A Minimal Deductive System for General Fuzzy RDF

Umberto Straccia

ISTI-CNR, Pisa, Italy

straccia@isti.cnr.it

www.straccia.info

Introduction

- Interest in Semantic Web technologies grows
- RDF/S is both a logic and standard W3C Semantic Web Languages
- Crisp RDF isn't the best choice to represent vague information

"ISWC-09 is held near Washington D.C"

(source: ISWC 2009 Web page)

- the sentence should be true to some degree depending, *e.g.*, on the distance and context

ISWC-09 is held near Washington D.C

(source: ISWC 2009 Web page)

- Fuzzy RDF variants are emerging . . .
- In this work we provide,
  - A very general semantics for Fuzzy RDF
  - A deductive system for a salient fragment of fuzzy RDF
  - We show how to compute top-k answers of the union of conjunctive queries in which answers may be scored by means of a scoring function
  - Crisp RDF is a special case (backward compatibility is guaranteed)
  - Implementation is simple
  - Computational complexity and scalability is as for crisp RDF

Outline

- Crash course on Fuzzy Sets & Mathematical Fuzzy Logic
- Fuzzy RDF
- Query answering
- Hints for implmentors
- Summary & Outlook

Preliminaries: Fuzzy Sets [Zad65]

- A fuzzy set R is a function $R\colon X \to [0,1]$
- A fuzzy set A is included in B (denoted $A \subseteq B$) iff $\forall x \in X, A(x) \leqslant B(x)$
- The degree of subsumption between A and B is $\inf_{x \in X} A(x) \Rightarrow B(x)$
- A (binary) fuzzy relation R over sets X and Y is $R\colon X \times Y \to [0,1]$
- The composition of $R_1 : X \times Y \to [0,1]$ and $R_2\colon Y \times Z \to [0,1]$ is $(R_1 \circ R_2)(x,z) = \sup_{y \in Y} R_1(x,y) \otimes R_2(y,z)$
- A fuzzy relation R is reflexive iff $\forall x \in X, R(x,x) = 1$
- R is symmetric iff $\forall x \in X, y \in Y, R(x,y) = R(y,x)$
- R is transitive iff $R(x,z) \geqslant (R \circ R)(x,z)$
- $\otimes$, $\Rightarrow$ is t-norm and r-implication (next slide ...)

Preliminaries: Mathematical Fuzzy Logic [Háj98]

- Fuzzy statements: $\phi[n]$, where $n \in [0,1]$ and ϕ is a FOL statement
  - The degree of truth of ϕ is *at least n*
- Fuzzy interpretation: $\mathcal{I} : Atoms \to [0,1]$ and is then extended inductively:

$$
\begin{array}{ll}
\mathcal{I}(\phi \wedge \psi) = \mathcal{I}(\phi) \otimes \mathcal{I}(\psi) & \mathcal{I}(\phi \vee \psi) = \mathcal{I}(\phi) \oplus \mathcal{I}(\psi), \\
\mathcal{I}(\phi \to \psi) = \mathcal{I}(\phi) \Rightarrow \mathcal{I}(\psi) & \mathcal{I}(\neg \phi) = \ominus \mathcal{I}(\phi), \\
\mathcal{I}(\exists x.\phi(x)) = \sup_{c \in \Delta^{\mathcal{I}}} \mathcal{I}(\phi(c)) & \mathcal{I}(\forall x.\phi(x)) = \inf_{c \in \Delta^{\mathcal{I}}} \mathcal{I}(\phi(c))
\end{array}
$$

$\otimes$, $\oplus$, $\Rightarrow$, and $\ominus$ are *truth combination functions*

	Łukasiewicz Logic	Gödel Logic	Product Logic	"Zadeh Logic"
$a \otimes b$	$\max(a+b-1, 0)$	$\min(a, b)$	$a \cdot b$	$\min(a, b)$
$a \oplus b$	$\min(a+b, 1)$	$\max(a, b)$	$a + b - a \cdot b$	$\max(a, b)$
$a \Rightarrow b$	$\min(1 - a + b, 1)$	$\begin{cases} 1 & \text{if } a \leqslant b \\ b & \text{otherwise} \end{cases}$	$\min(1, b/a)$	$\max(1-a, b)$
$\ominus a$	$1 - a$	$\begin{cases} 1 & \text{if } a = 0 \\ 0 & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } a = 0 \\ 0 & \text{otherwise} \end{cases}$	$1 - a$

