Fuzzyness in Semantic Web Languages

The case of RDFS

Fuzzy RDF

- Statement (triples) may have attached a degree in $[0, 1]$: for $n \in [0, 1]$

$$\langle (subject, predicate, object), n \rangle$$

- Meaning: the degree of truth of the statement is at least n
- For instance,

$$\langle (o1, IsAbout, snoopy), 0.8 \rangle$$

Fuzzy RDF Syntax

- Fuzzy RDF triple (or Fuzzy RDF atom):

$$\langle \tau, n \rangle \in (\mathbf{UBL} \times \mathbf{U} \times \mathbf{UBL}) \times [0, 1]$$

  - $s \in \mathbf{UBL}$ is the subject
  - $p \in \mathbf{U}$ is the predicate
  - $o \in \mathbf{UBL}$ is the object
  - $n \in (0, 1]$ is the degree of truth

- Example:

$$\langle (\text{audiTT}, \text{type}, \text{SportCar}), 0.8 \rangle$$

Fuzzy RDF Semantics

- Fuzzy RDF interpretation $\mathcal{I}$ over a vocabulary V is a tuple

$$\mathcal{I} = \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, P[\![\cdot]\!], C[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle \ ,$$

  where

  - $\Delta_R, \Delta_P, \Delta_C, \Delta_L$ are the interpretations domains of $\mathcal{I}$
  - $P[\![\cdot]\!], C[\![\cdot]\!], \cdot^{\mathcal{I}}$ are the interpretation functions of $\mathcal{I}$

$$\mathcal{I} = \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, P[\![\cdot]\!], C[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle$$

1. Δ_R is a nonempty set of resources, called the domain or universe of $\mathcal{I}$;
2. Δ_P is a set of property names (not necessarily disjoint from Δ_R);
3. $\Delta_C \subseteq \Delta_R$ is a distinguished subset of Δ_R identifying if a resource denotes a class of resources;
4. $\Delta_L \subseteq \Delta_R$, the set of literal values, Δ_L contains all plain literals in $\mathbf{L} \cap V$;
5. $P[\![\cdot]\!]$ maps each property name $p \in \Delta_P$ into a partial function $P[\![p]\!] : \Delta_R \times \Delta_R \to [0, 1]$, i.e. assigns a degree to each pair of resources, denoting the degree of being the pair an instance of the property p;
6. $C[\![\cdot]\!]$ maps each class $c \in \Delta_C$ into a partial function $C[\![c]\!] : \Delta_R \to [0, 1]$, i.e. assigns a degree to every resource, denoting the degree of being the resource an instance of the class c;
7. $\cdot^{\mathcal{I}}$ maps each $t \in \mathbf{UL} \cap V$ into a value $t^{\mathcal{I}} \in \Delta_R \cup \Delta_P$, i.e. assigns a resource or a property name to each element of **UL** in V, and such that $\cdot^{\mathcal{I}}$ is the identity for plain literals and assigns an element in Δ_R to elements in **L**;
8. $\cdot^{\mathcal{I}}$ maps each variable $x \in \mathbf{B}$ into a value $x^{\mathcal{I}} \in \Delta_R$, i.e. assigns a resource to each variable in **B**.

Models

Let G be a graph over ρdf.

- An interpretation $\mathcal{I}$ is a model of G under ρdf, denoted $\mathcal{I} \models G$, iff
  - $\mathcal{I}$ is an interpretation over the vocabulary $\rho\text{df} \cup \mathit{universe}(G)$
  - $\mathcal{I}$ satisfies the following conditions:

Simple:

1. for each $\langle (s, p, o), n \rangle \in G$, $p^{\mathcal{I}} \in \Delta_P$ and $P[\![p^{\mathcal{I}}]\!](s^{\mathcal{I}}, o^{\mathcal{I}}) \geq n$;

Subproperty:

1. $P[\![\mathrm{sp}^{\mathcal{I}}]\!]$ is transitive over Δ_P;
2. if $P[\![\mathrm{sp}^{\mathcal{I}}]\!](p, q)$ is defined then $p, q \in \Delta_P$ and

$$P[\![\mathrm{sp}^{\mathcal{I}}]\!](p, q) = \inf_{(x,y) \in \Delta_R \times \Delta_R} P[\![p]\!](x, y) \implies P[\![q]\!](x, y) \; ;$$

Models (cont.)

