Representing Fuzzy Ontologies in OWL 2

Fernando Bobillo and Umberto Straccia

***Abstract*— The need to deal with vague information in Semantic Web languages is rising in importance and, thus, calls for a standard way to represent such information. We may address this issue by either extending current Semantic Web languages to cope with vagueness, or by providing a procedure to represent such information within current Semantic Web languages. In this work, we follow the latter approach, by identifying the syntactic differences that a fuzzy ontology language has to cope with, and by proposing a concrete methodology to represent fuzzy ontologies using OWL 2 annotation properties.**

I. Introduction

It is widely agreed that classical ontology languages are not appropriate to deal with vagueness or imprecision in the knowledge, which is inherent to most of the real world application domains [19]. Since fuzzy set theory and fuzzy logic [21] are suitable formalisms to handle these types of knowledge, fuzzy ontologies emerge as useful in several applications, ranging from (multimedia) information retrieval to image interpretation, ontology mapping, matchmaking, decision making, or the Semantic Web.

Description Logics (DLs) are the basis of several ontology languages. The current standard for ontology representation is OWL (Web Ontology Language), which comprises three sublanguages (OWL Lite, OWL DL and OWL Full). OWL 2 is a recent W3C recommendation [11]. The logical counterparts of OWL Lite, OWL DL and OWL 2 are the DLs $\mathcal{SHIF}(\mathbf{D})$, $\mathcal{SHOIN}(\mathbf{D})$, and $\mathcal{SROIQ}(\mathbf{D})$, respectively.

Several fuzzy extensions of DLs can be found in the literature (see the survey in [10]) and some fuzzy DL reasoners have been implemented, such as FUZZYDL [4], DELOREAN [1] and FIRE [12]. Not surprisingly, each reasoner uses its own fuzzy DL 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 fuzzy ontologies. More precisely, we use OWL 2 annotation properties to encode fuzzy $\mathcal{SROIQ}(\mathbf{D})$ ontologies. The use of annotation properties makes possible *(i)* to use current OWL 2 editors for fuzzy ontology representation, and *(ii)* that OWL 2 reasoners discard the fuzzy part of a fuzzy ontology, producing the same results as if would not exist.

The remainder of this paper is organized as follows. In Section II we present a fuzzy extension of DL $\mathcal{SROIQ}(\mathbf{D})$, the logic behind OWL 2, including some additional constructs, peculiar to fuzzy logic. Section III discusses how to encode it using OWL 2 language. Section IV compares our proposal with the related work. Finally, Section V sets out some conclusions and ideas for future research.

Fernando Bobillo is with the Dpt. of Computer Sciences & Systems Engineering, University of Zaragoza, Spain (email: fbobillo@unizar.es).
Umberto Straccia is with the Istituto di Scienza e Tecnologie dell'Informazione (ISTI-CNR), Pisa, Italy (email: straccia@isti.cnr.it).

II. The Fuzzy DL $\mathcal{SROIQ}(\mathbf{D})$

In this section we describe the fuzzy DL $\mathcal{SROIQ}(\mathbf{D})$, inspired by the logics presented in [3], [4], [18]. Here, concepts denote fuzzy sets of individuals and roles denote fuzzy binary relations. Axioms are also extended to the fuzzy case and some of them hold to a degree.

A. Syntax

To begin with, we will introduce two important elements of our logic: fuzzy concrete domains and fuzzy modifiers.

a) Fuzzy concrete domains: A *fuzzy concrete domain* [17] $\mathbf{D}$ is a pair $\langle \Delta_{\mathbf{D}}, \Phi_{\mathbf{D}} \rangle$, where $\Delta_{\mathbf{D}}$ is a concrete interpretation domain, and $\Phi_{\mathbf{D}}$ is a set of fuzzy concrete predicates $\mathbf{d}$ with an arity n and an interpretation $\mathbf{d}_{\mathbf{D}} : \Delta_{\mathbf{D}}^n \to [0,1]$, which is an n-ary fuzzy relation over $\Delta_{\mathbf{D}}$.