- Satisfiability: $\mathcal{I} \models \phi[n]$ iff $\mathcal{I}(\phi) \geqslant n$
- Best Entailment Degree (BED): $bed(KB, \phi) = \sup \{r \mid KB \models \phi[r]\}$

From RDF to Fuzzy RDF

RDF Syntax

- Pairwise disjoint alphabets
  - **U** (RDF URI references)
  - **B** (Blank nodes)
  - **L** (Literals)
- For simplicity we will denote unions of these sets simply concatenating their names
- We call elements in **UBL** terms (denoted *t*)
- We call elements in **B** variables (denoted *x*)

- RDF triple (or RDF atom):

$$(s, p, o) \in \mathbf{UBL} \times \mathbf{U} \times \mathbf{UBL}$$

  - s is the subject
  - p is the predicate
  - o is the object
- Example:

$$(\mathit{airplane}, \mathit{has}, \mathit{enginefault})$$

ρdf (restricted RDF) [MPG07]

- ρdf (read rho-df, the ρ from restricted rdf)
- ρdf is defined as the following subset of the RDFS vocabulary:

$$\rho\text{df} = \{\text{sp}, \text{sc}, \text{type}, \text{dom}, \text{range}\}$$

- (p, sp, q)
  - property p is a sub property of property q
- (c, sc, d)
  - class c is a sub class of class d
- (a, type, b)
  - a is of type b
- (p, dom, c)
  - domain of property p is c
- (p, range, c)
  - range of property p is c

- RDF graph (or simply a graph, or RDF Knowledge Base) is a set of RDF triples τ
- A subgraph is a subset of a graph
- The universe of a graph G, denoted by $universe(G)$ is the set of elements in **UBL** that occur in the triples of G
- The vocabulary of G, denoted by $voc(G)$ is the set $universe(G) \cap \mathbf{UL}$
- A graph is ground if it has no blank nodes (*i.e.* variables)

- A variable assignment: a function $\mu : \mathbf{UBL} \rightarrow \mathbf{UBL}$ preserving URIs and literals, *i.e.*,
  - $\mu(t) = t$, for all $t \in \mathbf{UL}$
- Given a graph G, we define

$$\mu(G) = \{(\mu(s), \mu(p), \mu(o)) \mid (s, p, o) \in G\}$$

- We speak of a variable assignment μ from G_1 to G_2, and write $\mu : G_1 \rightarrow G_2$, if μ is such that $\mu(G_1) \subseteq G_2$

Fuzzy RDF

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

$$(s, p, o)[n]$$

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

$$(ISWC09, near, WashingtonDC)[0.8]$$

Fuzzy RDF Syntax

- Fuzzy RDF triple (or Fuzzy RDF atom):

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

  - $s \in \mathbf{UBL}$ 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:

$$(\text{audiTT}, \text{type}, \text{SportCar})[0.8]$$

- Degree n may be omitted and in that case degree 1 is assumed

Fuzzy RDF Semantics

- Semantics generalizes that of crisp RDF
- 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 $\mathcal{I}$

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

Common parts between Crisp RDF and Fuzzy RDF

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. $\cdot^{\mathcal{I}}$ 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**
6. $\cdot^{\mathcal{I}}$ 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**
7. What are $P[\![\cdot]\!]$ and $C[\![\cdot]\!]$?