Subclass:

1. $P[\![\text{sc}^{\mathcal{I}}]\!]$ is transitive over Δ_C;
2. if $P[\![\text{sc}^{\mathcal{I}}]\!](c, d)$ is defined then $c, d \in \Delta_C$ and

$$P[\![\text{sc}^{\mathcal{I}}]\!](c, d) = \inf_{x \in \Delta_R} C[\![c]\!](x) \implies C[\![d]\!](x) ;$$

Typing I:

1. $C[\![c]\!](x) = P[\![\text{type}^{\mathcal{I}}]\!](x, c)$;
2. if $P[\![\text{dom}^{\mathcal{I}}]\!](p, c)$ is defined then

$$P[\![\text{dom}^{\mathcal{I}}]\!](p, c) = \inf_{(x,y) \in \Delta_R \times \Delta_R} P[\![p]\!](x, y) \implies C[\![c]\!](x) ;$$

3. if $P[\![\text{range}^{\mathcal{I}}]\!](p, c)$ is defined then

$$P[\![\text{range}^{\mathcal{I}}]\!](p, c) = \inf_{(x,y) \in \Delta_R \times \Delta_R} P[\![p]\!](x, y) \implies C[\![c]\!](y) ;$$

Typing II:

1. For each e $\in \rho$df, $\text{e}^{\mathcal{I}} \in \Delta_P$
2. if $P[\![\text{dom}^{\mathcal{I}}]\!](p, c)$ is defined then $p \in \Delta_P$ and $c \in \Delta_C$
3. if $P[\![\text{range}^{\mathcal{I}}]\!](p, c)$ is defined then $p \in \Delta_P$ and $c \in \Delta_C$
4. if $P[\![\text{type}^{\mathcal{I}}]\!](x, c)$ is defined then $c \in \Delta_C$

Models (cont.)

Note:

- In the crisp case, if c is a sub-class of d then we impose that $C[\![c]\!] \subseteq C[\![d]\!]$
- This may be seen as the formula

$$\forall x.c(x) \implies d(x) ,$$

- The fuzzyfication is

$$P[\![\mathrm{sc}^{\mathcal{I}}]\!](c,d) = \inf_{x \in \Delta_R} C[\![c]\!](x) \implies C[\![d]\!](x) ;$$

- Similarly, e.g., "property p has domain c" may be seen as the formula

$$\forall x \forall y.p(x,y) \implies c(x) ,$$

- The fuzzyfication is

$$P[\![\mathrm{dom}^{\mathcal{I}}]\!](p,c) = \inf_{(x,y) \in \Delta_R \times \Delta_R} P[\![p]\!](x,y) \implies C[\![c]\!](x) .$$

- G entails H under ρdf, denoted $G \models H$, iff
  - every model under ρdf of G is also a model under ρdf of H

Example & Model

$G = \{\langle(\mathit{audiTT}, \text{type}, \mathit{SportsCar}), 0.8\rangle, \langle(\mathit{SportsCar}, \text{sc}, \mathit{PassengerCar}), 0.9\rangle\}$ t-norm: Product

$$\mathcal{I} = \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, P[\![\cdot]\!], C[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle$$

$$\begin{aligned}
\Delta_R &= \{\mathit{audiTT}, \mathit{SportsCar}, \mathit{PassengerCar}\} \\
\Delta_P &= \{\text{type}, \text{sc}\} \\
\Delta_C &= \{\mathit{SportsCar}, \mathit{PassengerCar}\} \\
P[\![\text{type}]\!] &= \{\langle\langle \mathit{audiTT}, \mathit{SportsCar}\rangle, 0.8\rangle, \langle\langle \mathit{audiTT}, \mathit{PassengerCar}\rangle, 0.72\rangle\} \\
P[\![\text{sc}]\!] &= \{\langle\langle \mathit{SportsCar}, \mathit{PassengerCar}\rangle, 0.9\rangle\} \\
C[\![\mathit{SportsCar}]\!] &= \{\langle \mathit{audiTT}, 0.8\rangle\} \\
C[\![\mathit{PassengerCar}]\!] &= \{\langle \mathit{audiTT}, 0.72\rangle\} \\
t^{\mathcal{I}} &= t \text{ for all } t \in \mathbf{UL}
\end{aligned}$$