As fuzzy concrete predicates we allow the following functions defined over an interval $[k_1, k_2] \subseteq \mathbb{Q}$: trapezoidal membership function (Figure 1 (a)), the triangular (Figure 1 (b)), the left-shoulder function (Figure 1 (c)) and the right-shoulder function (Figure 1 (d)) [17]. Formally:

$$
\begin{array}{llll}
\mathbf{d} \to & \mathit{left}(k_1, k_2, a, b) & \mid & \text{(D1)} \\
& \mathit{right}(k_1, k_2, a, b) & \mid & \text{(D2)} \\
& \mathit{triangular}(k_1, k_2, a, b, c) & \mid & \text{(D3)} \\
& \mathit{trapezoidal}(k_1, k_2, a, b, c, d) & & \text{(D4)}
\end{array}
$$

Example 1: We may define the fuzzy datatype $\mathsf{YoungAge}\colon [0,200] \to [0,1]$, denoting the degree of a person being young, as $\mathsf{YoungAge}(x) = \mathit{left}(0, 200, 10, 30)$.

b) Fuzzy modifiers: A *fuzzy modifier* mod is a function $f_{mod}\colon [0,1] \to [0,1]$ which applies to a fuzzy set to change its membership function. We will allow modifiers defined in terms of linear hedges (Figure 1 (e)) and triangular functions (Figure 1 (b)) [17]. In linear modifiers, we assume that $a = c/(c+1), b = 1/(c+1)$. Formally:

$$
\begin{array}{llll}
mod \to & \mathit{linear}(c) & \mid & \text{(M1)} \\
& \mathit{triangular}(a, b, c) & & \text{(M2)}
\end{array}
$$

Example 2: Modifier very can be defined as $\mathit{linear}(0.8)$.

c) Symbols: Fuzzy $\mathcal{SROIQ}(\mathbf{D})$ assumes three alphabets of symbols, for (abstract and concrete) *fuzzy concepts*, *fuzzy roles* and *individuals*. The syntax of fuzzy concepts and roles is shown in Table II.

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

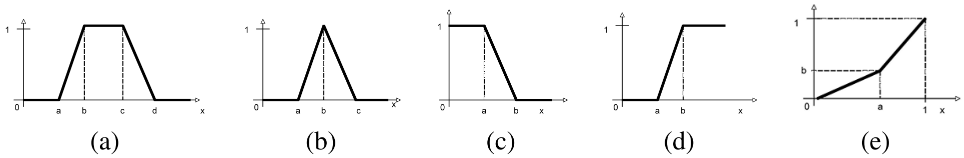

Fig. 1. (a) Trapezoidal function; (b) Triangular function; (c) L-function; (d) R-function; (e) Crisp interval; (f) Linear function.

Concept constructors (C1)–(C16) correspond to the concept constructors of crisp $\mathcal{SROIQ}(\mathbf{D})$. The only difference here are fuzzy nominals of the form $\{\alpha/a\}$ [2]. Concept constructors (C17)–(C21) are usual in the setting of fuzzy DLs, namely implication concepts (C17), modified concepts (C18), cut concepts (C19–C20), and weighted concepts (C21).

Example 3: Concept $\mathsf{Human} \sqcap \exists \mathsf{hasAge.YoungAge}$ denotes the fuzzy set of young humans. $\mathsf{very}(\mathsf{Human} \sqcap \exists \mathsf{hasAge.YoungAge})$ denotes *very* young humans.

Role constructors (R1)–(R3) correspond to the role constructors of crisp $\mathcal{SROIQ}(\mathbf{D})$. (R4) and (R5) correspond to modified roles and cut roles, respectively.

d) Notation: Let us introduce some notation that will be used in the rest of the paper:

- C, D are (possibly complex) fuzzy concepts,
- A is an atomic fuzzy concept,
- R is a (possibly complex) abstract fuzzy role,
- R_A is an atomic fuzzy role,
- S is a simple fuzzy role [1],
- T is a concrete fuzzy role,
- a, b are abstract individuals, v is a concrete individual,
- $\mathbf{d}$ is a fuzzy concrete predicate,
- n, m are natural numbers with $n \geqslant 0, m > 0$,
- mod is a fuzzy modifier,
- $\rhd \in \{\geq, >\}, \lhd \in \{\leq, <\}, \bowtie \in \{\geq, >, \leq, <\}$,
- $\alpha \in (0, 1]$,
- $\otimes, \oplus, \Rightarrow, \ominus$ are a t-norm, a t-conorm, an implication, and a negation function, respectively.
- $X \in \{Z, G, Ł, \Pi\}$ specifies a fuzzy logic (Zadeh, Gödel, Łukasiewicz and Product, respectively, see Table I).

TABLE I

THE MORE POPULAR FUZZY LOGICS.