Crisp $P[\![\cdot]\!]$: $P[\![\cdot]\!]$ maps each property name $p \in \Delta_P$ into a subset $P[\![p]\!] \subseteq \Delta_R \times \Delta_R$, *i.e.* assigns an extension to each property name; *i.e.*

$$P[\![p]\!] : \Delta_R \times \Delta_R \to \{0, 1\}$$

Fuzzy $P[\![\cdot]\!]$: $P[\![\cdot]\!]$ 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;

Crisp $C[\![\cdot]\!]$: $C[\![\cdot]\!]$ maps each class $c \in \Delta_C$ into a subset $C[\![c]\!] \subseteq \Delta_R$, *i.e.* assigns a set of resources to every resource denoting a class; *i.e.*

$$C[\![c]\!] : \Delta_R \to \{0, 1\}$$

Fuzzy $C[\![\cdot]\!]$: $C[\![\cdot]\!]$ 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

Crisp $P[\![\cdot]\!]$: $P[\![\cdot]\!]$ maps each property name $p \in \Delta_P$ into a subset $P[\![p]\!] \subseteq \Delta_R \times \Delta_R$, *i.e.* assigns an extension to each property name; *i.e.*

$$P[\![p]\!] : \Delta_R \times \Delta_R \to \{0, 1\}$$

Fuzzy $P[\![\cdot]\!]$: $P[\![\cdot]\!]$ 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;

Crisp $C[\![\cdot]\!]$: $C[\![\cdot]\!]$ maps each class $c \in \Delta_C$ into a subset $C[\![c]\!] \subseteq \Delta_R$, *i.e.* assigns a set of resources to every resource denoting a class; *i.e.*

$$C[\![c]\!] : \Delta_R \to \{0, 1\}$$

Fuzzy $C[\![\cdot]\!]$: $C[\![\cdot]\!]$ 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

Crisp $P[\![\cdot]\!]$: $P[\![\cdot]\!]$ maps each property name $p \in \Delta_P$ into a subset $P[\![p]\!] \subseteq \Delta_R \times \Delta_R$, *i.e.* assigns an extension to each property name; *i.e.*

$$P[\![p]\!] : \Delta_R \times \Delta_R \to \{0, 1\}$$

Fuzzy $P[\![\cdot]\!]$: $P[\![\cdot]\!]$ 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;

Crisp $C[\![\cdot]\!]$: $C[\![\cdot]\!]$ maps each class $c \in \Delta_C$ into a subset $C[\![c]\!] \subseteq \Delta_R$, *i.e.* assigns a set of resources to every resource denoting a class; *i.e.*

$$C[\![c]\!] : \Delta_R \to \{0, 1\}$$

Fuzzy $C[\![\cdot]\!]$: $C[\![\cdot]\!]$ 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

Crisp $P[\![\cdot]\!]$: $P[\![\cdot]\!]$ maps each property name $p \in \Delta_P$ into a subset $P[\![p]\!] \subseteq \Delta_R \times \Delta_R$, *i.e.* assigns an extension to each property name; *i.e.*

$$P[\![p]\!] : \Delta_R \times \Delta_R \to \{0, 1\}$$

Fuzzy $P[\![\cdot]\!]$: $P[\![\cdot]\!]$ 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;

Crisp $C[\![\cdot]\!]$: $C[\![\cdot]\!]$ maps each class $c \in \Delta_C$ into a subset $C[\![c]\!] \subseteq \Delta_R$, *i.e.* assigns a set of resources to every resource denoting a class; *i.e.*

$$C[\![c]\!] : \Delta_R \to \{0, 1\}$$

Fuzzy $C[\![\cdot]\!]$: $C[\![\cdot]\!]$ 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

Models (Intuitively)

Crisp RDF : For ground triples, $\mathcal{I} \models (s, p, o)$ if

- p is interpreted as a property name
- s and o are interpreted as resources
- the interpretation of the pair (s, o) belongs to the extension of the property assigned to p

Fuzzy RDF : For ground triples, $\mathcal{I} \models (s, p, o)[n]$ if

- p is interpreted as a property name
- s and o are interpreted as resources
- the interpretation of the pair (s, o) belongs to the extension of the property assigned to p to degree not less than n

Models (Intuitively)

Crisp RDF : For ground triples, $\mathcal{I} \models (s, p, o)$ if

- p is interpreted as a property name
- s and o are interpreted as resources
- the interpretation of the pair (s, o) belongs to the extension of the property assigned to p

Fuzzy RDF : For ground triples, $\mathcal{I} \models (s, p, o)[n]$ if

- p is interpreted as a property name
- s and o are interpreted as resources
- the interpretation of the pair (s, o) belongs to the extension of the property assigned to p to degree not less than n