$\mathcal{I} \models G$ $\mathcal{I}$ is a model of G

Example (Entailment)

$G = \{\langle(\mathit{audiTT}, \text{type}, \mathit{SportsCar}), 0.8\rangle, \langle(\mathit{SportsCar}, \text{sc}, \mathit{PassengerCar}), 0.9\rangle\}$ t-norm: Product

$$\mathcal{I} = \langle \Delta_R, \Delta_P, \Delta_C, \Delta_L, P[\![\cdot]\!], C[\![\cdot]\!], \cdot^{\mathcal{I}} \rangle$$

$$\begin{aligned}
\Delta_R &= \{\mathit{audiTT}, \mathit{SportsCar}, \mathit{PassengerCar}\} \\
\Delta_P &= \{\text{type}, \text{sc}\} \\
\Delta_C &= \{\mathit{SportsCar}, \mathit{PassengerCar}\} \\
P[\![\text{type}]\!] &= \{\langle\langle \mathit{audiTT}, \mathit{SportsCar}\rangle, 0.8\rangle, \langle\langle \mathit{audiTT}, \mathit{PassengerCar}\rangle, 0.72\rangle\} \\
P[\![\text{sc}]\!] &= \{\langle\langle \mathit{SportsCar}, \mathit{PassengerCar}\rangle, 0.9\rangle\} \\
C[\![\mathit{SportsCar}]\!] &= \{\langle \mathit{audiTT}, 0.8\rangle\} \\
C[\![\mathit{PassengerCar}]\!] &= \{\langle \mathit{audiTT}, 0.72\rangle\} \\
t^{\mathcal{I}} &= t \text{ for all } t \in \mathbf{UL}
\end{aligned}$$

$G \models \langle(\mathit{audiTT}, \text{type}, \mathit{PassengerCar}), 0.72\rangle$ In all models $\mathcal{I}$ of G, $P[\![\text{type}]\!](\mathit{audiTT}, \mathit{PassengerCar}) = 0.72$

Deduction System for fuzzy RDF

1. Simple:

$$(a) \quad \frac{G}{G'} \text{ for a map } \mu : G' \to G \qquad (b) \quad \frac{G}{G'} \text{ for } G' \subseteq G$$

2. Subproperty:

$$(a) \quad \frac{\langle (A, \text{sp}, B), n\rangle, \langle (B, \text{sp}, C), m\rangle}{\langle (A, \text{sp}, C), n \otimes m\rangle} \qquad (b) \quad \frac{\langle (A, \text{sp}, B), n\rangle, \langle (X, A, Y), m\rangle}{\langle (X, B, Y), n \otimes m\rangle}$$

3. Subclass:

$$(a) \quad \frac{\langle (A, \text{sc}, B), n\rangle, \langle (B, \text{sc}, C), m\rangle}{\langle (A, \text{sc}, C), n \otimes m\rangle} \qquad (b) \quad \frac{\langle (A, \text{sc}, B), n\rangle, \langle (X, \text{type}, A), m\rangle}{\langle (X, \text{type}, B), n \otimes m\rangle}$$

4. Typing:

$$(a) \quad \frac{\langle (A, \text{dom}, B), n\rangle, \langle (X, A, Y), m\rangle}{\langle (X, \text{type}, B), n \otimes m\rangle} \qquad (b) \quad \frac{\langle (A, \text{range}, B), n\rangle, \langle (X, A, Y), m\rangle}{\langle (Y, \text{type}, B), n \otimes m\rangle}$$

5. Implicit Typing:

$$(a) \quad \frac{\langle (A, \text{dom}, B), n\rangle, \langle (C, \text{sp}, A), m\rangle, \langle (X, C, Y), r\rangle}{\langle (X, \text{type}, B), n \otimes m \otimes r\rangle}$$

$$(b) \quad \frac{\langle (A, \text{range}, B), n\rangle, \langle (C, \text{sp}, A), m\rangle, \langle (X, C, Y), r\rangle}{\langle (Y, \text{type}, B), n \otimes m \otimes r\rangle}$$

Deduction System for Fuzzy RDF (cont.)

