# Fuzzy Ontologies and Fuzzy Integrals

Fernando Bobillo
*Dpt. of Computer Science and Systems Engineering*
*University of Zaragoza, Spain*
*fbobillo@unizar.es*

Umberto Straccia
*ISTI - CNR*
*Pisa, Italy*
*straccia@isti.cnr.it*

***Abstract*—Fuzzy ontologies extend classical ontologies to allow the representation of imprecise and vague knowledge. Although a relatively important amount of work has been carried out in the last years and they have been successfully used in several applications, several notions from fuzzy logic, such as fuzzy integrals, have not been considered yet in fuzzy ontologies.**

**In this work, we show how to support fuzzy integrals in fuzzy ontologies. As a theoretical formalism, we provide the syntax and semantics of a fuzzy Description Logic with fuzzy integrals. We also provide a reasoning algorithm for a family of fuzzy integrals and show how to encode them into the language Fuzzy OWL 2.**

***Keywords*-Fuzzy ontologies, Fuzzy description logics, Fuzzy OWL 2, Fuzzy integrals, Choquet integral, Sugeno integral**

## I. INTRODUCTION

In the last years, the use of *ontologies* as formalisms for knowledge representation in many different application domains has grown significantly. An ontology consists of a hierarchical description of important concepts in a particular domain, along with the description of the properties of their instances.

*Description Logics* (DLs) [1] play a key role in the design of *ontologies*. Notably, DLs are essential in the theoretical counterpart of the *Web Ontology Language OWL 2* [2], the standard language to represent ontologies.

Nowadays, it is widely agreed that "classical" ontology languages are not appropriate to deal with *imprecise* and *vague* knowledge, which is inherent to several real world domains [3]. Fuzzy set theory and fuzzy logic [4] have proved to be suitable formalisms to handle these types of knowledge. Therefore *fuzzy ontologies* emerge as useful in several applications, such as (multimedia) information retrieval, image interpretation, ontology mapping, matchmaking, the Semantic Web, and multi-criteria decision making [5]. So far, several fuzzy extensions of DLs can be found in the literature (see the survey in [6]).

*Aggregation Operators* (AOs) are mathematical functions that are used to combine information [7]. AOs have been widely used in computational intelligence because of their ability to fuse linguistically expressed pieces of information. The arithmetic mean, the weighted minimum and maximum, the weighted sum, the median and the Ordered Weighted Averaging (OWA) operators [8] are well-known examples.

*Fuzzy integral* is the extension of the concept of integral in the fuzzy case. Fuzzy integrals have been widely used in the setting of multi-criteria decision making because several AOs (including all the examples in the previous paragraph) can be expressed as particular cases of fuzzy integrals. Fuzzy integrals can also be used for the evaluation of type I quantified sentences. Typical examples of fuzzy integrals are the Sugeno [9], the quasi-Sugeno [10], and the Choquet [11] integrals.

The integration of fuzzy ontologies and fuzzy integrals is of great interest, in order to combine the advantages of both formalisms. This is the objective of this paper. More precisely, we provide syntax and semantics of a fuzzy DL extended with fuzzy integrals, provide a calculus for a family of fuzzy integrals (including the Choquet, the Sugeno and the quasi-Sugeno integrals), and show how to represent them. To this end, we will a use a representation of fuzzy ontologies using OWL 2 annotation properties that has been recently proposed (we will call it *Fuzzy OWL 2*)[1] [12]. The use of annotation properties makes it possible to use current OWL 2 editors for fuzzy ontology representation, such as Protégé. This approach has reached a sufficient level of maturity: there exists a Protégé plug-in to edit fuzzy ontologies, as well as some parsers that translate fuzzy ontologies represented using this approach into the languages supported by some fuzzy DL reasoners.

To the best of our knowledge, this is the first effort in the combination of fuzzy ontologies and fuzzy integrals. However, there is some previous research combining AOs and fuzzy DLs. [13] presents a fuzzy DL with general AOs, but the expressivity of the logic ($\mathcal{EL}$) is very limited. More recently, [14] presents a more general fuzzy DL ($\mathcal{ALC}$) but focused on a family of AOs (weighted sums, OWA operators, and quantifier-guided OWA operators). Our work extends it as fuzzy integrals are a more general case. A related work is [15], which considers fuzzy ontologies with general quantifiers, that could be used for some type of quantifier-guided aggregation.

In the following, we proceed as follows. Section II recalls basic notions on aggregation operators and fuzzy integrals. Section III provides the syntax and semantics of a fuzzy DL with fuzzy integrals. Section IV provides a reasoning algorithm. Next, Section V shows how to support them within Fuzzy OWL 2. Finally, Section VI sets out some conclusions and discusses future work.

## II. AGGREGATION OPERATORS AND FUZZY INTEGRALS

There is a large number of different AOs that impose different assumptions on the type of information to be aggregated. To what concerns us, an AO of dimension $n$ is a mapping $@ : \mathbb{R}^n \to \mathbb{R}$ that satisfies:

1) $@(a) = a$ (idempotent if unary)

---

[1] http://www.straccia.info/software/FuzzyOWL

978-1-4577-1675-1

2) $@(0,\ldots,0)=0$, $@(1,\ldots,1)=1$ (boundary conditions)
3) $@(a_1,\ldots,a_n)\leq @(b_1,\ldots,b_n)$ if $\forall i, a_i \leq b_i$ (monotone)

Often, an AO @ is parameterised with a vector of $n$ weights $W=[w_1,\ldots,w_n]$ such that $w_i \in [0,1]$ and $\sum_i w_i = 1$. In that case we will denote the AO as $@_W$.

Examples of AOs are the *arithmetic mean* (or average, denoted avg), the *weighted sum* (or weighted mean), the *median* (med), and the *Ordered Weighted Averaging* (OWA) operators [8], a class of AOs defined as

$$@_W^{\text{owa}}(a_1,\ldots,a_n)=\sum_j w_j b_j \ , \tag{1}$$

where $b_j$ is the $j$-th largest of the $a_i$.

