

# fuzzyDL: An Expressive Fuzzy Description Logic Reasoner

Fernando Bobillo and Umberto Straccia

**Abstract**—In this paper we present *fuzzyDL*, an expressive fuzzy Description Logic reasoner. We present its salient features, including some novel concept constructs and queries, and examples of use cases: matchmaking and fuzzy control.

## I. INTRODUCTION

*Description Logics* (DLs) [2] are a logical reconstruction of the so-called frame-based knowledge representation languages, with the aim of providing a simple well-established Tarski-style declarative semantics to capture the meaning of the most popular features of structured representation of knowledge. Nowadays, DLs have gained even more popularity due to their application in the context of the *Semantic Web* [3]. Indeed, the current standard language for specifying ontologies is the Web Ontology Language (OWL) [19], which comprises three sublanguages of increasing expressive power: OWL Lite, OWL DL and OWL Full. The logical counterpart of OWL Lite and OWL DL are the DLs *SHIF(D)* and *SHOIN(D)*, respectively [13].

Fuzzy DLs have been proposed as an extension to classical DLs with the aim to deal with *fuzzy/vague/imprecise concepts*. Since the first work of J. Yen in 1991 [29], an important number of works can be found in the literature. For a good survey on *fuzzy DLs* we refer the reader to [16].

This paper focuses on presenting the *fuzzyDL* system, an expressive fuzzy Description Logic reasoner. We present its salient features and some of its applications.

In the remainder, we proceed as follows. Section II reviews some preliminaries on fuzzy set theory and fuzzy logic. Section III presents the fuzzy DL which is currently supported by *fuzzyDL*. Next, Section IV describes its more relevant features, while Section V presents two use cases (matchmaking and fuzzy control). Finally, Section VI overviews some related work and presents some conclusions as well as some ideas for future work.

## II. FUZZY SET THEORY AND FUZZY LOGIC

Fuzzy set theory and fuzzy logic were proposed by Zadeh [30] to manage imprecise and vague knowledge. While in classical set theory elements either belong to a set or not, in fuzzy set theory elements can belong to some degree. More formally, let  $X$  be a set of elements called the reference set. A fuzzy subset  $A$  of  $X$ , is defined by a membership function  $\mu_A(x)$ , or simply  $A(x)$ , which assigns any  $x \in X$  to a value in the real interval between 0 and 1.

Fernando Bobillo is with the Department of Computer Science and Artificial Intelligence, University of Granada, Spain (email: fbo-billo@decsai.ugr.es).

Umberto Straccia is with the Istituto di Scienza e Tecnologie dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa, Italy (email: straccia@isti.cnr.it).

As in the classical case, 0 means no-membership and 1 full membership, but now a value between 0 and 1 represents the extent to which  $x$  can be considered as an element of  $X$ .

All crisp set operations are extended to fuzzy sets. The intersection, union, complement and implication set operations are performed by a t-norm function  $\otimes$ , a t-conorm function  $\oplus$ , a negation function  $\ominus$  and an implication function  $\Rightarrow$ , respectively. For a definition of these functions we refer the reader to [14]. Here, we just summarize the fuzzy operators which will be treated throughout this paper:

Łukasiewicz negation	$\ominus_{\mathbf{L}} \alpha =$	$1 - \alpha$
Gödel t-norm	$\alpha \otimes_G \beta =$	$\min\{\alpha, \beta\}$
Łukasiewicz t-norm	$\alpha \otimes_{\mathbf{L}} \beta =$	$\max\{\alpha + \beta - 1, 0\}$
Gödel t-conorm	$\alpha \oplus_G \beta =$	$\max\{\alpha, \beta\}$
Łukasiewicz t-conorm	$\alpha \oplus_{\mathbf{L}} \beta =$	$\min\{\alpha + \beta, 1\}$
Gödel implication	$\alpha \Rightarrow_G \beta =$	$\begin{cases} 1, & \text{if } \alpha \leq \beta \\ \beta, & \text{if } \alpha > \beta \end{cases}$
Łukasiewicz implication	$\alpha \Rightarrow_{\mathbf{L}} \beta =$	$\min\{1, 1 - \alpha + \beta\}$
Kleene-Dienes implication	$\alpha \Rightarrow_Z \beta =$	$\max\{1 - \alpha, \beta\}$

where  $\alpha, \beta \in [0, 1]$ . Fuzzy implications can be used to define families of fuzzy operators. We recall here the fuzzy families that we will consider in this paper:

- Zadeh logic uses Łukasiewicz negation, Gödel t-norm, Gödel t-conorm and Kleene-Dienes (KD) implication.
- Łukasiewicz logic uses Łukasiewicz negation, Łukasiewicz t-norm, Łukasiewicz t-conorm and Łukasiewicz implication.