	Łukasiewicz	Gödel	Product	Zadeh
$\alpha \otimes \beta$	$\max(\alpha + \beta - 1, 0)$	$\min(\alpha, \beta)$	$\alpha \cdot \beta$	$\min(\alpha, \beta)$
$\alpha \oplus \beta$	$\min(\alpha + \beta, 1)$	$\max(\alpha, \beta)$	$\alpha + \beta - \alpha \cdot \beta$	$\max(\alpha, \beta)$
$\alpha \Rightarrow \beta$	$\min(1 - \alpha + \beta, 1)$	1 if $\alpha \leq \beta$ β otherwise	$\min(1, \beta/\alpha)$	$\max(1 - \alpha, \beta)$
$\ominus \alpha$	$1 - \alpha$	1 if $\alpha = 0$ 0 otherwise	1 if $\alpha = 0$ 0 otherwise	$1 - \alpha$

A *Fuzzy Knowledge Base* (KB) contains a finite number of axioms. The axioms that are allowed in our logic are shown in Table II. They can be grouped into a fuzzy ABox with axioms (A1)–(A7), a fuzzy TBox with axioms (A8), and a fuzzy RBox with axioms (A8)–(A17). All the axioms have a equivalent in crisp $\mathcal{SROIQ}(\mathbf{D})$.

[1]Simple roles are needed to guarantee the decidability of the logic. Intuitively, simple roles cannot take part in cyclic role inclusion axioms (see [2] for a formal definition).

Example 4: The fuzzy concept assertion $\langle \mathsf{paul} \colon \mathsf{Tall} \geq 0.5 \rangle$ states that Paul is tall with at least degree 0.5. The fuzzy RIA $\langle \mathsf{isFriendOf}\ \mathsf{isFriendOf} \sqsubseteq_L \mathsf{isFriendOf} \geq 0.75 \rangle$ states that the friends of my friends can also be considered as my friends with at least degree 0.75.

The pair of axioms $\langle \tau \geq \alpha \rangle$ and $\langle \tau \leq \alpha \rangle$ is abbreviated as $\langle \tau = \alpha \rangle$. Also, an axiom of the form $\langle \tau \geq 1 \rangle$ is usually abbreviated as simply τ.

B. Semantics

e) Fuzzy interpretation: A fuzzy interpretation $\mathcal{I}$ with respect to a fuzzy concrete domain $\mathbf{D}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a non empty set $\Delta^{\mathcal{I}}$ (the interpretation domain) disjoint with $\Delta_{\mathbf{D}}$ and a fuzzy interpretation function $\cdot^{\mathcal{I}}$ mapping:

- A fuzzy *abstract individual* a onto an element $a^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$.
- A fuzzy *concrete individual* v onto an element $v_{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}$.
- A fuzzy *concept* C onto a function $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0, 1]$.
- A fuzzy *abstract role* R onto a function $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0, 1]$.
- A fuzzy *concrete role* T onto a function $T^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}} \to [0, 1]$.
- An n-ary fuzzy *concrete predicate* $\mathbf{d}$ onto the fuzzy relation $\mathbf{d}_{\mathbf{D}} : \Delta^n_{\mathbf{D}} \to [0, 1]$.
- A fuzzy *modifier* mod onto a function $f_{mod} : [0, 1] \to [0, 1]$.

$C^{\mathcal{I}}$ (resp. $R^{\mathcal{I}}$) denotes the membership function of the fuzzy concept C (resp. fuzzy role R) w.r.t. $\mathcal{I}$. $C^{\mathcal{I}}(a)$ (resp. $R^{\mathcal{I}}(a, b)$) gives us to what extent the individual a can be considered as an element of the fuzzy concept C (resp. to what extent (a, b) can be considered as an element of the fuzzy role R) under the fuzzy interpretation $\mathcal{I}$.

The fuzzy interpretation function is extended to complex fuzzy concepts, complex fuzzy roles and fuzzy axioms as shown in Table II. We say that a fuzzy interpretation $\mathcal{I}$ satisfies a fuzzy KB $\mathcal{K}$ iff $\mathcal{I}$ satisfies each element in $\mathcal{K}$.

There are several reasoning tasks in fuzzy $\mathcal{SROIQ}(\mathbf{D})$ (e.g., fuzzy KB satisfiability, fuzzy concept satisfiability, fuzzy entailment, fuzzy concept subsumption, greatest lower bound, etc.) [15], [18]. However, they are part of the query language and not of the representation language. Thus, we shall not represent them in a fuzzy ontology.

TABLE II

SYNTAX AND SEMANTICS OF THE FUZZY DL $\mathcal{SROIQ}(\mathbf{D})$.