A fundamental aspect of OWA operators is the reordering step. A weight $w_i$ is not associated with a specific argument but with an ordered position of the aggregate. By choosing different $W$ we can implement different AOs. The OWA operator is a non-linear operator as a result of the process of determining the $b_j$. Notable OWA operators are:

$\max(a_1,\ldots,a_n)$ for $W=[1,0,\ldots,0]$
$\min(a_1,\ldots,a_n)$ for $W=[0.\ldots,0,1]$
$\text{avg}(a_1,\ldots,a_n)$ for $W=[1/n,1/n,\ldots,1/n]$
$\text{med}(a_1,\ldots,a_n)$ for $w_i=0, n$ odd and $w_{(n+1)/2}=1$,
or n even and $w_{n/2}=0.5=w_{n/2+1}$

In the rest of this section, we will focus on fuzzy integrals. Firstly, we need a preliminary definition. Let $X=\{x_1,\ldots,x_n\}$. A *fuzzy measure* $\mu$ is a function $\mu: 2^X \rightarrow [0,1]$ verifying the following axioms:

- $\mu(\emptyset)=0$ (boundary condition).
- $A\subseteq B \Rightarrow \mu(A)\leq\mu(B)$ (monotonicity).

A fuzzy measure $\mu$ is normalized iff $\mu(X)=1$.

The *Sugeno integral* [9] of a function $f: X\rightarrow[0,1]$ with respect to $\mu$ is defined by

$$\mathcal{SI}_\mu(f)=\max_{i=1}^{n}\{\min\{f(x_{(i)}),\mu(A_{(i)})\}\} \ , \tag{2}$$

where $\cdot_{(i)}$ indicates that the indices have been permuted so that $0\leq f(x_{(1)})\leq\cdot\leq f(x_{(n)})\leq 1$ and $A_{(i)}=\{x_{(i)},x_{(i+1)},\ldots,x_{(n)}\}$. The Sugeno integral generalizes some AOs, such as weighted minimum/maximum, and median.

This definition can be generalized by considering any t-norm function $\otimes$. However, the only meaningful t-conorm is the maximum. This way, the *quasi-Sugeno integral* [10] of a function $f: X\rightarrow[0,1]$ with respect to $\mu$ is defined by

$$\mathcal{QSI}_\mu(f)=\max_{i=1}^{n}\{f(x_{(i)})\otimes\mu(A_{(i)})\} \ . \tag{3}$$

The *Choquet integral* [11] of a function $f: X\rightarrow[0,1]$ with respect to $\mu$ is defined by

$$\mathcal{CI}_\mu(f)=\sum_{i=1}^{n}(f(x_{(i)})-f(x_{(i-1)}))\mu(A_{(i)}) \tag{4}$$

where $f(x_{(0)})=0$. The Choquet integral generalizes weighted sum and OWA operators.

For multi-criteria decision making, the function $f$ measures the global satisfaction of an user according to several criteria $C_i$ (that will be represented by fuzzy concepts). Furthermore, we will parameterise the fuzzy integrals using a vector $W$, such that $w_i=\mu(A_{(i)})$. Every weight $w_i$ measures the degree of satisfaction when the criteria $C_{(i)},C_{(i+1)},\ldots,C_{(n)}$ are satisfied simultaneously. Consequently, we will use the denote the fuzzy integrals as $\int_W(C_1,\ldots,C_n)$.

## III. A FUZZY DL WITH FUZZY INTEGRALS

The aim of this section is to show how current fuzzy DLs can be extended to support fuzzy integrals. For illustrative purposes and for reasons of space, we will just consider a minimal fuzzy DL (see, e.g. [16] for a more expressive fuzzy DL). For the sake of our purpose, we will consider $\mathcal{ALCF}(\mathbf{D})$ [17], which is the basic DL $\mathcal{ALC}$ extended with functional roles (letter $\mathcal{F}$) and fuzzy concrete domains (letter $\mathbf{D}$).

A *fuzzy concrete domain* is a pair $\langle\Delta_{\mathbf{D}},\Phi_{\mathbf{D}}\rangle$, where $\Delta_{\mathbf{D}}$ is an interpretation domain and $\Phi_{\mathbf{D}}$ is the set of *fuzzy domain predicates* $\mathbf{d}$ with a predefined arity $n$ and an interpretation $\mathbf{d}^{\mathbf{D}}\colon\Delta_{\mathbf{D}}^n\rightarrow[0,1]$, which is a $n$-ary fuzzy relation over $\Delta_{\mathbf{D}}$ [17]. In our specific fuzzy DL, we assume that predicates are unary and $\Delta_{\mathbf{D}}$ are non-negative real numbers.

Now, consider pair wise disjoint alphabets of *concepts names* (denoted $A$), *abstract roles names* (denoted $R$) and *concrete roles names* (denoted $T$). Within the alphabet of abstract and concrete roles, we have distinguished subsets of *abstract functional roles names* (denoted $f$) and *concrete functional roles names* (denoted $t$), respectively. Functional roles are also called *features*.

*Concepts* (denoted $C$ or $D$) of the language can be built inductively from atomic concepts ($A$), top concept $\top$, bottom concept $\bot$, abstract roles ($R$), concrete roles ($T$) as follows. The syntax of fuzzy concepts is as follows [16]:

$$\begin{array}{rcl} C,D & \rightarrow & \top \mid \ \bot \ \mid A \mid C\sqcap D \mid C\sqcup D \mid \neg C \mid \\ & & \forall R.C \mid \exists R.C \mid \forall T.\mathbf{d} \mid \exists T.\mathbf{d} \\ \mathbf{d} & \rightarrow & ls(a,b) \mid rs(a,b) \mid tri(a,b,c) \mid trz(a,b,c,d) \end{array}$$

where $ls, rs, tri, trz$ stand for left-shoulder, right-shoulder, triangular and trapezoidal membership functions [17]. These are our concrete domain predicates. So, for instance the expression $Human\sqcap\exists hasChild.Lovely$ is a concept (unary predicate), which will denote the fuzzy set of humans that have a lovely child. Also, the expression $Human\sqcap\exists hasAge.ls(10,30)$ will denote the set of young humans (their age is $ls(10,30)$).