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 $\rho\mathrm{df} \cup \mathit{universe}(G)$
  - $\mathcal{I}$ satisfies the following conditions:

Crisp Simple:

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

Fuzzy Simple:

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

Crisp Subclass:

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

Fuzzy Subclass:

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

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

Corresponds to compute the degree of subsumption among classes

Crisp Simple:

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

Fuzzy Simple:

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

Crisp Subclass:

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

Fuzzy Subclass:

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

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

Corresponds to compute the degree of subsumption among classes

Crisp Simple:

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

Fuzzy Simple:

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

Crisp Subclass:

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

Fuzzy Subclass:

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

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

Corresponds to compute the degree of subsumption among classes

Crisp Simple:

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

Fuzzy Simple:

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

Crisp Subclass:

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

Fuzzy Subclass:

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

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

Corresponds to compute the degree of subsumption among classes

Crisp Subproperty:

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

Fuzzy 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

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

Corresponds to compute the degree of subsumption among properties

Crisp Subproperty:

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

Fuzzy 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

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

Corresponds to compute the degree of subsumption among properties

Crisp Typing I:

1. $x \in C[\![c]\!]$ iff $(x, c) \in P[\![\text{type}^{\mathcal{I}}]\!]$;
2. if $(p, c) \in P[\![\text{dom}^{\mathcal{I}}]\!]$ and $(x, y) \in P[\![p]\!]$ then $x \in C[\![c]\!]$;
3. if $(p, c) \in P[\![\text{range}^{\mathcal{I}}]\!]$ and $(x, y) \in P[\![p]\!]$ then $y \in C[\![c]\!]$;

Fuzzy 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

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

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

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

Crisp Typing I:

1. $x \in C[\![c]\!]$ iff $(x, c) \in P[\![\text{type}^{\mathcal{I}}]\!]$;
2. if $(p, c) \in P[\![\text{dom}^{\mathcal{I}}]\!]$ and $(x, y) \in P[\![p]\!]$ then $x \in C[\![c]\!]$;
3. if $(p, c) \in P[\![\text{range}^{\mathcal{I}}]\!]$ and $(x, y) \in P[\![p]\!]$ then $y \in C[\![c]\!]$;

Fuzzy 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

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

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

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

Crisp Typing II:

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

Fuzzy Typing II:

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

Crisp Typing II:

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

Fuzzy Typing II:

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

Models (cont.)

- 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

$$\forall x.c(x) \Rightarrow d(x) \; ,$$

- The fuzzyfication is

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

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

$$\forall x \forall y.p(x,y) \Rightarrow c(x) \; ,$$

- The fuzzyfication is

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

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

Proposition (Consistency)

Like crisp RDF, any fuzzy RDF graph has a model.

Deduction System for Fuzzy RDF

- The system is arranged in groups of rules that captures the semantic conditions of models
- In every rule, A, B, C, X, and Y are meta-variables representing elements in **UBL**
- An instantiation of a rule is a uniform replacement of the metavariables occurring in the triples of the rule by elements of **UBL**, such that all the triples obtained after the replacement are well formed

Deduction System for fuzzy RDF

1. Crisp/Fuzzy 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. Crisp Subproperty:

$$(a) \quad \frac{(A, \mathrm{sp}, B), (B, \mathrm{sp}, C)}{(A, \mathrm{sp}, C)} \qquad (b) \quad \frac{(A, \mathrm{sp}, B), (X, A, Y)}{(X, B, Y)}$$

3. Fuzzy Subproperty:

$$(a) \quad \frac{(A, \mathrm{sp}, B)[n], (B, \mathrm{sp}, C)[m]}{(A, \mathrm{sp}, C)[n \otimes m]} \qquad (b) \quad \frac{(A, \mathrm{sp}, B)[n], (X, A, Y)[m]}{(X, B, Y)[n \otimes m]}$$