- Notion of proof (as for crisp RDF)):
  - Let G and H be graphs
  - Then $G \vdash H$ iff there is a sequence of graphs $P_1, \ldots, P_k$ with $P_1 = G$ and $P_k = H$, and for each j $(2 \leq j \leq k)$ one of the following holds:
    1. there exists a map $\mu : P_j \to P_{j-1}$ (rule (1a));
    2. $P_j \subseteq P_{j-1}$ (rule (1b));
    3. there is an instantiation $\frac{R}{R'}$ of one of the rules (2)–(5), such that $R \subseteq P_{j-1}$ and $P_j = P_{j-1} \cup R'$.
- The sequence of rules used at each step (plus its instantiation or map), is called a proof of H from G.

Proposition (Soundness and completeness)

The fuzzy RDF proof system $\vdash$ is sound and complete for $\models$, that is, $G \vdash H$ iff $G \models H$.

Example (Proof)

$G = \{\langle(\mathit{audiTT}, \text{type}, \mathit{SportsCar}), 0.8\rangle, \langle(\mathit{SportsCar}, \text{sc}, \mathit{PassengerCar}), 0.9\rangle\}$ t-norm: Product

Let us proof that

$$G \models \langle(\mathit{audiTT}, \text{type}, \mathit{PassengerCar}), 0.72\rangle$$

G	$\vdash$	$\langle(\mathit{audiTT}, \text{type}, \mathit{SportsCar}), 0.8\rangle,$	(1)	Rule Simple (b)
G	$\vdash$	$\langle(\mathit{SportsCar}, \text{sc}, \mathit{PassengerCar}), 0.9\rangle$	(2)	Rule Simple (b)
G	$\vdash$	$\langle(\mathit{audiTT}, \text{type}, \mathit{PassengerCar}), 0.72\rangle$	(3)	Rule SubClass (b) applied to (1) + (2) using product t-norm

Fuzzy RDF Query Answering

- We assume that a fuzzy RDF graph G is *ground* and *closed*, i.e., G is closed under the application of the rules (2)-(5)
- Conjunctive query: extends a crisp RDF query and is of the form

$$\langle q(\mathbf{x}), s\rangle \leftarrow \exists \mathbf{y}.\langle \tau_1, s_1\rangle, \ldots, \langle \tau_n, s_n\rangle, s = f(s_1, \ldots, s_n, p_1(\mathbf{z}_1), \ldots, p_h(\mathbf{z}_h))$$

where additionally

  - $\mathbf{z_i}$ are tuples of terms in **UL** or variables in $\mathbf{x}$ or $\mathbf{y}$;
  - p_j is an n_j-ary *fuzzy predicate* assigning to each n_j-ary tuple $\mathbf{t}_j$ in **UL** a *score* $p_j(\mathbf{t}_j) \in [0, 1]$. Such predicates are called *expensive predicates* as the score is not pre-computed off-line, but is computed on query execution. We require that an n-ary fuzzy predicate p is *safe*, that is, there is not an m-ary fuzzy predicate p' such that $m < n$ and $p = p'$. Informally, all parameters are needed in the definition of p;
  - f is a *scoring* function $f\colon ([0, 1])^{n+h} \rightarrow [0, 1]$, which combines the scores s_i of the n triples and the h fuzzy predicates into an overall *score* to be assigned to the rule head. We assume that f is *monotone*, that is, for each $\mathbf{v}, \mathbf{v}' \in ([0, 1])^{n+h}$ such that $\mathbf{v} \leq \mathbf{v}'$, it holds $f(\mathbf{v}) \leq f(\mathbf{v}')$, where $(v_1, \ldots, v_{n+h}) \leq (v'_1, \ldots, v'_{n+h})$ iff $v_i \leq v'_i$ for all i;
  - the scoring variables s and s_i are distinct from those in $\mathbf{x}$ and $\mathbf{y}$ and s is distinct from each s_i
- If clear from the context, we may omit the exitential quantification $\exists \mathbf{y}$
- We may omit s_i and in that case $s_i = 1$ is assumed
- $s = f(s_1, \ldots, s_n, p_1(\mathbf{z}_1), \ldots, p_h(\mathbf{z}_h))$ is called the *scoring atom*. We may also omit the scoring atom and in that case $s = 1$ is assumed.
- For instance, the query