We next extend fuzzy DLs to support fuzzy integrals. Let $\int_W(C_1,\ldots,C_n)$ be the fuzzy integral of a function. Then, the following are concept expressions

$$C \quad\rightarrow\quad C \mid \int_W(C_1,\ldots,C_n) \ .$$

*Example 1:* The following concept expressions denote the fuzzy set of hotels that are cheap, close to the venue and comfortable, in which the degree is computed respectively as the Choquet integral and the Sugeno integral.

$Hotel\sqcap\mathcal{CI}_W(Cheap, CloseToVenue, Comfortable)$
$Hotel\sqcap\mathcal{SI}_W(Cheap, CloseToVenue, Comfortable)$

A *Fuzzy Knowledge Base* (or *fuzzy Ontology*) comprises a fuzzy ABox $\mathcal{A}$ and a fuzzy TBox $\mathcal{T}$. A *fuzzy ABox* consists of

| Concept | | Semantics | Concept | | Semantics |
|---|---|---|---|---|---|
| $(\top)^{\mathcal{I}}(x)$ | = | $1$ | $(\forall R.C)^{\mathcal{I}}(x)$ | = | $\inf_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \Rightarrow C^{\mathcal{I}}(y)\}$ |
| $(\bot)^{\mathcal{I}}(x)$ | = | $0$ | $(\exists R.C)^{\mathcal{I}}(x)$ | = | $\sup_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y)\}$ |
| $(A)^{\mathcal{I}}(x)$ | = | $A^{\mathcal{I}}(x)$ | $(\forall T.\mathbf{d})^{\mathcal{I}}(x)$ | = | $\inf_{v \in \Delta_{\mathbf{D}}} \{T^{\mathcal{I}}(x,v) \Rightarrow \mathbf{d}^{\mathbf{D}}(v)\}$ |
| $(C \sqcap D)^{\mathcal{I}}(x)$ | = | $C^{\mathcal{I}}(x) \otimes D^{\mathcal{I}}(x)$ | $(\exists T.\mathbf{d})^{\mathcal{I}}(x)$ | = | $\sup_{v \in \Delta_{\mathbf{D}}} \{T^{\mathcal{I}}(x,v) \otimes \mathbf{d}^{\mathbf{D}}(v)\}$ |
| $(C \sqcup D)^{\mathcal{I}}(x)$ | = | $C^{\mathcal{I}}(x) \oplus D^{\mathcal{I}}(x)$ | $(a{:}C)^{\mathcal{I}}$ | = | $C^{\mathcal{I}}(a^{\mathcal{I}})$ |
| $(\neg C)^{\mathcal{I}}(x)$ | = | $\ominus C^{\mathcal{I}}(x)$ | $((a,b){:}R)^{\mathcal{I}}$ | = | $R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}})$ |
| $(\int_W (C_1, \ldots, C_n))^{\mathcal{I}}(x)$ | = | $\int_W ({C_1}^{\mathcal{I}}(x), \ldots, {C_n}^{\mathcal{I}}(x))$ | $(C \sqsubseteq D)^{\mathcal{I}}$ | = | $\inf_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x)\}$ |

Figure 1. Semantics of fuzzy concepts and axioms.

a finite set of *fuzzy assertions* of one of the following types: a fuzzy *concept assertion* of the form $\langle a{:}C, r\rangle$ (with informal meaning, individual $a$ is an instance of concept $C$ with degree at least $r$), or a fuzzy *role assertion* of the form $\langle (a,b){:}R, r\rangle$ (the pair of individuals $(a,b)$ is an instance of role $R$ with degree at least $r$), with $r \in [0,1]$. In FOL, $\langle a{:}C, r\rangle$ may be seen as a fuzzy statement of the form $C(a) \geqslant r$, while $\langle (a,b){:}R, r\rangle$ may be seen as a fuzzy statement of the form $R(a,b) \geqslant r$.

A *fuzzy TBox* consists of a finite set of *fuzzy General Concept Inclusions (fuzzy GCIs)*, which are expressions of the form $\langle C \sqsubseteq D, r\rangle$ (with informal meaning, the degree of subsumption between concept $C$ and $D$ is not less than $r$). $C \equiv D$ is a shorthand for both $\langle C \sqsubseteq D, 1\rangle$ and $\langle D \sqsubseteq C, 1\rangle$.

From a semantics point of view, a *fuzzy interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ relative to the fuzzy concrete domain $\langle \Delta_{\mathbf{D}}, \Phi_{\mathbf{D}}\rangle$, consists of a nonempty set $\Delta^{\mathcal{I}}$ (the *domain*), disjoint from $\Delta_{\mathbf{D}}$, and of a *fuzzy interpretation function* $\cdot^{\mathcal{I}}$ that coincides with $\cdot_{\mathbf{D}}$ on every fuzzy concrete predicate, and it assigns:

- to each abstract concept $C$ a function $C^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \to [0,1]$;
- to each abstract role $R$ a function $R^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$;
- to each abstract feature $f$ a partial function $f^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to \{0,1\}$[2] such that for all $u \in \Delta^{\mathcal{I}}$ there is an unique $w \in \Delta^{\mathcal{I}}$ on which $f^{\mathcal{I}}(u,w)$ is defined;
- to each concrete role $T$ a function $T^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}} \to [0,1]$;
- to each concrete feature $t$ a partial function $t^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}} \to \{0,1\}$ such that for all $u \in \Delta^{\mathcal{I}}$ there is an unique $v \in \Delta_{\mathbf{D}}$ on which $t^{\mathcal{I}}(u,v)$ is defined.

