Representing Fuzzy Ontologies in OWL 2

Fernando Bobillo

fbobillo@unizar.es

Department of Computer Science and Systems Engineering
University of Zaragoza, Spain

Joint research with U. Straccia (ISTI-CNR, Pisa, Italy)

FUZZ-IEEE 2010
Barcelona (Spain)
July 2010

Outline

1. Introduction
2. The Fuzzy DL $\mathcal{SROIQ}(\mathbf{D})$
3. Representing Fuzzy Ontologies using OWL 2
4. Related Work
5. Conclusions and Future Work

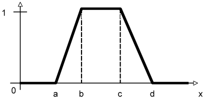
Outline

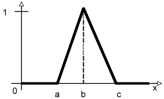
Motivation

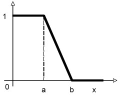
- Fuzzy ontologies emerge as useful in several applications.
  - Several extension of Description Logics (DLs) can be found.
  - Some **fuzzy DL reasoners** have been implemented, such as FUZZYDL, DELOREAN, and FIRE.
  - Not surprisingly, **each reasoner uses its own language** for representing fuzzy ontologies and, thus, there is a need for a standard way to represent such information.
- In this work, as we do not expect a fuzzy OWL extension to become a W3C proposed standard in the near future, we identify the **syntactic differences** that a fuzzy ontology language has to cope with, and **propose to use OWL 2** itself to represent them.
- More precisely, **we use OWL 2 annotation properties** to encode fuzzy $\mathcal{SROIQ}(\mathbf{D})$ ontologies, making it possible:
  - To use current OWL 2 editors for fuzzy ontology representation.
  - OWL 2 reasoners discard the fuzzy part of a fuzzy ontology, producing the same results as if would not exist.

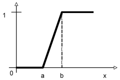
Outline

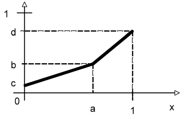
Fuzzy concrete domains and fuzzy modifiers











- **Fuzzy concrete domains**. A pair $\langle \Delta_{\mathbf{D}}, \Phi_{\mathbf{D}} \rangle$, a concrete interpretation domain $\Delta_{\mathbf{D}}$, and fuzzy concrete predicates $\mathbf{d} \in \Phi_{\mathbf{D}}$:

$$
\begin{array}{lrcl}
\mathbf{d} \rightarrow & \texttt{left}(k_1, k_2, a, b) & | & \text{(D1)} \\
 & \texttt{right}(k_1, k_2, a, b) & | & \text{(D2)} \\
 & \texttt{triangular}(k_1, k_2, a, b, c) & | & \text{(D3)} \\
 & \texttt{trapezoidal}(k_1, k_2, a, b, c, d) & & \text{(D4)}
\end{array}
$$

- **Fuzzy modifiers** A function $f_{mod} : [0, 1] \rightarrow [0, 1]$ applies to a fuzzy set to change its membership function:

$$
\begin{array}{lrcl}
mod \rightarrow & \texttt{linear}(c) & | & \text{(M1)} \\
 & \texttt{triangular}(a, b, c) & & \text{(M2)}
\end{array}
$$