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)	$\exists S.\texttt{Self}$	$S^{\mathcal{I}}(x,x)$
(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)	$\alpha \cdot C$	$\alpha \cdot C^{\mathcal{I}}(x)$
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)	$\langle C \sqsubseteq_X D \rhd \alpha \rangle$	$\inf_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) \Rightarrow_X D^{\mathcal{I}}(x)\} \rhd \alpha$
(A9)	$\langle R_1 \ldots R_n \sqsubseteq_X R \rhd \alpha \rangle$	$\inf_{x_1, x_{n+1} \in \Delta^{\mathcal{I}}} \{\sup_{x_2 \ldots x_n \in \Delta^{\mathcal{I}}} \{(R_1^{\mathcal{I}}(x_1,x_2) \otimes_X \cdots \otimes_X R_n^{\mathcal{I}}(x_n,x_{n+1})) \Rightarrow_X R^{\mathcal{I}}(x_1,x_{n+1})\}\} \rhd \alpha$
(A10)	$\langle T_1 \sqsubseteq_X T_2 \rhd \alpha \rangle$	$\inf_{x \in \Delta^{\mathcal{I}}, v \in \Delta_{\mathbf{D}}} \{T_1^{\mathcal{I}}(x,v) \Rightarrow_X T_2^{\mathcal{I}}(x,v)\} \rhd \alpha$
(A11)	$\texttt{trans}_X(R)$	$\forall x,y,z \in \Delta^{\mathcal{I}}, R^{\mathcal{I}}(x,z) \otimes_X R^{\mathcal{I}}(z,y) \leq R^{\mathcal{I}}(x,y)$
(A12)	$\texttt{dis}_X(S_1,S_2)$	$\forall x,y \in \Delta^{\mathcal{I}}, S_1^{\mathcal{I}}(x,y) \otimes_X S_2^{\mathcal{I}}(x,y) = 0$
(A13)	$\texttt{dis}_X(T_1,T_2)$	$\forall x \in \Delta^{\mathcal{I}}, v \in \Delta_{\mathbf{D}}, T_1^{\mathcal{I}}(x,v) \otimes_X T_2^{\mathcal{I}}(x,v) = 0$
(A14)	$\texttt{ref}(R)$	$\forall x \in \Delta^{\mathcal{I}}, R^{\mathcal{I}}(x,x) = 1$
(A15)	$\texttt{irr}(S)$	$\forall x \in \Delta^{\mathcal{I}}, S^{\mathcal{I}}(x,x) = 0$
(A16)	$\texttt{sym}(R)$	$\forall x,y \in \Delta^{\mathcal{I}}, R^{\mathcal{I}}(x,y) = R^{\mathcal{I}}(y,x)$
(A17)	$\texttt{asy}(S)$	$\forall x,y \in \Delta^{\mathcal{I}}$, if $S^{\mathcal{I}}(x,y) > 0$ then $S^{\mathcal{I}}(y,x) = 0$

C. Definable elements.

The following concepts can be defined using the concept constructors of our logic:

- Weighted sum: $(\alpha_1 \cdot C_1) + \cdots + (\alpha_k \cdot C_k) = (\alpha_1 \cdot C_1) \sqcup_{\text{Ł}} \cdots \sqcup_{\text{Ł}} (\alpha_k \cdot C_k)$, with $\sum_{i=1}^{k} \alpha_i \leq 1$.
- Fuzzy one-of: $\{\alpha_1/o_1, \alpha_2/o_2, \ldots, \alpha_k/o_k\} = \{\alpha_1/o_1\} \sqcup_G \{\alpha_2/o_2\} \sqcup_G \cdots \sqcup_G \{\alpha_k/o_k\}$.

Let X be a fuzzy logic with an R-implication. Then, there are also several interesting definable axioms [2]:

- *Concept equivalence*: C_1 and C_2 are the equivalent iff $\langle C_1 \sqsubseteq_X C_2 \geq 1 \rangle$ and $\langle C_2 \sqsubseteq_X C_1 \geq 1 \rangle$ hold.
- *Disjoint concepts*: $C_1, C_2, \ldots, C_n$ are mutually disjoint iff $\langle C_1 \sqcap_X \cdots \sqcap_X C_n \sqsubseteq_X \bot \geq 1 \rangle$ holds.

[2]Note that the semantics does not depend on the fuzzy logic. Since every R-implication verifies that $\alpha \Rightarrow \beta = 1$ iff $\alpha \leq \beta$, it only matters the partial order and not the exact value of the R-implication.