Given arbitrary t-norm $\otimes$, t-conorm $\oplus$, negation function $\ominus$ and implication function $\Rightarrow$ [18], the fuzzy interpretation function is extended to *complex concepts* and *fuzzy axioms* as in Figure 1, where $\int_W$ is a fuzzy integral[3].

Now, we will define some common reasoning tasks [19]. Let $\alpha \in \{a{:}C, (a,b){:}R, C \sqsubseteq D\}$ and $\tau = \langle \alpha, r\rangle$. A fuzzy interpretation $\mathcal{I}$ *satisfies* (is a *model* of) a fuzzy axiom $\tau$ iff $\alpha^{\mathcal{I}} \geq r$. A fuzzy KB $\mathcal{K}$ is *satisfiable* iff there is a model $\mathcal{I}$ which satisfies each of the fuzzy axioms in $\mathcal{K}$. A fuzzy axiom $\tau$ is a *logical consequence* of a fuzzy KB $\mathcal{K}$ iff, for every model $\mathcal{I}$ of $\mathcal{K}$, $\mathcal{I} \models \tau$. The *Best Entailment Degree* (BED) of an axiom $\alpha$ w.r.t. a fuzzy KB $\mathcal{K}$ is defined as $bed(\mathcal{K}, \alpha) = \sup\{r \mid \mathcal{K} \models \langle \alpha, r\rangle\}$. The *Best Satisfiability Degree* (BSD) of a concept $C$ w.r.t. a fuzzy KB $\mathcal{K}$ as $bsd(\mathcal{K}, C) = \sup_{\mathcal{I} \models \mathcal{K}} \sup_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x)$ [16].

*Example 2:* Suppose we have some data about hotels, as shown in the following table:

| *hotel* | *price* | *distance* | *star* |
|---|---|---|---|
| $h_1$ | 150 | 300 | 4 |
| $h_2$ | 100 | 500 | 2 |

Assume that we are looking for a cheap, close to the venue and comfortable hotel. To this end, we may encode the problem using a fuzzy KB $\mathcal{K}$ with the following ABox $\mathcal{A}$:

$$\begin{array}{l} Hotel(h_1), Hotel(h_2), \\ price(h_1, 150), price(h_2, 100), \\ distance(h_1, 300), distance(h_2, 500), \\ star(h_1, 4), star(h_2, 2) \end{array}$$

where $price, distance$ and $star$ are concrete features.

The TBox $\mathcal{T}$ holds some definitions for cheap, close and comfortable hotels and contains the following axioms:

$$\begin{array}{rcl} Cheap & = & Hotel \sqcap \exists price.ls(60, 120) \\ CloseToVenue & = & Hotel \sqcap \exists distance.ls(200, 400) \\ Comfortable & = & Hotel \sqcap \exists star.rs(3, 5) \ . \end{array}$$

Note that, e.g., we have that $\mathcal{K} \models \langle h_2{:}Cheap, 0.33\rangle$, i.e., $h_2$ is a cheap hotel to degree no less than $0.33$. Now, or some fuzzy integral $\int_W$, we may compute the degree $n_i$ to which hotel $h_i$ satisfies our request by suitably aggregating the degree of cheapness, closeness and comfortableness of hotel $h_i$:

$$n_i = bed(\mathcal{K}, h_i{:}\textstyle\int_W (Cheap, CloseToVenue, Comfortable)) \ .$$

## IV. A REASONING ALGORITHM

We next provide a reasoning algorithm to solve the BED problem in fuzzy DLs with fuzzy integrals. Our algorithm is very similar to others in the literature, with the only difference being the management of fuzzy integrals in Section IV-A, but it is described here in detail to make the paper self-contained. We will only consider here the BED problem, but other reasoning tasks can be reduced to it [19].

To simplify our exposition, we will make the following assumptions. We restrict the fuzzy DL to the case of Lukasiewicz fuzzy logic (and, thus, support "Zadeh logic" as well). Other fuzzy logics could be dealt with similarly as in [17], [19]. Furthermore, we will restrict KBs to be *unfoldable*, which is defined below. The reason is that the combination of GCIs and fuzzy DLs is problematic, as recently shown in [20].

A fuzzy TBox $\mathcal{T}$ is *unfoldable* iff it verifies 3 constraints:

---

[2] In fuzzy DLs, abstract and concrete features are usually crisp [19].

[3] For ease of presentation, we identify the interpretation of $ls, rs, tri, trz$ and $\int_W$ with the functions themselves.

- every axiom in $\mathcal{T}$ is either of the form $\langle A \sqsubseteq C, 1\rangle$ or $A \equiv C$, where $A$ is an atomic concept[4];
- there is no concept $A$ such that it appears more than once on the left hand side of some axiom in $\mathcal{T}$; and
- no cyclic definitions are present in $\mathcal{T}$[5]

Analogously, a fuzzy KB is *unfoldable* when its TBox is unfoldable. Unfoldable TBoxes can be eliminated through an *expansion process* [20].Essentially, each axiom $A \sqsubseteq C \in \mathcal{T}$ can be replaced with $A = C \sqcap A^*$, where $A^*$ is a new concept name. Let $\mathcal{K}'$ the obtained knowledge base. $\mathcal{K}'$ can be *expanded* by substituting every concept name $A$ occurring in $\mathcal{K}$, which is defined in $\mathcal{T}$, with its defining term in $\mathcal{T}$.