Fuzzy concepts

Concepts	Syntax (C)	Semantics of $C^{\mathcal{I}}(x)$
(C1)	A	$A^{\mathcal{I}}(x)$
(C2)	$\top$	1
(C3)	$\bot$	0
(C4)	$C \sqcap_X D$	$C^{\mathcal{I}}(x) \otimes_X D^{\mathcal{I}}(x)$
(C5)	$C \sqcup_X D$	$C^{\mathcal{I}}(x) \oplus_X D^{\mathcal{I}}(x)$
(C6)	$\neg_X C$	$\ominus_X C^{\mathcal{I}}(x)$
(C7)	$\forall_X R.C$	$\inf_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x, y) \Rightarrow_X C^{\mathcal{I}}(y)\}$
(C8)	$\exists_X R.C$	$\sup_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x, y) \otimes_X C^{\mathcal{I}}(y)\}$
(C9)	$\forall_X T.\mathbf{d}$	$\inf_{v \in \Delta_{\mathbf{D}}} \{T^{\mathcal{I}}(x, v) \Rightarrow_X \mathbf{d}_{\mathbf{D}}(v)\}$
(C10)	$\exists_X T.\mathbf{d}$	$\sup_{v \in \Delta_{\mathbf{D}}} \{T^{\mathcal{I}}(x, v) \otimes_X \mathbf{d}_{\mathbf{D}}(v)\}$
(C11)	$\{\alpha / a\}$	α if $x = o_i^{\mathcal{I}}$, 0 otherwise
(C12)	$\geq_X m\ S.C$	$\sup_{y_1, \ldots, y_m \in \Delta^{\mathcal{I}}} (\min_{i=1}^{m} \{S^{\mathcal{I}}(x, y_i) \otimes_X C^{\mathcal{I}}(y_i)\}) \bigotimes_X ((\otimes_X)_{1 \leq j < k \leq m} \{y_j \neq y_k\})$
(C13)	$\leq_X n\ S.C$	$\inf_{y_1, \ldots, y_{n+1} \in \Delta^{\mathcal{I}}} (\min_{i=1}^{n+1} \{S^{\mathcal{I}}(x, y_i) \otimes_X C^{\mathcal{I}}(y_i)\}) \Rightarrow_X ((\oplus_X)_{1 \leq j < k \leq n+1} \{y_j = y_k\})$
(C14)	$\geq_X m\ T.\mathbf{d}$	$\sup_{v_1, \ldots, v_m \in \Delta_{\mathbf{D}}} (\min_{i=1}^{m} \{T^{\mathcal{I}}(x, v_i) \otimes_X \mathbf{d}_{\mathbf{D}}(v_i)\}) \bigotimes_X ((\otimes_X)_{j < k} \{v_j \neq v_k\})$
(C15)	$\leq_X n\ T.\mathbf{d})$	$\inf_{v_1, \ldots, v_{n+1} \in \Delta_{\mathbf{D}}} (\min_{i=1}^{n+1} \{T^{\mathcal{I}}(x, v_i) \otimes_X \mathbf{d}_{\mathbf{D}}(v_i)\}) \Rightarrow_X ((\oplus_X)_{j < k} \{v_j = v_k\})$
(C16)	$\exists S.\mathtt{Self}$	$S^{\mathcal{I}}(x, x)$
(C17)	$C \rightarrow_X D$	$C^{\mathcal{I}}(x) \Rightarrow_X D^{\mathcal{I}}(x)$
(C18)	$mod(C)$	$f_{mod}(C^{\mathcal{I}}(x))$
(C19)	$[C \geq \alpha]$	1 if $C^{\mathcal{I}}(x) \geq \alpha$, 0 otherwise
(C20)	$[C \leq \alpha]$	1 if $C^{\mathcal{I}}(x) \leq \alpha$, 0 otherwise
(C21)	$\alpha \cdot C$	$\alpha \cdot C^{\mathcal{I}}(x)$