- *Role domain*: C is the domain of R iff $\langle \top \sqsubseteq_X \forall_X R^-.C \geq 1 \rangle$ hold, or, equivalently, iff $\langle \exists_X R.\top \sqsubseteq_X C \geq 1 \rangle$ holds.
- *Role range*: C is range of an abstract role R iff $\langle \top \sqsubseteq_X \forall_X R.C \geq 1 \rangle$ hold.
- *Role functionality*: An abstract role R is functional iff $\langle \top \sqsubseteq_X (\leq_X 1\ R.\top) \geq 1 \rangle$ hold.

Let X be a fuzzy logic with an R-implication. Then, some of the axioms of our logic are syntactic sugar (and consequently it could be assumed that they do not appear in fuzzy KBs, even if we do not assume this for similarity with OWL 2) due to the following equivalences [2]:

- $\texttt{irr}(S) = \top \sqsubseteq_X \neg \exists S.\texttt{Self}$,
- $\texttt{trans}(R) = RR \sqsubseteq_X R$,
- $\texttt{sym}(R) = R \sqsubseteq_X R^-$.

III. Representing Fuzzy Ontologies using OWL 2

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.

For the sake of clarity, we will use OWL 2 abstract syntax [11] for OWL 2, and an XML syntax to write the value of annotation properties[3].

Example 5: Consider the fuzzy concept assertion of Example 4, $\langle \mathsf{paul}\colon \mathsf{Tall} \geq 0.5\rangle$. To represent it in OWL 2, we consider the crisp assertion $\mathsf{paul}\colon \mathsf{Tall}$ as represented in OWL 2, `ClassAssertion(paul Tall)` and then we add an annotation property including the information ≥ 0.5 to it.

It is worth to note that OWL 2 only provides for annotations on ontologies, axioms, and entities [11].

A. Syntactic Requirements of Fuzzy Ontologies

To begin with, 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).

Case 1. Fuzzy datatypes do not have an equivalence in the non-fuzzy case: (D1)–(D4).

Case 2. Fuzzy modifiers do not have an equivalence in the non-fuzzy case: (M1), (M2).

Case 3. Some fuzzy concepts require a fuzzy logic: (C4)–(C10), (C12)–(C15), (C17).

Case 4. Some fuzzy concepts require a degree of truth: (C11), (C21).

Case 5. Some fuzzy concepts do not have an equivalence in the non-fuzzy case: (C17)–(C21).

Case 6. Some fuzzy roles do not have an equivalence in the non-fuzzy case: (R4), (R5).

Case 7. Some axioms require an inequality sign and a degree of truth: (A1–(A5), (A8)–(A10).

Case 8. Some axioms require a fuzzy logic: (A3), (A5), (A8)–(A13).

B. Representation in OWL 2

Now, we will see how to address each of the cases. Every annotation will be delimited by a start tag `<FuzzyOwl2>` and an end tag `</FuzzyOwl2>`. The tag has an attribute `fuzzyType`, specifying the fuzzy element being tagged.

Case 1. According to Section II-A, the fuzzy datatypes that we want to represent have parameters k_1, k_2, a, b, c, d. The first four parameters are common to all of them, c only appears in (D4), (D5); and d only appears in (D5).

Given a datatype restriction of the type base of the fuzzy datatype (e.g. integer or double), we represent $k1$ and $k2$ using the attributes `xsd:minInclusive` and `xsd:maxInclusive`, respectively.

[3] Of course, the final result depends on the syntax (for instance, in OWL 2 XML syntax the characters $\geq$ and $\leq$ of the annotations are escaped), but OWL 2 ontology editors make these issues transparent to the user.

Then, we use an annotation property defining the values of the remaining parameters. The value of `fuzzyType` is `datatype`, and there is a tag `Datatype` with an attribute `type` (possible values `leftshoulder`, `rightshoulder`, `triangular`, and `trapezoidal`), and attributes `a`, `b`, `c`, `d`, depending on the type of the datatype. Representing k_1 and k_2 is optional and, if they are not represented, then the minimum and maximum of the attributes is considered as default.