One of the most important features of fuzzy logic is its ability to perform approximate reasoning, which involves inference rules with premises, consequences or both of them containing fuzzy propositions. A very popular formalism, due to their practical success, are fuzzy rule-based systems.

A fuzzy IF-THEN system consists of a rule base (a set of IF-THEN rules) and a reasoning algorithm performing an inference mechanism such as Rete [7]. In general, the input of the system is the current value for the input variable and the output is a fuzzy set, which can be defuzzified into a single value. In a fuzzy IF-THEN rule, its antecedents, consequences or both are fuzzy. Fuzzy IF-THEN rules are fired to a degree which is a function of the degree of match between their antecedent and the input. The deduction rule is Generalized Modus Ponens. Roughly speaking, given a rule “IF  $A$  THEN  $B$ ”, where  $A$  and  $B$  are fuzzy propositions, it is possible from a premise “ $A'$ ” which matches  $A$  to some degree, to deduce “ $B'$ ”, which is similar to  $B$ .

One of the most popular IF-THEN systems is the Mamdani model [18]. In a Mamdani model, fuzzy rules have the form

$$\text{IF } X_1 \text{ IS } A_1 \text{ AND } \dots \text{ AND } X_n \text{ IS } A_n \text{ THEN } Y \text{ IS } B, \quad (1)$$

where  $A_i$  and  $B$  are linguistic values defined by fuzzy sets on universes of discourse  $X_i$  and  $Y$  respectively,  $i = 1, \dots, n$ .

For every clause in the antecedent of the rule, the matching degree between the current value of the variable and the linguistic label in the rule is computed (typically, using the minimum or another t-conorm). If there exist several clauses, they are aggregated into a firing degree, using a fuzzy logic operator (typically, the maximum). Then, this firing degree is used for modifying the consequent of the rule using some function (typically the minimum).

Rules are fired using some inference algorithm. The computed consequences related to the same variable are aggregated (typically, using the maximum). Then, the output variables can be defuzzified. Some examples of defuzzification methods are the LOM (largest of maxima), SOM (smallest of maxima) and MOM (middle of maxima). More precisely, let  $B$  denote the fuzzy set to be defuzzified and let  $x$  be an arbitrary element of the universe. Then for all  $x$ :

- $x_{LOM}$  is the LOM iff  $\mu_B(x_{LOM}) \geq \mu_B(x)$  and, if  $\mu_B(x_{LOM}) = \mu_B(x)$  then  $x_{LOM} > x$ .
- $x_{SOM}$  is the SOM iff  $\mu_B(x_{SOM}) \geq \mu_B(x)$  and, if  $\mu_B(x_{SOM}) = \mu_B(x)$  then  $x_{SOM} < x$ .
- $x_{MOM}$  is the MOM iff  $x_{MOM} = (x_{LOM} + x_{SOM})/2$ .

### III. FUZZY DESCRIPTION LOGICS

We next present the fuzzy DL which is currently behind the *fuzzyDL* reasoner, an extension of [23], [24], [25], [26]. However, instead of dealing with the infinite set of truth degrees in  $[0, 1]$  as usual, *fuzzyDL* restricts to a discretized set  $[0, 1]_D = \{0, \frac{1}{n}, \dots, \frac{n-1}{n}, 1\}$  for an appropriate big natural  $n$  such that there is a machine representable  $0 < \epsilon < \frac{1}{n}$ . Assuming a finite set of truth degrees is unavoidable since *fuzzyDL* relies on numerical calculus (Mixed Integer Linear Programming or MILP [21]) using fixed precision, due to the physical limitations of computers<sup>1</sup>.

#### A. Syntax

The main ingredients of DLs are concepts, which denote unary predicates, and roles, which denote binary predicates. Then there are connectives which allow to construct complex concepts. For instance, if we use the concept *Human* to denote the set of humans, and the concept *Male* to denote the set of male objects, the complex concept (conjunction)  $Human \sqcap Male$  will denote the set of male humans. On the other hand, if *hasChild* denotes a role then the concept  $Human \sqcap \exists hasChild.Male$  will denote the the set of humans having a male child.