Fuzzy roles and axioms

Roles	**Syntax (R)**	**Semantics of $R^{\mathcal{I}}(x, y)$**
(R1)	R_A	$R_A^{\mathcal{I}}(x, y)$
(R2)	R^-	$R^{\mathcal{I}}(y, x)$
(R3)	U	1
(R4)	$mod(R)$	$f_{mod}(R^{\mathcal{I}}(x, y))$
(R5)	$[R \geq \alpha]$	1 if $R^{\mathcal{I}}(x, y) \geq \alpha$, 0 otherwise
Axiom	**Syntax (τ)**	**Semantics** ($\mathcal{I}$ satisfies τ if ...)
(A1)	$\langle a : C \bowtie \alpha \rangle$	$C^{\mathcal{I}}(a^{\mathcal{I}}) \bowtie \alpha$
(A2)	$\langle (a, b) : R \bowtie \alpha \rangle$	$R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \bowtie \alpha$
(A3)	$\langle (a, b) : \neg_X R \bowtie \alpha \rangle$	$\ominus_X R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \bowtie \alpha$
(A4)	$\langle (a, v) : T \bowtie \alpha \rangle$	$T^{\mathcal{I}}(a^{\mathcal{I}}, v_{\mathbf{D}}) \bowtie \alpha$
(A5)	$\langle (a, v) : \neg_X T \bowtie \alpha \rangle$	$\ominus_X T^{\mathcal{I}}(a^{\mathcal{I}}, v_{\mathbf{D}}) \bowtie \alpha$
(A6)	$\langle a \neq b \rangle$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$
(A7)	$\langle a = b \rangle$	$a^{\mathcal{I}} = b^{\mathcal{I}}$
(A8)	$\langle C \sqsubseteq_X D \triangleright \alpha \rangle$	$\inf_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) \Rightarrow_X D^{\mathcal{I}}(x)\} \triangleright \alpha$
(A9)	$\langle R_1 \dots R_n \sqsubseteq_X R \triangleright \alpha \rangle$	$\inf_{x_1, x_{n+1} \in \Delta^{\mathcal{I}}} \{\sup_{x_2 \dots x_n \in \Delta^{\mathcal{I}}} \{(R_1^{\mathcal{I}}(x_1, x_2) \otimes_X \cdots \otimes_X R_n^{\mathcal{I}}(x_n, x_{n+1})) \Rightarrow_X R^{\mathcal{I}}(x_1, x_{n+1})\}\} \triangleright \alpha$
(A10)	$\langle T_1 \sqsubseteq_X T_2 \triangleright \alpha \rangle$	$\inf_{x \in \Delta^{\mathcal{I}}, v \in \Delta_{\mathbf{D}}} \{T_1^{\mathcal{I}}(x, v) \Rightarrow_X T_2^{\mathcal{I}}(x, v)\} \triangleright \alpha$
(A11)	$\mathtt{trans}_X(R)$	$\forall x, y, z \in \Delta^{\mathcal{I}}, R^{\mathcal{I}}(x, z) \otimes_X R^{\mathcal{I}}(z, y) \leq R^{\mathcal{I}}(x, y)$
(A12)	$\mathtt{dis}_X(S_1, S_2)$	$\forall x, y \in \Delta^{\mathcal{I}}, S_1^{\mathcal{I}}(x, y) \otimes_X S_2^{\mathcal{I}}(x, y) = 0$
(A13)	$\mathtt{dis}_X(T_1, T_2)$	$\forall x \in \Delta^{\mathcal{I}}, v \in \Delta_{\mathbf{D}}, T_1^{\mathcal{I}}(x, v) \otimes_X T_2^{\mathcal{I}}(x, v) = 0$
(A14)	$\mathtt{ref}(R)$	$\forall x \in \Delta^{\mathcal{I}}, R^{\mathcal{I}}(x, x) = 1$
(A15)	$\mathtt{irr}(S)$	$\forall x \in \Delta^{\mathcal{I}}, S^{\mathcal{I}}(x, x) = 0$
(A16)	$\mathtt{sym}(R)$	$\forall x, y \in \Delta^{\mathcal{I}}, R^{\mathcal{I}}(x, y) = R^{\mathcal{I}}(y, x)$
(A17)	$\mathtt{asy}(S)$	$\forall x, y \in \Delta^{\mathcal{I}}$, if $S^{\mathcal{I}}(x, y) > 0$ then $S^{\mathcal{I}}(y, x) = 0$

Definable elements

- Definable concepts:
  - **Weighted sum**: $(\alpha_1 \cdot C_1) \sqcup_{Ł} \cdots \sqcup_{Ł} (\alpha_k \cdot C_k)$.
  - **Fuzzy one-of**: $\{\alpha_1/o_1\} \sqcup_G \{\alpha_2/o_2\} \sqcup_G \cdots \sqcup_G \{\alpha_k/o_k\}$.
- Definable axioms (given an R-implication):
  - **Concept equivalence**: $\langle C_1 \sqsubseteq_X C_2 \geq 1 \rangle$ and $\langle C_2 \sqsubseteq_X C_1 \geq 1 \rangle$.
  - **Disjoint concepts**: $\langle C_1 \sqcap_X \cdots \sqcap_X C_n \sqsubseteq_X \perp \geq 1 \rangle$.
  - **Role domain**: $\langle \exists_X R.\top \sqsubseteq_X C \geq 1 \rangle$.
  - **Role range**: $\langle \top \sqsubseteq_X \forall_X R.C \geq 1 \rangle$.
  - **Role functionality**: $\langle \top \sqsubseteq_X (\leq_X 1\ R.\top) \geq 1 \rangle$.
- Syntactic sugar (not assumed for similarity with OWL 2):
  - $\texttt{irr}(S) = \top \sqsubseteq_X \neg \exists S.\texttt{Self}$
  - $\texttt{trans}(R) = RR \sqsubseteq_X R$
  - $\texttt{sym}(R) = R \sqsubseteq_X R^-$