Example 6: Let us represent the fuzzy datatype $\mathit{left}(0, 200, 10, 30)$ denoting the age of a young person. This fuzzy datatype is represented using 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>
```

Case 2. According to Section II-A, the fuzzy modifiers that we want to represent have parameters a, b, c. Consequently, they can be represented as in the previous case, with the particularity that there is no need to use `xsd:minInclusive` and `xsd:maxInclusive`, since fuzzy modifiers do not have parameters k_1, k_2.

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 of the modifier.

Example 7: Relative to Example 2, we define the datatype very and add the following annotation property to it:

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

Cases 3–4. In these cases, it is possible to add an annotation to a concept, specifying the fuzzy logic (case 3) or the degree of truth (case 4).

The value of `fuzzyType` is `concept`. There are two optional tags `Degree` (with attribute `value`) and `Logic` (possible values `goedel`, `lukasiewicz`, `product`, and `zadeh`). The default values are `1` and `goedel`, respectively.

It is important to keep in mind that it is only possible to add annotation properties to entities (named concepts), and not to anonymous concept expressions. Hence, in order to add an annotation property to an anonymous concept expression, it is firstly mandatory to name it.

Example 8: Let us represent a fuzzy one-of concept denoting German-speaking countries: $\{1/\mathsf{germany}\} \sqcup_G \{1/\mathsf{austria}\} \sqcup_G \{0.67/\mathsf{switzerland}\}$.

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

```
Class ( Nom1 Annotation( rdfs:comment
  <fuzzyOwl2 fuzzyType="concept">
    <Degree value="1" />
  </fuzzyOwl2>
) )

Class ( Nom2 Annotation( rdfs:comment
  <fuzzyOwl2 fuzzyType="concept">
    <Degree value="1" />
  </fuzzyOwl2>
) )

Class ( Nom3 Annotation( rdfs:comment
  <fuzzyOwl2 fuzzyType="concept">
    <Degree value="0.67" />
  </fuzzyOwl2>
) )

EquivalentClasses( Nom1 ObjectOneOf ( germany ) )
EquivalentClasses( Nom2 ObjectOneOf ( austria ) )
EquivalentClasses( Nom3 ObjectOneOf ( switzerland ) )
EquivalentClasses ( C ObjectUnionOf ( Nom1 Nom2 Nom3 ) )
```

Cases 5–6. For the fuzzy concepts and roles that do not have an equivalence in the non-fuzzy case, the solution is to create a new entity (a concept or a role) denoting the elements, and to add an annotation property to the entity, describing the type of the constructor and the value of their parameters.

Now, the value of `fuzzyType` is `concept`, and there is a tag `Concept` with an attribute `type` (possible values `implication`, `modified`, `cut`, and `weighted`) and other attributes, depending on the value of `type`.

- In concept (C17), the type is `implication`. The additional parameters are `antecedent` and `consequent`. Recall that it can also have a fuzzy logic (case 3).
- In (C18), and (R4) the type is `modified`. The parameters are a fuzzy `modifier`, and `base` (modified concept / role).
- In (C19), (C20) and (R5) the type is `cut`. The parameters are `base` (a concept / role), `value` and `sign` (possible values `geq`, and `leq`).
- In (C21) the type is `weighted`. The parameters are `base` (a concept), and `value`.

Example 9: Let us represent $\mathsf{very}(\mathsf{C}) \rightarrow_G [\mathsf{D} \geq 0.25]$.

```
Class ( ModC Annotation( rdfs:comment
  <fuzzyOwl2 fuzzyType="concept">
    <Concept type="modified" modifier="very" base="C" />
  </fuzzyOwl2>
) )

Class ( CutD Annotation( rdfs:comment
  <fuzzyOwl2 fuzzyType="concept">
    <Concept type="cut" base="D" sign="geq" value="0.5" />
  </fuzzyOwl2>
) )

Class ( Imp Annotation( rdfs:comment
  <fuzzyOwl2 fuzzyType="concept">
    <Concept type="implication" antecedent="ModC"
      consequent="CutD" />
    <Logic>goedel</logic>
  </fuzzyOwl2>
) )
```

Cases 7–8. Some axioms may require a fuzzy logic, an inequality sign, or a degree of truth. The value of `fuzzyType` is `axiom`. Similarly to cases 3–4, there are two optional tags `Degree`, with attributes `value` and `sign` (possible values `geq`, `gre`, `leq`, and `les`), and `Logic` (possible values `goedel`, `lukasiewicz`, `product`, or `zadeh`). The default values are `1`, `geq`, and `goedel`, respectively.

Example 10: Let us consider again, in greater detail, Example 5. It can be represented by extending the OWL 2 axiom `ClassAssertion(paul Tall)` with the following annotation property:

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

IV. Related Work

This is, to the best of our knowledge, the first effort towards fuzzy ontology representation using OWL 2. A similar work provides an OWL ontology for fuzzy ontology representation [6]. There, annotation properties are not used, but concepts, roles and axioms are represented as individuals. For instance, Example 4 would be represented using the following axioms (in abstract syntax):