*fuzzyDL* supports concrete data types such as reals, integers, strings and allows the definition of concepts with explicit representation of fuzzy membership functions. This is implemented by relying on so-called fuzzy data type theory. A *fuzzy data type theory*  $\mathbf{D} = \langle \Delta_{\mathbf{D}}, \cdot_{\mathbf{D}} \rangle$  is such that  $\cdot_{\mathbf{D}}$  assigns to every  $n$ -ary data type predicate  $d$  an  $n$ -ary fuzzy relation over  $\Delta_{\mathbf{D}}$  [17]. For instance, the predicate  $\leq_{18}$  may be a

<sup>1</sup>It is well known that, e.g., we may loose associativity for very small numbers, however these cases are very rare in practice.

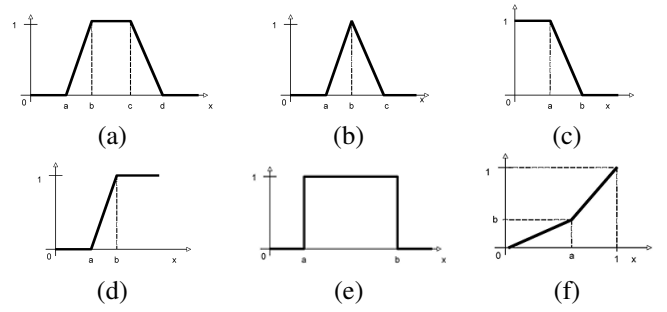


Fig. 1. (a) Trapezoidal function; (b) Triangular function; (c)  $L$ -function; (d)  $R$ -function; (e) Crisp interval; (f) Linear function.

unary crisp predicate over the natural numbers denoting the set of integers smaller or equal to 18. Concerning non-crisp fuzzy domain predicates, we recall that in fuzzy set theory and practice, there are many functions for specifying fuzzy set membership degrees. However, the trapezoidal (Fig. 1 (a)), the triangular (Fig. 1 (b)), the  $L$ -function (left-shoulder function, Fig. 1 (c)), and the  $R$ -function (right-shoulder function, Fig. 1 (d)) are simple, but most frequently used to specify membership degrees. For backwards compatibility, *fuzzyDL* also allows crisp intervals (Fig. 1 (e)). These functions are defined over the set of non-negative rationals  $\mathbb{Q}^+ \cup \{0\}$ . For instance, we may define *Young*:  $\mathbb{N} \rightarrow [0, 1]_D$  to be a fuzzy concrete predicate over the natural numbers denoting the degree of youngness of a person's age, as  $Young(x) = L(10, 30)$ .

*fuzzyDL* allows fuzzy modifiers, such as *very*, *more\_or\_less* and *slightly*, which apply to fuzzy sets to change their membership function. Formally, a *modifier* is a function  $f_m: [0, 1]_D \rightarrow [0, 1]_D$ . We will allow modifiers defined in terms of linear hedges (Figure 1 (f)) and triangular functions (Figure 1 (d)). For instance,  $very(x) = linear(0.8)$ . Modifiers have also been considered in [28], [12].

*fuzzyDL* is based on the fuzzy DL  $\mathcal{SHIF}(D)$ , i.e.,  $\mathcal{SHIF}$  with concrete data types, which we define next. Let  $\mathbf{A}$ ,  $\mathbf{R}_a$ ,  $\mathbf{R}_c$ ,  $\mathbf{I}$ ,  $\mathbf{I}_c$  and  $\mathbf{M}$  be non-empty finite and pair-wise disjoint sets of *concepts names* (denoted  $A$ ), *abstract roles names* (denoted  $R$ ), *concrete roles names* (denoted  $T$ ), *abstract individual names* (denoted  $x, y$ ), *concrete individual names* (denoted  $v$ ) and *modifiers* (denoted  $m$ ). Concepts may be seen as unary predicates, while roles may be seen as binary predicates.  $\mathbf{R}_a$  also contains a non-empty subset  $\mathbf{F}_a$  of *abstract feature names* (denoted  $r$ ), while  $\mathbf{R}_c$  contains a non-empty subset  $\mathbf{F}_c$  of *concrete feature names* (denoted  $t$ ). Features are functional roles. The syntax of fuzzy  $\mathcal{SHIF}$  concepts (denoted  $C, D$ ) is as follows:

$$C, D := \top \mid \perp \mid A \mid C \sqcap D \mid C \sqcup D \mid \neg C \mid \forall R.C \mid \exists R.C$$