The basic idea of the algorithm follows. Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$, with $\mathcal{T}$ unfoldable. In order to solve the BED problem, we combine appropriate DL tableaux rules with methods developed in the context of *many-valued logics* (MVLs) [21]. In order to determine e.g., $bed(\mathcal{K}, a{:}C)$, we consider an expression of the form $\langle a{:}\neg C, 1-x\rangle$ (informally, $\langle a{:}C \leq x\rangle$), where $x$ is a $[0,1]$-valued variable. Then we construct a tableaux for $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \cup \{\langle a{:}\neg C, 1-x\rangle\}\rangle$ in which the application of satisfiability preserving rules generates new fuzzy assertion axioms together with *inequations* over $[0,1]$-valued variables. These inequations have to hold in order to respect the semantics of the DL constructors. Hence, we *minimise* the original variable $x$ such that all constraints are satisfied[6].

In general, depending on the semantics of the DL constructors and fuzzy domain predicates we may end up with a general, bounded *Mixed Integer Non Linear Programming* (MINLP) optimization problem. In this paper, however, due to our restrictions above and those we will impose to AOs, we will end up with a *Mixed Integer Linear Program* (MILP) optimization problem [22]. Interestingly, as for the MVL case, the tableaux we are generating contains *one* branch only and, thus, just *one* MILP problem has to be solved.

We recall that a MILP problem consists in minimizing a linear function with respect to a set of constraints that are linear inequations in which rational and integer variables can occur. More precisely, let $x = \langle x_1, \ldots, x_k\rangle$ and $y = \langle y_1, \ldots, y_m\rangle$ be variables over $\mathbb{Q}$ and $\mathbb{Z}$ respectively, over the integers and let $A$, $B$ be integer matrices and $h$ an integer vector. The variables in $y$ are called *control variables*. Let $f(x,y)$ be an $k+m$-ary linear function. Then the general MILP problem is to find $\bar{x} \in \mathbb{Q}^k, \bar{y} \in \mathbb{Z}^m$ such that $f(\bar{x}, \bar{y}) = \min\{f(x,y) \,|\, Ax + By \geq h\}$. Furthermore, we say that $M \subseteq [0,1]^k$ is MILP-representable iff there is a MILP $(A, B, h)$ with $k$ real and $m$ $0-1$ variables such that $M = \{x : \exists y \in \{0,1\}^m$ such that $Ax + By \geq h\}$. We will require that a constructor $f$ is MILP-representable. In particular, the sets $g(f) = \{\langle x_1, \ldots, x_k, x\rangle \colon f(x_1, \ldots, x_k) \geq x\}$ and $\bar{g}(f) = \{\langle x_1, \ldots, x_k, x\rangle \colon f(x_1, \ldots, x_k) \leq x\}$ should be MILP-representable. For instance, in Łukasiewicz fuzzy logic, $x_1 \otimes x_2 \geq l$ can be written as $\{l \leq y,\, x_1 + x_2 - 1 \leq l,\, x_1 + x_2 - y \geq l,\, y \in \{0,1\}\}$. Therefore, $g(\otimes)$ is MILP-representable.

To guarantee that our reasoning process ends up to solve a MILP problem, we require that every constructor is MILP-representable. For instance, classical logic, Zadeh fuzzy logic, and Lukasiewicz fuzzy logic, are MILP-representable. It is not difficult to see that indeed the functions $ls, rs, tri$ and $trz$ are MILP-representable as well (see, e.g. [17], [19]). Furthermore, as we will see in Section IV-A, the Choquet, the Sugeno and the quasi-Sugeno integrals are also MILP-representable.

Like most of the tableaux algorithms, our algorithm works on *completion-forests*. A completion-forest $\mathcal{F}$ for $\mathcal{K}$ is a collection of trees whose distinguished roots are arbitrarily connected by edges. Each node $v$ is labelled with a set $\mathcal{L}(v)$ of expressions of the form $\langle C, l\rangle$, where $C \in sub(\mathcal{K})$ ($sub(\mathcal{K})$ is the set of named concepts appearing in $\mathcal{K}$), and $l$ is either a rational, a variable $x$, or an expression of the form $1-x$. The intuition is that $v$ is an instance of $C$ to degree equal or greater than the evaluation of $l$.

Each edge $\langle v, w\rangle$ is labelled with a set $\mathcal{L}(\langle v, w\rangle)$ of expressions of the form $\langle R, l\rangle$, where $R \in \mathcal{R}_{\mathcal{K}}$ ($\mathcal{R}_{\mathcal{K}}$ is the set of roles in $\mathcal{K}$). The intuition here is that $\langle v, w\rangle$ is an instance of $R$ to degree greater or equal than the evaluation of $l$.

The forest has associated a set $\mathcal{C}_{\mathcal{F}}$ of constraints of the form $l \leq l'$, $l = l'$, $x_i \in [0,1], y_i \in \{0,1\}$, where $l, l'$ are arithmetic expressions on the variables occurring in the forest.

The algorithm initializes a forest $\mathcal{F}$ to contain

- a root node $v_0^i$, for each individual $a_i$ occurring in $\mathcal{A}$, labelled with $\mathcal{L}(v_0^i)$ such that $\mathcal{L}(v_0^i)$ contains $\langle C_i, r\rangle$ for each fuzzy assertion $\langle a_i{:}C_i, r\rangle \in \mathcal{A}$, and
- an edge $\langle v_0^i, v_0^j\rangle$, for each fuzzy assertion $\langle (a_i, a_j){:}R_i, r\rangle \in \mathcal{A}$, labelled with $\mathcal{L}(\langle v_0^i, v_0^j\rangle)$ such that $\mathcal{L}(\langle v_0^i, v_0^j\rangle)$ contains $\langle R_i, r\rangle$.

$\mathcal{F}$ is then expanded by repeatedly applying the completion rules described below. The completion-forest is complete when none of the completion rules are applicable. Then, the MILP problem on the set of constraints $\mathcal{C}_{\mathcal{F}}$ is solved.

With $x_\alpha$ we denote the variable associated to the *assertion* $\alpha$ of the form $a{:}C$ or $(a,b){:}R$. $x_\alpha$ will take the truth value associated to $\alpha$. With $x_c$ we will denote the variable associated to the concrete individual $c$.