Outline

Intuitive idea

- The idea of our representation is to use an OWL 2 ontology, **extending their elements with annotation properties** of type `rdfs:comment`, representing the features of the fuzzy ontology that OWL 2 cannot directly encode.

Example

Consider the fuzzy concept assertion $\langle \texttt{paul}: \texttt{Tall} \geq 0.5 \rangle$.
To represent it in OWL 2, we consider the crisp assertion $\texttt{paul}: \texttt{Tall}$ as represented in OWL 2, i.e., `ClassAssertion(paul Tall)`.
Next, we add an annotation property stating ≥ 0.5 to it.

- It is worth to note that OWL 2 only provides for annotations on ontologies, axioms, and entities.
- OWL DL is less expressive and only provides for annotations on ontologies and entities.
- For the sake of clarity, we will combine OWL 2 abstract syntax (for OWL 2), and an XML syntax (for annotation properties).

Syntactic Requirements of Fuzzy Ontologies

We will summarize the syntactic differences between the fuzzy and non-fuzzy ontologies. There are 8 cases which are non-exclusive (cases 3–5 can occur simultaneously, as well as cases 7–8).

1. **Fuzzy datatypes** do not have an equivalence in OWL 2: (D1–D4).
2. **Fuzzy modifiers** do not have an equivalence in OWL 2: (M1–M2).
3. Some **fuzzy concepts** require a **fuzzy logic**: (C4–C10), (C12–C15), (C17).
4. Some **fuzzy concepts** require a **degree** of truth: (C11), (C21).
5. Some **fuzzy concepts** do not have an equivalence in OWL 2: (C17)–(C21).
6. Some **fuzzy roles** do not have an equivalence in OWL 2: (R4–R5).
7. Some **axioms** require an inequality sign and a **degree** of truth: (A1–(A5), (A8)–(A10).
8. Some **axioms** require a **fuzzy logic**: (A3), (A5), (A8–A13).

1. Representing fuzzy datatypes

Example

Represent the age of a young person as left$(0, 200, 10, 30)$. We use a datatype definition of base type xsd:nonNegativeInteger with range in $[0, 200]$:

```
DatatypeDefinition( YoungAge DatatypeRestriction(
  xsd:nonNegativeInteger
  xsd:minInclusive "0"^^xsd:integer
  xsd:maxInclusive "200"^^xsd:integer
) )
```

Then we add the following annotation property to it:

```
<fuzzyOwl2 fuzzyType="datatype">
  <Datatype type="leftshoulder" a="10" b="30" />
</fuzzyOwl2>
```

2. Representing fuzzy modifiers

- Our fuzzy modifiers have parameters a, b, c.
- They can be represented as in the previous case, without representing `xsd:minInclusive` and `xsd:maxInclusive`.
- The value of `fuzzyType` will be `modifier`, and there will be a tag `Modifier` with an attribute `type` (possible values `linear`, and `triangular`), and attributes `a`, `b`, `c`, depending on the type.

Example

We define the datatype `very`

```
DatatypeDefinition( very xsd:nonNegativeInteger )
```

Then, we add this annotation property to it:

```
<fuzzyOwl2 fuzzyType="modifier">
  <Modifier type="linear" c="0.8" />
</fuzzyOwl2>
```

3. Representing fuzzy concepts with a fuzzy logic

- An annotation in a named concept can specify the fuzzy logic.
  - Anonymous concept expressions must be explicitly named.
- The value of `fuzzyType` is `concept`.
- There is an optional tag `Logic` (possible values `goedel`, `lukasiewicz`, `product`, and `zadeh`). Default value: `goedel`.