Now, Fuzzy  $\mathcal{SHIF}(\mathbf{D})$  extends  $\mathcal{SHIF}$  with concrete data types, i.e., it has the additional concept constructs:

$$\begin{aligned} C, D &:= \forall T.d \mid \exists T.d \mid DR \\ d &:= crisp(a, b) \mid L(a, b) \mid R(a, b) \mid \\ &\quad triangular(a, b, c) \mid trapezoidal(a, b, c, d) \\ DR &:= \geq t \text{ val} \mid \leq t \text{ val} \mid = t \text{ val} \end{aligned}$$

where  $val$  is an integer, a real or a string depending on the range of the concrete feature  $t$ . For instance, the expression  $Human \sqcap (\leq hasAge\ 18)$  will denote the set of humans, which have an age less or equal than 18, while  $Human \sqcap \exists hasAge.L(10, 30)$  will denote the set of young humans (their age is  $L(10, 30)$ ).

Finally, *fuzzyDL* extends  $\mathcal{SHIF}(\mathbf{D})$  as follows:

$$\begin{aligned} C, D &:= C \sqcap_G D \mid C \sqcap_{\perp} D \mid C \sqcup_G D \mid C \sqcup_{\perp} D \mid \\ &C \rightarrow D \mid C \rightarrow_G D \mid C \rightarrow_{\perp} D \mid m(C) \mid \\ &n C \mid w_1 C_1 + \dots + w_k C_k \mid C \geq n \mid C \leq n \\ m &:= linear(a) \mid triangular(a, b, c) \end{aligned}$$

where  $n \in [0, 1]_D$ ,  $w_i \in [0, 1]_D$ ,  $\sum_{i=1}^k w_i = 1$ . For instance, the concept  $m(C)$  applies the modifier  $m$  to the concept  $C$  and, thus, e.g.,  $Human \sqcap \exists hasAge.linear(0.8)(L(10, 30))$  denotes the set of *very young* humans.

A *fuzzyDL fuzzy knowledge base* (KB)  $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$  consists of a fuzzy ABox  $\mathcal{A}$ , a fuzzy TBox  $\mathcal{T}$  and a fuzzy RBox  $\mathcal{R}$ .

A *fuzzy ABox*  $\mathcal{A}$  consists of a finite set of *fuzzy concept* and *fuzzy role assertion axioms* of the form  $\langle x:C, \alpha \rangle$  and  $\langle (x, y):R, \alpha \rangle$ , where  $\alpha \in (0, 1]_D$ . Informally, from a semantical point of view,  $\langle \tau, \alpha \rangle$  constrains the membership degree of  $\tau$  to be at least  $\alpha$ . Hence,  $\langle jim:YoungPerson, 0.2 \rangle$  says that *jim* is a *YoungPerson* with degree at least 0.2, while  $\langle (jim, tom):hasFriend, 1 \rangle$ , states that *jim* and *tom* are friends. If the  $\alpha$  is omitted, 1 is assumed.

A *fuzzy TBox*  $\mathcal{T}$  is a finite set of *fuzzy General Concept Inclusion axioms* (GCIs)  $\langle C \sqsubseteq D, \alpha \rangle$ , where  $\alpha \in (0, 1]_D$  and  $C, D$  are concepts. Informally,  $\langle C \sqsubseteq D, \alpha \rangle$  states that all instances of concept  $C$  are instances of concept  $D$  to degree  $\alpha$ , that is, the subsumption degree between  $C$  and  $D$  is at least  $\alpha$ . For instance,  $\langle Elephant \sqsubseteq Animal, 1 \rangle$  states that the class of elephants is a subclass of the class of animals. We write  $C = D$  as a shorthand of the two axioms  $\langle C \sqsubseteq D, 1 \rangle$  and  $\langle D \sqsubseteq C, 1 \rangle$ . For instance,  $Minor = Person \sqcap (\leq hasAge18)$  defines a person, whose age is less or equal to 18 (*hasAge* is a concrete feature), i.e., a minor. If the truth value  $\alpha$  is omitted then the value 1 is assumed.