```
(ClassAssertion paul Individual)
(ClassAssertion tall Concept)
(ClassAssertion ax1 ConceptAssertion)
(ObjectPropertyAssertion ax1 isComposedOfAbstractIndividual)
(ObjectPropertyAssertion ax1 isComposedOfAbstractConcept)
```

However, this representation has many problems. Representing concepts, roles and axioms as individuals causes (meta)logical problems. Furthermore, instead of reusing current ontology editors, the method requires a completely different and user-unfriendly way of modelling, e.g., a concept conjunction is not represented using `intersectionOf`, but using a specific encoding using a individual (representing the concept) related with two individuals (each of them representing one of the conjuncts). Last but no least, it is not an efficient representation, since the ontology grows exponentially with the size of the ontology.

The W3C Uncertainty Reasoning for the World Wide Web Incubator Group (URW3-XG) defined an ontology of uncertainty, a vocabulary which can be used to annotate different pieces of information with different types of uncertainty (e.g. vagueness, randomness or incompleteness), the nature of the uncertainty, etc. [19]. But unlike our approach, it can only be used to identify some kind of uncertainty, and not to represent and manage uncertain pieces of information.

Naïve fuzzy extensions of ontology languages have been presented, more precisely OWL [7], [14] and OWL 2 [13], but they are obviously not complaint with OWL 2 and current ontology editors, as it happens under out approach. However, these works are interesting since they define the correspondence between OWL 2 and fuzzy DLs.

A pattern for uncertainty representation in ontologies has been presented in [20]. However, it is restricted to a subset of our case 7, only for axioms (A1). It also relies in OWL Full, thus not making possible to reason with the ontology.

A closer approach to ours is [9], which also uses annotation properties to add probabilistic constraints, but it is restricted to a subset of our case 7, axioms (A1) and (A8).

Finally, our approach should not be confused with a series of works that describe, given a fuzzy ontology, how to obtain an equivalent OWL 2 ontology (see for example [2], [3], [5], [13], [16]). In these works it is possible to reason using a crisp DL reasoner instead of a fuzzy DL reasoner, which is not our case. However, the advantage of our approach is that we provide a specific format to represent fuzzy ontologies which can be easily managed by current OWL editors and understood by humans.

V. Conclusions and Future Work

In this paper we have dealt with the problem of fuzzy ontology representation. Our objective is not to provide a standard language for fuzzy ontology representation, because we think that this process should involve the whole community and will be not available in the short term. On the contrary, we have identified the syntactical differences that a fuzzy ontology language has to cope with, and provided a representation using the language OWL 2.

Our work consider a very general fuzzy extension of the DL $\mathcal{SROIQ}(\mathbf{D})$. In fact, our logic is not restricted to a simple fuzzy ABox, but there are many differences with respect to the case, such as fuzzy datatypes, fuzzy modifiers or fuzzy implication concepts. We have simplified the definition of the logic by identifying some definable elements of the logic. Ideally, fuzzy ontology languages should provide some sort of syntactic sugar for them. Then, we have summarized the syntactic differences that fuzzy ontologies introduce, grouping them into 8 different cases, and we have provided a methodology to represent them using OWL 2 ontologies extended with annotation properties.

Our approach is extensible and can easily be augmented to support alternative fuzzy logics, modifier functions, fuzzy predicates... Also, non-fuzzy reasoners applied over such a fuzzy OWL ontology can discard the fuzzy part, i.e., the annotations, producing the same results as if they would not exist. A similar approach cannot be represented in OWL DL as it does not support rich enough annotation capabilities.

This work suggests a methodology for fuzzy ontology development. First, we can build the *core part* of the ontology by using any ontology editor supporting OWL 2, such as Protégé [8]. This allows to reason with this part using standard ontology reasoners. Then, we add the *fuzzy part* of the ontology by using annotation properties. This can also be done directly with an OWL 2 ontology editor, even if some sort of user assistance would be highly appreciated.