$$\langle q(x), s\rangle \leftarrow \langle (x, \text{type}, \text{SportCar}), s_1\rangle, (x, \text{hasPrice}, y), s = s_1 \cdot \text{cheap}(y)$$

where e.g. $\text{cheap}(p) = \text{ls}(0, 10000, 12000)$, has intended meaning to retrieve all cheap sports car. Any answer is scored according to the product of being cheap and a sports car

Fuzzy RDF Query Answering (cont.)

- We will also write a query as

$$\langle q(\mathbf{x}), s\rangle \leftarrow \exists \mathbf{y}.\langle \varphi(\mathbf{x},\mathbf{y}), \mathbf{s}\rangle \ ,$$

  where
  - $\varphi(\mathbf{x},\mathbf{y})$ is $\langle \tau_1, s_1\rangle, \ldots, \langle \tau_n, s_n\rangle, s = f(\mathbf{s}, p_1(\mathbf{z}_1), \ldots, p_h(\mathbf{z}_h))$
  - $\mathbf{s} = \langle s_1, \ldots, s_n\rangle$
- Furthermore, $q(\mathbf{x})$ is called the head of the query, while $\exists \mathbf{y}.\varphi(\mathbf{x},\mathbf{y})$ is is called the body of the query
- Finally, a disjunctive query (or, *union of conjunctive queries*) $\mathbf{q}$ is, as usual, a finite set of conjunctive queries in which all the rules have the same head
- For instance, the disjunctive query

$$\begin{aligned}
\langle q(x), s\rangle &\leftarrow \langle (x, \text{type}, \text{SportCar}), s_1\rangle, (x, \text{hasPrice}, y), s = s_1 \cdot \text{cheap}(y)\\
\langle q(x), s\rangle &\leftarrow \langle (x, \text{type}, \text{PassengerCar}), s_1\rangle, s = s_1
\end{aligned}$$

  has intended meaning to retrieve all sports cars or passenger cars

Fuzzy RDF Query Answering (cont.)

- Consider a fuzzy graph G, a query $\langle q(\mathbf{x}), s\rangle \leftarrow \exists \mathbf{y}.\langle \varphi(\mathbf{x}, \mathbf{y}), \mathbf{s}\rangle$, and a vector $\mathbf{t}$ of terms in **UL** and $s \in [0, 1]$
- We say that $\langle q(\mathbf{t}), s\rangle$ is entailed by G, denoted $G \models \langle q(\mathbf{t}), s\rangle$, iff
  - in any model $\mathcal{I}$ of G, there is a vector $\mathbf{t}'$ of terms in **UL**, a vector $\mathbf{s}$ of scores in $[0, 1]$ such that $\mathcal{I}$ is a model of $\langle \varphi(\mathbf{t}, \mathbf{t}'), \mathbf{s}\rangle$ (the scoring atom is satisfied iff s is the value of the evaluation of the score combination function)
- For a disjunctive query $\mathbf{q} = \{q_1, \ldots, q_m\}$, we say that $\langle \mathbf{q}(\mathbf{t}), s\rangle$ is entailed by G, denoted $G \models \langle \mathbf{q}(\mathbf{t}), s\rangle$, iff $G \models \langle q_i(\mathbf{t}), s\rangle$ for some $q_i \in \mathbf{q}$
- We say that s is *tight* iff $s = \sup\{s' \mid G \models \langle \mathbf{q}(\mathbf{t}), s'\rangle\}$
- If $G \models \langle \mathbf{q}(\mathbf{t}), s\rangle$ and s is tight then $\langle \mathbf{t}, s\rangle$ is called an *answer* to $\mathbf{q}$
- The answer set of $\mathbf{q}$ w.r.t. G is defined as

$$ans(G, \mathbf{q}) = \{\langle \mathbf{t}, s\rangle \mid G \models \langle \mathbf{q}(\mathbf{t}), s\rangle,\ s \text{ is tight}\}$$