*fuzzyDL* also allows to write  $\sqsubseteq_{\Rightarrow}$  in order to specify the particular implication function to be used in the semantics of the GCI (currently, Kleene-Dienes, Łukasiewicz or Gödel).

Some important particular cases of GCIs are:

- *Domain* axioms, stating that the domain of role  $R$  is the concept  $C$ , i.e.,  $\langle \exists R.T \sqsubseteq C, 1 \rangle$ .
- *Range* axioms, stating that the range of role  $R$  is the concept  $C$ , i.e.,  $\langle T \sqsubseteq \forall R.C, 1 \rangle$ .
- *Disjointness* axioms, stating that concepts  $C_1 \dots C_k$  are disjoint, i.e.,  $\forall_{i,j \in \{1, \dots, k\}, i < j}, \langle C_i \sqcap C_j \sqsubseteq \perp, 1 \rangle$

A *fuzzy RBox*  $\mathcal{R}$  is a finite set of role axioms of the form:

- $(fun\ R)$ , stating that a role  $R$  is functional, i.e.,  $R$  is a feature.
- $(trans\ R)$ , stating that a role  $R$  is transitive.
- $R_1 \sqsubseteq R_2$ , meaning that role  $R_2$  subsumes role  $R_1$ .
- $(inv\ R_1\ R_2)$ , stating that role  $R_2$  is the inverse of  $R_1$  (and vice versa).

A simple role is a role which is neither transitive nor has a transitive subroles. An important restriction is that functional needs to be simple.

## B. Semantics

The main idea is that concepts and roles are interpreted as fuzzy subsets of an interpretation's domain. Therefore, axioms, rather than being "classical" evaluated (being either true or false), they are "many-valued" evaluated in  $[0, 1]_D$ .

A *fuzzy interpretation*  $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$  relative to a fuzzy data type theory  $\mathbf{D} = \langle \Delta_{\mathbf{D}}, \cdot \rangle_{\mathbf{D}}$  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 data value, data type, and fuzzy data type predicate, and it assigns:

- to each abstract concept  $C$  a function  $C^{\mathcal{I}}: \Delta^{\mathcal{I}} \rightarrow [0, 1]_D$ ;
- to each abstract role  $R$  a function  $R^{\mathcal{I}}: \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0, 1]_D$ ;
- to each abstract feature  $r$  a partial function  $r^{\mathcal{I}}: \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0, 1]_D$  such that for all  $x \in \Delta^{\mathcal{I}}$  there is an unique  $y \in \Delta^{\mathcal{I}}$  on which  $r^{\mathcal{I}}(x, y)$  is defined;
- to each concrete role  $T$  a function  $R^{\mathcal{I}}: \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}} \rightarrow [0, 1]_D$ ;
- to each concrete feature  $t$  a partial function  $t^{\mathcal{I}}: \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}} \rightarrow [0, 1]_D$  such that for all  $x \in \Delta^{\mathcal{I}}$  there is an unique  $v \in \Delta_{\mathbf{D}}$  on which  $t^{\mathcal{I}}(x, v)$  is defined;
- to each modifier  $m$  the modifier function  $f_m: [0, 1]_D \rightarrow [0, 1]_D$ ;
- to each abstract individual  $x$  an element in  $\Delta^{\mathcal{I}}$ ;
- to each concrete individual  $v$  an element in  $\Delta_{\mathbf{D}}$ .

The mapping  $\cdot^{\mathcal{I}}$  is extended to roles and complex concepts as specified in Table I, while the mapping  $\cdot^{\mathcal{I}}$  is extended to the other constructs as specified in Table II.

The notion of *satisfaction* of a fuzzy axiom  $E$  by a fuzzy interpretation  $\mathcal{I}$ , denoted  $\mathcal{I} \models E$ , is defined as follows:

- $\mathcal{I} \models \langle \tau \geq \alpha \rangle$  iff  $\tau^{\mathcal{I}} \geq \alpha$ ,
- $\mathcal{I} \models (trans\ R)$  iff  $\forall x, y, z \in \Delta^{\mathcal{I}}, R^{\mathcal{I}}(x, y) \geq \sup_{z \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, z) \otimes R^{\mathcal{I}}(z, y)$ ,
- $\mathcal{I} \models R_1 \sqsubseteq R_2$  iff  $\forall x, y \in \Delta^{\mathcal{I}}, R_1^{\mathcal{I}}(x, y) \leq R_2^{\mathcal{I}}(x, y)$ ,
- $\mathcal{I} \models (inv\ R_1\ R_2)$  iff  $\forall x, y \in \Delta^{\mathcal{I}}, R_1^{\mathcal{I}}(x, y) = R_2^{\mathcal{I}}(y, x)$ .

