensive method for representing probabilistic relations in RDF; we focus mostly on pRDF semantics and query answering methods. In terms of probabilistic extensions to logic and ontology languages, Ding et al [2] propose a Bayesian network-based extension to the ontology language OWL; Koller et. al [7] and Giugno et al. [5] propose probabilistic extensions of description logics; Costa et al. [1] extend OWL with uncertainty based on first-order Bayesian logic. Our work shows that RDF is also good basis for a comprehensive probabilistic extension with query processing capabilities. The simplicity of the pRDF language (as compared to OWL) allows for very efficient query algorithms. Pan et al. [9] and Mazzieri et. al [8] propose fuzzy extension to causal networks and the OWL language respectively. Our work makes use of some fuzzy concepts, such as the use of t-norms; the methods shown for pRDF querying avoid the space and running time requirements described in [9].

In this paper, we have developed a *Probabilistic RDF* framework within which users can express probabilistic information about the relationships expressed in RDF. We have developed results on the consistency of pRDF theories, together with a fixpoint semantics and algorithms to answer queries to pRDF theories. Our prototype implementation of pRDF is, to our knowledge, the first of its kind.

References

[1] P. C. G. da Costa, K. B. Laskey, and K. J. Laskey. Pr-owl: A bayesian ontology language for the semantic web. In *ISWC-URSW*, pages 23–33, 2005.

[2] Z. Ding and Y. Peng. A probabilistic extension to ontology language owl. In *Proceedings of HICCS '04 – Track 4*, page 10. IEEE Computer Society, 2004.

[3] R. Fagin. Combining fuzzy information from multiple systems. *J. Computer and Systems Sciences*, 58:83–99, 1999.

[4] Y. Fukushige. Representing probabilistic relations in rdf. In *ISWC-URSW*, pages 106–107, 2005.

[5] R. Giugno and T. Lukasiewicz. P-SHOQ(D): A Probabilistic Extension of SHOQ(D) for Probabilistic Ontologies in the Semantic Web. In *Proceedings of JELIA '02*, pages 86–97. Springer-Verlag, 2002.

[6] J. Halpern. The relationship between knowledge, belief, and certainty. In *Proceedings of UAI-90*, pages 142–151. Elsevier Science Publishing Company, Inc., 1990.

[7] D. Koller, A. Y. Levy, and A. Pfeffer. P-classic: A tractable probablistic description logic. In *AAAI/IAAI*, pages 390–397, 1997.

[8] M. Mazzieri and A. F. Dragoni. A fuzzy semantics for semantic web languages. In *ISWC-URSW*, pages 12–22, 2005.

[9] H. Pan and D. McMichael. Fuzzy causal probabilistic networks - a new ideal and practical inference engine. In *Proc. 1st International Conference on Multisource-Multisensor Information Fusion*, 1998.