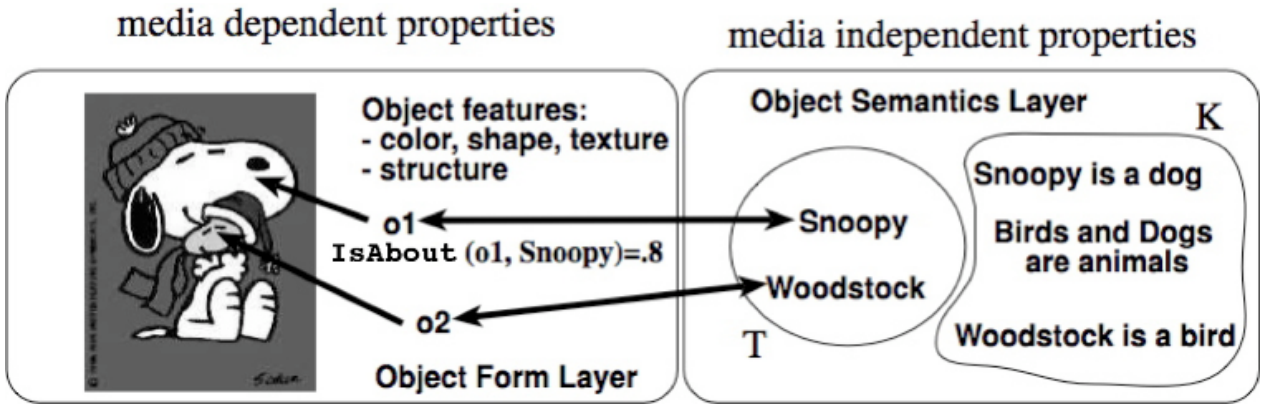
Top-k Retrieval: Given a fuzzy graph G, and a disjunctive query $\mathbf{q}$, retrieve k answers $\langle \mathbf{t}, s\rangle$ with maximal scores and rank them in decreasing order relative to the score s, denoted

$$ans_k(G, \mathbf{q}) = \text{Top}_k\ ans(G, \mathbf{q})\ .$$

Fuzzy RDF Query Answering (cont.)

- A simple query answering procedure is the following:
  - Compute the closure of a graph off-line
  - Store the fuzzy RDF triples into a relational database supporting Top-k retrieval (e.g., RankSQL, Postgres)
  - Translate the fuzzy query into a top-k SQL statement
  - Execute the SQL statement over the relational database
- In practice, some care should be in place due to the large size of data: $\geq 10^9$ triples
- To date, no systems exists

Example



$$G = \left\{ \begin{array}{ll} \langle (o1, \mathit{IsAbout}, \mathit{snoopy}), 0.8 \rangle & \langle (o2, \mathit{IsAbout}, \mathit{woodstock}), 0.9 \rangle \\ (\mathit{snoopy}, \text{type}, \mathit{dog}) & (\mathit{woodstock}, \text{type}, \mathit{bird}) \\ \langle (\mathit{Bird}, \text{sc}, \mathit{SmallAnimal}), 0.7 \rangle & \langle (\mathit{Dog}, \text{sc}, \mathit{SmallAnimal}), 0.4 \rangle \\ (\mathit{dog}, \text{sc}, \mathit{Animal}) & (\mathit{bird}, \text{sc}, \mathit{Animal}) \\ (\mathit{SmallAnimal}, \text{sc}, \mathit{Animal}) & \end{array} \right\}$$

Consider the query

$$\langle q(x), s \rangle \quad \leftarrow \quad \langle (x, \mathit{IsAbout}, y), s_1 \rangle, \langle (y, \text{type}, \mathit{SmallAnimal}), s_2 \rangle, s = s_1 \cdot s_2$$

Then (under any t-norm)

$$\mathit{ans}(G, q) \quad = \quad \{\langle o1, 0.32 \rangle, \langle o2, 0.63 \rangle\}, \quad \mathit{ans}_1(G, q) = \{\langle o2, 0.63 \rangle\}$$