Deduction System for fuzzy RDF

1. Crisp/Fuzzy 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. Crisp Subproperty:

$$(a) \quad \frac{(A, \mathrm{sp}, B), (B, \mathrm{sp}, C)}{(A, \mathrm{sp}, C)} \qquad (b) \quad \frac{(A, \mathrm{sp}, B), (X, A, Y)}{(X, B, Y)}$$

3. Fuzzy Subproperty:

$$(a) \quad \frac{(A, \mathrm{sp}, B)[n], (B, \mathrm{sp}, C)[m]}{(A, \mathrm{sp}, C)[n \otimes m]} \qquad (b) \quad \frac{(A, \mathrm{sp}, B)[n], (X, A, Y)[m]}{(X, B, Y)[n \otimes m]}$$

Deduction System for fuzzy RDF

1. Crisp/Fuzzy Simple:

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

2. Crisp Subproperty:

   $$(a) \quad \frac{(A, \mathrm{sp}, B), (B, \mathrm{sp}, C)}{(A, \mathrm{sp}, C)} \qquad (b) \quad \frac{(A, \mathrm{sp}, B), (X, A, Y)}{(X, B, Y)}$$

3. Fuzzy Subproperty:

   $$(a) \quad \frac{(A, \mathrm{sp}, B)[n], (B, \mathrm{sp}, C)[m]}{(A, \mathrm{sp}, C)[n \otimes m]} \qquad (b) \quad \frac{(A, \mathrm{sp}, B)[n], (X, A, Y)[m]}{(X, B, Y)[n \otimes m]}$$

1. Crisp Subclass:

$$(a) \quad \frac{(A, \mathrm{sc}, B), (B, \mathrm{sc}, C)}{(A, \mathrm{sc}, C)} \qquad (b) \quad \frac{(A, \mathrm{sc}, B), (X, \mathrm{type}, A)}{(X, \mathrm{type}, B)}$$

2. Fuzzy Subclass:

$$(a) \quad \frac{(A, \mathrm{sc}, B)[n], (B, \mathrm{sc}, C)[m]}{(A, \mathrm{sc}, C)[n \otimes m]} \qquad (b) \quad \frac{(A, \mathrm{sc}, B)[n], (X, \mathrm{type}, A)[m]}{(X, \mathrm{type}, B)[n \otimes m]}$$

3. Crisp Typing:

$$(a) \quad \frac{(A, \mathrm{dom}, B), (X, A, Y)}{(X, \mathrm{type}, B)} \qquad (b) \quad \frac{(A, \mathrm{range}, B), (X, A, Y)}{(Y, \mathrm{type}, B)}$$

4. Fuzzy Typing:

$$(a) \quad \frac{(A, \mathrm{dom}, B)[n], (X, A, Y)[m]}{(X, \mathrm{type}, B)[n \otimes m]} \qquad (b) \quad \frac{(A, \mathrm{range}, B)[n], (X, A, Y)[m]}{(Y, \mathrm{type}, B)[n \otimes m]}$$

1. Crisp Subclass:

$$(a)\quad \frac{(A, \mathrm{sc}, B), (B, \mathrm{sc}, C)}{(A, \mathrm{sc}, C)} \qquad (b)\quad \frac{(A, \mathrm{sc}, B), (X, \mathrm{type}, A)}{(X, \mathrm{type}, B)}$$

2. Fuzzy Subclass:

$$(a)\quad \frac{(A, \mathrm{sc}, B)[n], (B, \mathrm{sc}, C)[m]}{(A, \mathrm{sc}, C)[n \otimes m]} \qquad (b)\quad \frac{(A, \mathrm{sc}, B)[n], (X, \mathrm{type}, A)[m]}{(X, \mathrm{type}, B)[n \otimes m]}$$

3. Crisp Typing:

$$(a)\quad \frac{(A, \mathrm{dom}, B), (X, A, Y)}{(X, \mathrm{type}, B)} \qquad (b)\quad \frac{(A, \mathrm{range}, B), (X, A, Y)}{(Y, \mathrm{type}, B)}$$

4. Fuzzy Typing:

$$(a)\quad \frac{(A, \mathrm{dom}, B)[n], (X, A, Y)[m]}{(X, \mathrm{type}, B)[n \otimes m]} \qquad (b)\quad \frac{(A, \mathrm{range}, B)[n], (X, A, Y)[m]}{(Y, \mathrm{type}, B)[n \otimes m]}$$