In future work we plan to develop a graphical interface (such as a Protégé plug-in) to assist users in the development of fuzzy ontologies, making the encoding of annotation properties transparent to the user. It will generate an OWL 2 file and translate it into a format supported by some popular fuzzy DL reasoners, such as FUZZYDL or DELOREAN.

Acknowledgment

F. Bobillo has been partially funded by Spanish Ministry of Science and Technology (project TIN2009-14538-C02-01).

References

[1] F. Bobillo, M. Delgado and J. Gómez-Romero. DeLorean: A reasoner for fuzzy OWL 1.1. In *Proceedings of the 4th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2008)*, vol. 423. of CEUR Workshop Proceedings, 2008.

[2] F. Bobillo, M. Delgado and J. Gómez-Romero. Crisp representations and reasoning for fuzzy ontologies. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 17(4):501–530, 2009.

[3] F. Bobillo, M. Delgado, J. Gómez-Romero and U. Straccia. Fuzzy description logics under Gödel semantics. *International Journal of Approximate Reasoning*, 50(3):494–514, 2009.

[4] F. Bobillo and U. Straccia. fuzzyDL: An expressive fuzzy description logic reasoner. In *Proceedings of the 17th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008)*, pp. 923–930. 2008.

[5] F. Bobillo and U. Straccia. Towards a crisp representation of fuzzy description logics under Łukasiewicz semantics. In *Proceedings of the 17th International Symposium on Methodologies for Intelligent Systems (ISMIS 2008)*, vol. 4994 of *Lecture Notes in Computer Science*, pp. 309–318. Springer-Verlag, 2008.

[6] F. Bobillo and U. Straccia. An OWL ontology for fuzzy OWL 2. In *Proceedings of the 18th International Symposium on Methodologies for Intelligent Systems (ISMIS 2009)*, vol. 5722 of *Lecture Notes in Computer Science*, pp. 151–160. Springer-Verlag, 2009.

[7] M. Gao and C. Liu. Extending OWL by fuzzy description logic. In *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2005)*, pp. 562–567. 2005.

[8] M. Horridge, D. Tsarkov and T. Redmond. Supporting early adoption of OWL 1.1 with Protege-OWL and FaCT++. In *Proceedings of the 2nd International Workshop on OWL: Experience and Directions (OWLED 2006)*, vol. 216. of CEUR Workshop Proceedings, 2006.

[9] P. Klinov and B. Parsia. Optimization and evaluation of reasoning in probabilistic description logic: Towards a systematic approach. In *Proceedings of the 7th International Semantic Web Conference (ISWC 2008)*, vol. 5318 of *Lecture Notes in Computer Science*, pp. 213–228. Springer-Verlag, 2008.

[10] T. Lukasiewicz and U. Straccia. Managing uncertainty and vagueness in description logics for the semantic web. *Journal of Web Semantics*, 6(4):291–308, 2008.

[11] B. Motik, P. F. Patel-Schneider and B. Parsia (editors). OWL 2 Web Ontology Language structural specification and functional-style syntax, 2009. `http://www.w3.org/TR/owl2-syntax/`.

[12] G. Stoilos, N. Simou, G. Stamou and S. Kollias. Uncertainty and the semantic web. *IEEE Intelligent Systems*, 21(5):84–87, 2006.

[13] G. Stoilos and G. Stamou. Extending fuzzy description logics for the semantic web. In *Proceedings of OWLED 2007*, vol. 258. of CEUR Workshop Proceedings, 2007.

[14] G. Stoilos, G. Stamou and J. Z. Pan. Fuzzy extensions of OWL: Logical properties and reduction to fuzzy description logics. *International Journal of Approximate Reasoning*. In press.

[15] U. Straccia. Reasoning within fuzzy description logics. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.

[16] U. Straccia. Transforming fuzzy description logics into classical description logics. In *Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA 2004)*, vol. 3229 of *Lecture Notes in Computer Science*, pp. 385–399. Springer-Verlag, 2004.

[17] U. Straccia. Description logics with fuzzy concrete domains. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI 2005)*. AUAI Press, 2005.

[18] U. Straccia. A fuzzy description logic for the semantic web. In E. Sanchez, editor, *Fuzzy Logic and the Semantic Web*, volume 1 of *Capturing Intelligence*, pp. 73–90. Elsevier Science, 2006.

[19] W3C Incubator Group on Uncertainty Reasoning for the World Wide Web Final Report: `http://www.w3.org/2005/Incubator/urw3/XGR-urw3`, 2008.

[20] M. Vacura, V. Svátek and P. Smrž. A pattern-based framework for representation of uncertainty in ontologies. In *Proc. of the 11th Int. Conf. on Text, Speech, and Dialogue (TSD 2008)*, vol. 5246 of *Lecture Notes in Computer Science*, pp. 227–234. Springer-Verlag, 2008.

[21] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.