We say that concept  $C$  is *satisfiable* iff there is an interpretation  $\mathcal{I}$  and an individual  $x \in \Delta^{\mathcal{I}}$  such that  $C^{\mathcal{I}}(x) > 0$ .

For a set of axioms  $\mathcal{E}$ , we say that  $\mathcal{I}$  *satisfies*  $\mathcal{E}$  iff  $\mathcal{I}$  satisfies each element in  $\mathcal{E}$ . We say that  $\mathcal{I}$  is a *model* of  $E$  (resp.  $\mathcal{E}$ ) iff  $\mathcal{I} \models E$  (resp.  $\mathcal{I} \models \mathcal{E}$ ).  $\mathcal{I}$  *satisfies* (is a *model* of) a fuzzy KB  $\mathcal{K} = \langle \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ , denoted  $\mathcal{I} \models \mathcal{K}$ , iff  $\mathcal{I}$  is a model of each component  $\mathcal{A}, \mathcal{T}$  and  $\mathcal{R}$ , respectively.

An axiom  $E$  is a *logical consequence* of a knowledge base  $\mathcal{K}$ , denoted  $\mathcal{K} \models E$  iff every model of  $\mathcal{K}$  satisfies  $E$ .

Given  $\mathcal{K}$  and a fuzzy axiom  $\tau$  of the forms  $\langle x:C, \alpha \rangle$ ,  $\langle (x, y):R, \alpha \rangle$  or  $\langle C \sqsubseteq D, \alpha \rangle$ , it is of interest to compute  $\tau$ 's best lower degree value bound. The *greatest lower bound* of  $\tau$  w.r.t.  $\mathcal{K}$  (denoted  $glb(\mathcal{K}, \tau)$ ) is  $glb(\mathcal{K}, \tau) = \sup\{n \mid \mathcal{K} \models \langle \tau \geq n \rangle\}$ , where  $\sup \emptyset = 0$ . Determining the *glb* is called the *Best Degree Bound* (BDB) problem.

$\perp^{\mathcal{I}}(x) = 0$	$(m(C))^{\mathcal{I}}(x) = f_m(C^{\mathcal{I}}(x))$
$\top^{\mathcal{I}}(x) = 1$	$(\forall R.C)^{\mathcal{I}}(x) = \inf_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \Rightarrow C^{\mathcal{I}}(y)$
$(\neg C)^{\mathcal{I}}(x) = \ominus C^{\mathcal{I}}(x)$	$(\exists R.C)^{\mathcal{I}}(x) = \sup_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \otimes C^{\mathcal{I}}(y)$
$(C \sqcap D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \otimes D^{\mathcal{I}}(x)$	$(\forall T.d)^{\mathcal{I}}(x) = \inf_{y \in \Delta^{\mathcal{I}}} T^{\mathcal{I}}(x, y) \Rightarrow d^{\mathcal{I}}(y)$
$(C \sqcap_G D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \otimes_G D^{\mathcal{I}}(x)$	$(\exists T.d)^{\mathcal{I}}(x) = \sup_{y \in \Delta^{\mathcal{I}}} T^{\mathcal{I}}(x, y) \otimes d^{\mathcal{I}}(y)$
$(C \sqcap_{\mathbf{L}} D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \otimes_{\mathbf{L}} D^{\mathcal{I}}(x)$	$(n C)^{\mathcal{I}}(x) = n C^{\mathcal{I}}(x)$
$(C \sqcup D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \oplus D^{\mathcal{I}}(x)$	$(w_1 C_1 + \dots + w_k C_k)^{\mathcal{I}}(x) = w_1 C_1^{\mathcal{I}}(x) + \dots + w_k C_k^{\mathcal{I}}(x)$
$(C \sqcup_G D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \oplus_G D^{\mathcal{I}}(x)$	$(C[\geq n])^{\mathcal{I}}(x) = \begin{cases} C^{\mathcal{I}}(x), & \text{if } C^{\mathcal{I}}(x) \geq n \\ 0, & \text{otherwise} \end{cases}$
$(C \sqcup_{\mathbf{L}} D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \oplus_{\mathbf{L}} D^{\mathcal{I}}(x)$	$(C[\leq n])^{\mathcal{I}}(x) = \begin{cases} C^{\mathcal{I}}(x), & \text{if } C^{\mathcal{I}}(x) \leq n \\ 0, & \text{otherwise} \end{cases}$
$(C \rightarrow D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x)$	$(\geq t \text{ val})^{\mathcal{I}}(x) = \sup_{c \in \Delta_D} t(x, c) \otimes (c \geq \text{val})$
$(C \rightarrow_G D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \Rightarrow_G D^{\mathcal{I}}(x)$	$(\leq t \text{ val})^{\mathcal{I}}(x) = \sup_{c \in \Delta_D} t(x, c) \otimes (c \leq \text{val})$
$(C \rightarrow_{KD} D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \Rightarrow_{KD} D^{\mathcal{I}}(x)$	$(= t \text{ val})^{\mathcal{I}}(x) = \sup_{c \in \Delta_D} t(x, c) \otimes (c = \text{val})$
$(C \rightarrow_{\mathbf{L}} D)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) \Rightarrow_{\mathbf{L}} D^{\mathcal{I}}(x)$	