1. Crisp Subclass:

$$(a) \quad \frac{(A, \mathrm{sc}, B), (B, \mathrm{sc}, C)}{(A, \mathrm{sc}, C)} \qquad (b) \quad \frac{(A, \mathrm{sc}, B), (X, \mathrm{type}, A)}{(X, \mathrm{type}, B)}$$

2. Fuzzy Subclass:

$$(a) \quad \frac{(A, \mathrm{sc}, B)[n], (B, \mathrm{sc}, C)[m]}{(A, \mathrm{sc}, C)[n \otimes m]} \qquad (b) \quad \frac{(A, \mathrm{sc}, B)[n], (X, \mathrm{type}, A)[m]}{(X, \mathrm{type}, B)[n \otimes m]}$$

3. Crisp Typing:

$$(a) \quad \frac{(A, \mathrm{dom}, B), (X, A, Y)}{(X, \mathrm{type}, B)} \qquad (b) \quad \frac{(A, \mathrm{range}, B), (X, A, Y)}{(Y, \mathrm{type}, B)}$$

4. Fuzzy Typing:

$$(a) \quad \frac{(A, \mathrm{dom}, B)[n], (X, A, Y)[m]}{(X, \mathrm{type}, B)[n \otimes m]} \qquad (b) \quad \frac{(A, \mathrm{range}, B)[n], (X, A, Y)[m]}{(Y, \mathrm{type}, B)[n \otimes m]}$$

1. Crisp Implicit Typing:

$$(a) \quad \frac{(A, \text{dom}, B), (C, \text{sp}, A), (X, C, Y)}{(X, \text{type}, B)} \qquad (b) \quad \frac{(A, \text{range}, B), (C, \text{sp}, A), (X, C, Y)}{(Y, \text{type}, B)}$$

2. Fuzzy Implicit Typing:

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

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

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 $(2 \leqslant j \leqslant k)$ 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)

$$G = \{(audiTT, \mathrm{type}, SportsCar)[0.8], (SportsCar, \mathrm{sc}, PassengerCar)[0.9]\}$$

t-norm: Product

Let us proof that

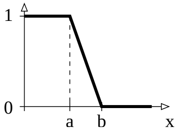
$$G \models (audiTT, \mathrm{type}, PassengerCar)[0.72]$$

G	$\vdash$	$(audiTT, \mathrm{type}, SportsCar)[0.8]$,	(1)	Rule Simple (b)
G	$\vdash$	$(SportsCar, \mathrm{sc}, PassengerCar)[0.9]$	(2)	Rule Simple (b)
G	$\vdash$	$(audiTT, \mathrm{type}, PassengerCar)[0.72]$	(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 rule application
- Query example: “find cheap sports cars”

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



where *e.g.* $\text{cheap}(p) = ls(30000, 50000)(p)$

- Conjunctive query: extends a crisp RDF query and is of the form

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

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* $p_j(\mathbf{t}_j) \in [0, 1]_m$. 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 $f\colon ([0, 1])^{n+h} \to [0, 1]$, 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 $\mathbf{v} \leqslant \mathbf{v}'$, it holds $f(\mathbf{v}) \leqslant f(\mathbf{v}')$, where $(v_1, \ldots, v_{n+h}) \leqslant (v'_1, \ldots, v'_{n+h})$ iff $v_i \leqslant v'_i$ 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.

Fuzzy RDF Query Answering (cont.)

- We will also write a query as

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

  where
  - $\varphi(\mathbf{x}, \mathbf{y})$ is $\tau_1[s_1], \ldots, \tau_n[s_n], s = f(\mathbf{s}, p_1(\mathbf{z}_1), \ldots, p_h(\mathbf{z}_h))$
  - $\mathbf{s} = \langle s_1, \ldots, s_n \rangle$
- 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

$$\begin{array}{lcl} q(x)[s] & \leftarrow & (x, \text{type}, \text{SportCar})[s_1], (x, \text{hasPrice}, y), s = s_1 \cdot \text{cheap}(y) \\ q(x)[s] & \leftarrow & (x, \text{type}, \text{PassengerCar})[s_1], s = s_1 \end{array}$$

  has intended meaning to retrieve all sports cars or passenger cars

Fuzzy RDF Query Answering (cont.)