Let us discuss now how to deal with feature roles. Let $\mathcal{F}$ be a forest, $p$ an abstract or concrete feature such that we have two edges $\langle v, w_1\rangle$ and $\langle v, w_1\rangle$ such that $\langle p, l_1\rangle$ and $\langle p, l_2\rangle$ occur in $\mathcal{L}(\langle v, w_1\rangle)$ and $\mathcal{L}(\langle v, w_2\rangle)$, respectively (informally, $\mathcal{F}$ contains $\langle (v, w_1){:}p, l_1\rangle$ and $\langle (v, w_2){:}p, l_2\rangle$). Then we call such a pair a *fork*. As $p$ is a partial function, such a fork means that $w_1$ and $w_2$ have to be interpreted as the same individual. Such a fork can be deleted by adding both $\mathcal{L}(\langle v, w_2\rangle)$ to $\mathcal{L}(\langle v, w_1\rangle)$ and $\mathcal{L}(w_2)$ to $\mathcal{L}(w_1)$, and then deleting node $w_2$. At the beginning, we remove the forks from the initial forest. We assume that forks are eliminated as soon as they appear (as part of a rule application) with the proviso that newly generated nodes are replaced by older ones and not vice-versa.

---

[4] In particular this forbids concept $\top$ on the left hand side of axioms.

[5] *A directly uses* primitive concept $B$ in $\mathcal{T}$, if there is some axiom $\tau \in \mathcal{T}$ such that $A$ is on the left hand side of $\tau$ and $B$ occurs in the right hand side of $\tau$. Let *uses* be the transitive closure of the relation *directly uses* in $\mathcal{T}$. $\mathcal{T}$ is *cyclic* iff there is $A$ such that $A$ uses $A$ in $\mathcal{T}$.

[6] Informally, suppose the minimal value is $\bar{n}$. We will know then that for any interpretation $\mathcal{I}$ satisfying the knowledge base such that $(a{:}C)^{\mathcal{I}} < \bar{n}$, the starting set is unsatisfiable and, thus, $(a{:}C)^{\mathcal{I}} \geq \bar{n}$ has to hold. Which means that $bed(\mathcal{K}, a{:}C) = \bar{n}$

Now we are ready to present the inference rules. Recall that $\mathbf{d} \in \{ls, rs, tri, trz\}$, while $\int \in \{\mathcal{SI}_W, \mathcal{QSI}_W, \mathcal{CI}_W\}$:

- $(\bot)$. If $\langle \bot, l\rangle \in \mathcal{L}(v)$ then $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{l = 0\}$.
- $(R)$. If $\langle R, l\rangle \in \mathcal{L}(\langle v, w\rangle)$ then $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{(v,w):R} \geq l\} \cup \{x_{(v,w):R} \in [0,1]\}$. The case of concrete roles $T$ is similar.
- $(f)$. If $\langle f, l\rangle \in \mathcal{L}(\langle v, w\rangle)$ then $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{(v,w):f} \geq l\} \cup \{x_{(v,w):f} \in \{0,1\}\}$. The case of concrete features $t$ is similar.
- $(C)$. If $\langle C, l\rangle \in \mathcal{L}(v)$ then $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{v:C} \geq l\} \cup \{x_{v:C} \in [0,1]\}$.
- $(\sqcap)$. If $\langle C \sqcap D, l\rangle \in \mathcal{L}(v)$ then *(i)* append $\langle C, x_{v:C}\rangle$, $\langle D, x_{v:D}\rangle$ to $\mathcal{L}(v)$, and *(ii)* $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{v:C} \otimes x_{v:D} \geq l\}$.
- $(\sqcup)$. If $\langle C \sqcup D, l\rangle \in \mathcal{L}(v)$ then *(i)* append $\langle C, x_{v:C}\rangle$, $\langle D, x_{v:D}\rangle$ to $\mathcal{L}(v)$, and *(ii)* $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{v:C} \oplus x_{v:D} \geq l\}$.
- $(\neg)$. If $\langle \neg C, l\rangle \in \mathcal{L}(v)$ then *(i)* append $\langle C, x_{v:C}\rangle$ to $\mathcal{L}(v)$, and *(ii)* $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{v:\neg C} = 1 - x_{v:C}\}$.
- $(\forall)$. If *(i)* $\langle \forall R.C, l_1\rangle \in \mathcal{L}(v)$, $\langle R, l_2\rangle \in \mathcal{L}(\langle v, w\rangle$, and *(ii)* the rule has not been already applied to this pair then *(i)* append $\langle C, x_{w:C}\rangle$ to $\mathcal{L}(w)$, and *(ii)* $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{w:C} \geq l_1 \otimes l_2\}$.
- $(\exists)$. If $\langle \exists R.C, l\rangle \in \mathcal{L}(v)$ then *(i)* create a new node $w$, and *(ii)* append $\langle R, x_{(v,w):R}\rangle$ to $\mathcal{L}(\langle v, w\rangle)$, and *(iii)* append $\langle C, x_{w:C}\rangle$ to $\mathcal{L}(w)$, and *(iv)* $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{w:C} \otimes x_{(v,w):R} \geq l\}$.
- $(\forall_\mathbf{D})$. If $\langle \forall T.\mathbf{d}, l\rangle \in \mathcal{L}(v)$, the case is similar as in rule $(\forall)$.
- $(\exists_\mathbf{D})$. If $\langle \exists T.\mathbf{d}, l\rangle \in \mathcal{L}(v)$, the case is similar as in rule $(\exists)$.
- $(\mathbf{d})$. If $\langle d, l\rangle \in \mathcal{L}(v)$ then $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{\gamma(v{:}\mathbf{d}, l)\}$, where the set $\gamma(v{:}\mathbf{d}, l)$ is obtained from the MILP representation of $g(\mathbf{d})$ by replacing all occurrences of $x$ with $l$ and $x_1$ with $x_v$.
- $(\bar{\mathbf{d}})$. The case $\langle \neg\mathbf{d}, l\rangle \in \mathcal{L}(v)$ is similar as in rule $(\mathbf{d})$, where we use the MILP representation of $\bar{g}(\mathbf{d})$ in place of $g(\mathbf{d})$.
- $(\int)$. If $\langle \int_W(C_1, \ldots, C_n), l\rangle \in \mathcal{L}(v)$ then *(i)* append $\langle C_i, x_{v:C_i}\rangle$ to $\mathcal{L}(v)$, and *(ii)* $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{\gamma(x_{v:C_1}, \ldots, x_{v:C_n}, l)\}$, where the set $\gamma(x_{v:C_1}, \ldots, x_{v:C_n}, l)$ is obtained from the MILP representation of $g(\int)$ by replacing all occurrences of $x$ with $l$ and $x_i$ with $x_{v:C_i}$.