Example (Concept $\{1/\texttt{germany}\} \sqcup_G \{0.67/\texttt{switzerland}\}$)

```
Class ( C Annotation ( rdfs:comment
  <fuzzyOwl2 fuzzyType="concept">
    <Logic>goedel</Logic>
  </fuzzyOwl2>
) )

EquivalentClasses( C ObjectUnionOf( Nom1 Nom2 ) )
```

4. Representing fuzzy concepts with a degree of truth

- An annotation in a named concept can specify the degree.
  - Anonymous concept expressions must be explicitly named.
- The value of `fuzzyType` is `concept`.
- There is an optional tag `Degree` (with attribute `value`).

Example (Concepts $\{1/\mathtt{germany}\}$, $\{0.67/\mathtt{switzerland}\}$)

```
Class ( Nom1 Annotation( rdfs:comment
  <fuzzyOwl2 fuzzyType="concept">
    <Degree value="1" />
  </fuzzyOwl2>
) ) Class ( Nom2 Annotation( rdfs:comment
  <fuzzyOwl2 fuzzyType="concept">
    <Degree value="0.67" />
  </fuzzyOwl2>
) ) EquivalentClasses( Nom1 ObjectOneOf ( germany ) )
EquivalentClasses( Nom2 ObjectOneOf ( switzerland ) )
```

5–6. Representing fuzzy concepts and roles without an equivalence in OWL

- We have **to create a new entity** (concept or role) denoting the elements, and to add an annotation property to it, describing the type of the constructor and the value of their parameters.

Example (very(R))

```
ObjectProperty ( veryR Annotation( rdfs:comment
  <fuzzyOwl2 fuzzyType="role">
    <Role type="modified" modifier="very" base="R" />
  </fuzzyOwl2>
) )
```

7–8. Representing fuzzy axioms

- Some axioms may require a fuzzy logic, an inequality sign, or a degree of truth.
- Similarly to cases 3–4, there are two optional tags:
  - `Degree`, with attributes `value` and `sign` (possible values `geq`, `gre`, `leq`, and `les`),
  - `Logic` (possible values `goedel`, `lukasiewicz`, `product`, or `zadeh`).

Example

Consider again the fuzzy concept assertion $\langle \texttt{paul} : \texttt{Tall} \geq 0.5 \rangle$.
We extend the OWL 2 axiom `ClassAssertion(paul Tall)` with the following annotation property:

```
<fuzzyOwl2 fuzzyType="axiom">
  <Degree sign="geq" value="0.5" />
  <Logic>lukasiewicz</Logic>
</fuzzyOwl2>
```

Outline

Related work

- An **OWL ontology** for fuzzy ontology representation using individuals to represent concepts, roles and axioms.
  - (Meta) logical problems, completely different and user-unfriendly way of modelling, and inefficient representation (it grows exponentially with the size of the ontology).
- **Fuzzy OWL and Fuzzy OWL 2**.
  - Obviously not complaint with OWL 2 and current ontology editors.
- Similar work covers just some of the cases:
  - A **pattern for uncertainty representation** in ontologies restricted to a subset of our case 7, axioms (A1).
  - **Probabilistic constraints** restricted to a subset of our case 7, axioms (A1) and (A8).
- **Crisp representations for fuzzy ontologies**.
  - Ok to reuse current DL reasoners, but not for modelling.

Outline

Conclusions and future work

- Our objective is **not to provide a standard** language for fuzzy ontology representation. This should involve the whole community.
- We identified the **syntactical differences** of a fuzzy ontology language and provided a **representation using OWL 2**.
  - A similar approach **cannot be represented in OWL DL** as it does not support rich enough annotation capabilities.
- Our logic is very expressive, but it is **extensible** and can easily be augmented to support more fuzzy logics, fuzzy predicates ...
- **Methodology for fuzzy ontology development**.
  - First, we can build the core part of the ontology as usual.
  - Then, we add the fuzzy part with annotation properties.
  - Non-fuzzy reasoners **discard the fuzzy part**.
- Parsers translating this representation into the syntax of some popular fuzzy DL reasoners (FUZZYDL and DELOREAN).
- In **future work**, we will develop a **graphical interface** (e.g. a Protégé plug-in) to make annotation properties transparent.

Comments?

Thank you very much for your attention