- Consider a fuzzy graph G, a query $q(\mathbf{x})[s] \leftarrow \exists \mathbf{y}.\varphi(\mathbf{x}, \mathbf{y})[\mathbf{s}]$, and a vector $\mathbf{t}$ of terms in **UL** and $s \in [0, 1]$
- We say that $q(\mathbf{t})[s]$ is entailed by G, denoted $G \models q(\mathbf{t})[s]$, 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 $\varphi(\mathbf{t}, \mathbf{t}')[\mathbf{s}]$ (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 $q(\mathbf{t})[s]$ is entailed by G, denoted $G \models \mathbf{q}(\mathbf{t})[s]$, iff $G \models q_i(\mathbf{t})[s]$ for some $q_i \in \mathbf{q}$
- We say that s is *tight* iff $s = \sup\{s' \mid G \models \mathbf{q}(\mathbf{t})[s']\}$
- If $G \models \mathbf{q}(\mathbf{t})[s]$ and s is tight then $\mathbf{t}[s]$ is called an *answer* to $\mathbf{q}$
- The answer set of $\mathbf{q}$ w.r.t. G is defined as

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

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

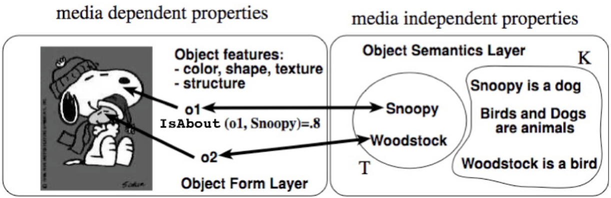
$$ans_k(G, \mathbf{q}) = \mathrm{Top}_k\ ans(G, \mathbf{q}) .$$

Fuzzy RDF Query Answering (cont.)

- A simple query answering procedure is the following:
  - Represent fuzzy triples as reified RDF triples
  - 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
- System has been implemented:
  - Using Java, Jena, TDB, MonetDB (each property is a table)
- Alternative implementation based on Logic Programming on the way
  - SWI-Prolog (XSB may work as well)
  - Top-k retrieval may be an issue . . .

Example:

RDF-based Multimedia Information Retrieval (based on [MSS01])



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

Consider the query

$$q(x)[s] \quad \leftarrow \quad (x, IsAbout, y)[s_1], (y, \mathrm{type}, SmallAnimal)[s_2], s = s_1 \cdot s_2$$

Then (under any t-norm)

$$ans(G, q) \quad = \quad \{o1[0.32], o2[0.63]\}, \quad ans_1(G, q) = \{o2[0.63]\}$$

Summary & Outlook

- We have presented Fuzzy RDF:
  - Conservative extension of RDF
  - Deductive system generalizes crisp RDF
  - Conservative extension of conjunctive query answering
  - Implementation relatively easy (prototype already available)
- Future issues:
  - Conservative extension of SPARQL to fuzzy case
    - SPARQL can already query fuzzy RDF data via reification, but not elegant at all . . .
  - Generalize Fuzzy RDF to arbitrary truth spaces
    - Allows to deal with temporal extensions, trustiness, confidence values, etc.

Questions ? Ask him . . .

Petr Hájek.
Metamathematics of Fuzzy Logic.
Kluwer, 1998.

Sergio Muñoz, Jorge Pérez, and Claudio Gutiérrez.
Minimal deductive systems for rdf.
In *4th European Semantic Web Conference (ESWC-07)*, number 4519 in Lecture Notes in Computer Science, pages 53–67. Springer Verlag, 2007.

Carlo Meghini, Fabrizio Sebastiani, and Umberto Straccia.
A model of multimedia information retrieval.
Journal of the ACM, 48(5):909–970, 2001.

L. A. Zadeh.
Fuzzy sets.
Information and Control, 8(3):338–353, 1965.