Now, the following proposition can be shown.

*Proposition 1 (Termination, Soundness, Completeness):* Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$, with $\mathcal{T}$ unfoldable, (i) the tableau algorithm terminates; (ii) if the expansion rules can be applied to $\mathcal{K}$ such that they yield a complete completion-forest $\mathcal{F}$ such that $\mathcal{C}_\mathcal{F}$ has a solution, then $\mathcal{K}$ has a model; and (iii) if $\mathcal{K}$ has a model, then the expansion rules can be applied in such a way that the tableaux algorithm yields a complete completion-forest for $\mathcal{K}$ such that $\mathcal{C}_\mathcal{F}$ has a solution. □

### A. MILP Encoding of Fuzzy Integrals

We succinctly show that the Choquet and the quasi-Sugeno integrals are MILP-representable. More precisely, we show that the set $g(\int_W) = \{\langle x_1, \ldots, x_k, x\rangle \colon \int_W(x_1, \ldots, x_k) \geq x\}$ is MILP-representable. These representations can be seen as an extension of the representations of OWA operators in [14].

Let us start with the Choquet integral. Let $N = \{1, \ldots, n\}$. We create some new $[0,1]$-valued variables $y_i$ $(i \in N)$ and add

$$y_1 \geq y_2 \geq \ldots \geq y_n \ . \tag{5}$$

The intuition is that $y_i$ will take the value of the $i$-th largest of the $x_j$. Next, we establish which one among $x_1, \ldots, x_n$ are the $y_i$. For $i, j \in N$, $k = 2$, and new binary variables $z_{ij}$, consider the following equations:

$$y_i \leq x_j + kz_{ij} \qquad x_j \leq y_i + kz_{ij}$$

$$\textstyle\sum_j z_{ij} = n - 1 \qquad \sum_i z_{ij} = n - 1 \qquad z_{ij} \in \{0,1\} \ . \tag{6}$$

The remaining of the encoding concerns to the definition of the Choquet integral:

$$y_1 \cdot w_1 + \textstyle\sum_{i=2}^{n} (y_i - y_{i-1}) w_i \geq x \ . \tag{7}$$

It can be verified that (i) if $z_{ij} = 0$ then $y_i = x_j$ is imposed; (ii) for any $y_i$, there is only one $x_j$ imposed to be equal to $y_i$; and (iii) for any $x_j$, there is only one $y_i$ imposed to be equal to $x_j$. That is, there is a bijection among the $y_j$ and the $x_i$, which together with Equation (7) guarantees that the Equations (5)–(7) correctly encode $\mathcal{CI}_W(x_1, \ldots, x_k) \geq x$.

Now we will consider the case of the quasi-Sugeno integral, which extends the case of the Sugeno integral. To begin with, the constraints expressed in Equations (5)–(6) are needed.

Next, we introduce some new $[0,1]$-valued variables $c_i (i \in N)$ that will take the value of the expression $y_i \otimes w_i$ (in the case of the Sugeno integral, the t-norm will be fixed to the minimum). This is possible because the t-norm is required to be MILP-representable. Hence, we add the constraints

$$c_i = y_i \otimes w_i. \tag{8}$$

Finally, we have to ensure that the maximum of the $c_i$ is greater or equal than $x$. To this end, we introduce some new binary variables $b_i (i \in N)$, and add the following constraints:

$$c_i + b_i \geq x \qquad \textstyle\sum_i b_i = n - 1 \qquad b_i \in \{0,1\} \ . \tag{9}$$

Again, it can easily be seen that Equations (5), (6), (8) and (9) correctly encode $\mathcal{QSI}_W(x_1, \ldots, x_k) \geq x$.

## V. REPRESENTATION IN FUZZY OWL 2

In this section, we will discuss the representation of fuzzy ontologies with some kinds of fuzzy integrals (namely the Choquet, the Sugeno and the quasi-Sugeno integrals) using a fuzzy extension of OWL 2.

The key idea of this representation is to use an OWL 2 ontology and extend their elements with annotations representing the features of the fuzzy ontology that OWL 2 cannot directly encode. In order to separate the annotations including fuzzy information from other annotations, we use a new annotation property called `fuzzyLabel`, and every annotation is identified by the tag `fuzzyOwl2`, with an attribute `fuzzyType` that specifies the type of element that is being annotated. For full details, we refer the reader to [12].

Now we are ready to show how to represent the fuzzy integrals. The representation is similar to the representation of OWA aggregation operators that can found in [14].

For each of these concepts, we create a new concept and add an annotation property to it, describing the type of the constructor and the value of their parameters. Hence, the domain of the annotation is an OWL 2 concept declaration, and the value of the attribute `fuzzyType` is `concept`.

Then, we use a tag `Concept`, with an attribute `type` describing the type of the fuzzy integral (`choquet`, `sugeno` or `quasisugeno`), and we represent the concepts and the weights associated to it. Due to the reordering step, it is not necessary to associate every weight to a particular concept, but we need two independent lists.

The formal syntax is the following:

```
<fuzzyOwl2 fuzzyType="concept">
  <Concept type=<INTEGRAL> >
    <Weights>
        ( <Weight><DOUBLE></Weight> )+
    </Weights>
    <Names>
        ( <Name><STRING></Name> )+
    </Names>
  </Concept>
</fuzzyOwl2>

<INTEGRAL> := "choquet" | "sugeno" | "quasisugeno"
```

*Example 3:* Assume that we want to represent the concept $\mathcal{CI}_W(Cheap, CloseToVenue, Comfortable)$, where $W = (0.1, 0.3, 0.6)$. So, we create a new atomic concept $A$ and annotate it as follows: (the example uses OWL/XML syntax [23]):

```
<AnnotationAssertion>
  <AnnotationProperty IRI=’#fuzzyLabel’/>
  <IRI>#A</IRI>
  <Literal datatypeIRI=’&rdf;PlainLiteral’>
    <fuzzyOwl2 fuzzyType="concept">
      <Concept type="choquet">
        <Weights>
            <Weight>0.1</Weight>
            <Weight>0.3</Weight>
            <Weight>0.6</Weight>
        </Weights>
        <Names>
            <Name>Cheap</Name>
            <Name>CloseToVenue</Name>
            <Name>Comfortable</Name>
        </Names>
      </Concept>
    </fuzzyOwl2>
  </Literal>
</AnnotationAssertion>
```

## VI. Conclusion

In this work we have shown how to include fuzzy integrals within fuzzy ontologies. In particular, we have defined the syntax and semantics for fuzzy DLs supporting fuzzy integrals. Then, we have provided a reasoning algorithm for a class of fuzzy integrals that are MILP-representable, including the Choquet, the Sugeno and the quasi-Sugeno integrals. We have also shown how to support these fuzzy integrals in Fuzzy OWL 2, a fuzzy version of the standard ontology language OWL 2.

In the future, we will implement our approach in the Fuzzy OWL 2 plug-in and in the fuzzyDL reasoner [16].

## References

[1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, Eds., The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, 2003.

[2] B. Cuenca-Grau, I. Horrocks, B. Motik, B. Parsia, P. F. Patel-Schneider, and U. Sattler, “OWL 2: The next step for OWL”, Journal of Web Semantics, vol. 6:4, pp. 309–322, 2008.

[3] E. Sanchez, Ed., Fuzzy Logic and the Semantic Web. Elsevier, 2006.

[4] L. A. Zadeh. “Fuzzy sets”, Information and Control, vol. 8, pp. 338–353, 1965.

[5] U. Straccia, “Multi-criteria decision making in fuzzy Description Logics: a first step”, in Proceedings of the 13th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2009), ser. LNAI, no. 5711. Springer, 2009, pp. 79–87.

[6] T. Lukasiewicz and U. Straccia, “Managing uncertainty and vagueness in Description Logics for the semantic web”, Journal of Web Semantics, vol. 6, pp. 291–308, 2008.

[7] V. Torra, Y. Narukawa, Modeling Decisions - Information Fusion and Aggregation Operators. Springer Verlag, 2007.

[8] R. R. Yager, “On ordered weighted averaging aggregation operators in multicriteria decision making”, IEEE Transactions on Systems, Man and Cybernetics, vol. 18, pp. 183–190, 1988.

[9] M. Sugeno. “Theory of fuzzy integrals and its applications”, Ph.D. dissertation, Tokyo Institute of Technology, Japan, 1974.

[10] S. Weber. “$\perp$–decomposable measures and integral for Archimedean t-conorms”, Journal of Mathematical Analysis and Applications, vol. 101, pp. 114–138, 1984.

[11] G. Choquet. “Theory of capacities”, Annales de l’Institut Fourier, vol. 5, pp. 131–295, 1953.

[12] F. Bobillo and U. Straccia. “Fuzzy ontology representation using OWL 2”, International Journal of Approximate Reasoning, doi:10.1016/j.ijar.2011.05.003.

[13] P. Vojtáš, “A fuzzy $\mathcal{EL}$ Description Logic with crisp roles and fuzzy aggregation for web consulting”, in Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2006), 2006, pp. 1834–1841.

[14] F. Bobillo and U. Straccia. “Aggregation operators and fuzzy OWL 2”, in Proceedings of the 20th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), 2011, pp. 1727–1734.

[15] D. Sánchez and A. G. B. Tettamanzi. “Fuzzy quantification in fuzzy Description Logics”, in Fuzzy Logic and the Semantic Web, E. Sanchez, Ed. Elsevier, 2006, pp. 135–159.

[16] F. Bobillo and U. Straccia, “fuzzyDL: an expressive fuzzy Description Logic reasoner”, in Proceedings of the 18th International Conference on Fuzzy Systems (FUZZ-IEEE 2008). IEEE Computer Society, 2008, pp. 923–930.

[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, pp. 559–567.

[18] P. Hájek, Metamathematics of fuzzy logic. Kluwer, 1998.

[19] F. Bobillo and U. Straccia, “Fuzzy Description Logics with general t-norms and datatypes”, Fuzzy Sets Systems, vol. 160(23), pp. 3382—3402, 2009.

[20] F. Bobillo, F. Bou, and U. Straccia, “On the failure of the finite model property in some fuzzy Description Logics”, Fuzzy Sets and Systems, vol. 172(1), pp. 1-12, 2011.

[21] R. Hähnle, “Advanced many-valued logics”, in Handbook of Philosophical Logic, 2nd Edition, D. M. Gabbay and F. Guenthner, Eds. Kluwer, 2001.

[22] H. Salkin and M. Kamlesh, Foundations of Integer Programming. North-Holland, 1988.

[23] B. Motik, B. Parsia, and P. F. Patel-Schneider, Eds., OWL 2 Web Ontology Language XML serialization, http://www.w3.org/TR/owl2-xml-serialization, 2009.