TABLE I  
SEMANTICS OF THE COMPLEX FUZZY CONCEPTS.

$(x:C)^{\mathcal{I}} = C^{\mathcal{I}}(x^{\mathcal{I}})$
$((x, y):R)^{\mathcal{I}} = R^{\mathcal{I}}(x^{\mathcal{I}}, y^{\mathcal{I}})$
$(C \sqsubseteq D)^{\mathcal{I}} = \inf_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x)$
$(C \sqsubseteq_G D)^{\mathcal{I}} = \inf_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x) \Rightarrow_G D^{\mathcal{I}}(x)$
$(C \sqsubseteq_{\mathbf{L}} D)^{\mathcal{I}} = \inf_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x) \Rightarrow_{\mathbf{L}} D^{\mathcal{I}}(x)$
$(C \sqsubseteq_{KD} D)^{\mathcal{I}} = \inf_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x) \Rightarrow_{KD} D^{\mathcal{I}}(x)$

TABLE II  
SEMANTICS OF OTHER CONSTRUCTS.

Finally, another similar problem is to compute the *best satisfiability bound* of a concept  $C$  and amounts to determine  $glb(\mathcal{K}, C) = \sup_{\mathcal{I}} \sup_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) \mid \mathcal{I} \models \mathcal{K}\}$ . Essentially, among all models  $\mathcal{I}$  of the KB, we are determining the maximal degree of truth that the concept  $C$  may have over all individuals  $x \in \Delta^{\mathcal{I}}$ .

#### IV. THE *fuzzyDL* REASONER

*fuzzyDL* is a DL reasoner supporting fuzzy logic reasoning, publicly available at <http://gaia.isti.cnr.it/~straccia/software/fuzzyDL>. It has been developed in Java, using the parser generator JavaCC<sup>2</sup> and the MILP-solver Cbc<sup>3</sup>. The reasoning algorithm uses a combination of a tableaux algorithm and a MILP optimization problem. A detailed description of the reasoning algorithm cannot fit into this paper, but it can be partially found in [24], [26], [27]. In the rest of this section, we describe some of the most interesting features of the reasoner.

a) *Expressivity of the representation language*: *fuzzyDL* extends the classical DL  $\mathcal{SHIF}(D)$  to the fuzzy case. But in addition to the constructs of  $\mathcal{SHIF}(D)$  it allows some

new concepts constructs, already introduced in Section III. We recall them here since they are not supported by other fuzzy DL reasoners:

- *New concept constructs*: weighted concepts ( $n C$ ), weighted sum concepts ( $w_1 C_1 + \dots + w_k C_k$ ) and threshold concepts [6] ( $C[\geq n]$ ) and ( $C[\leq n]$ ).
- *Explicit definitions of fuzzy concepts* are allowed, by means of crisp intervals, trapezoidal, triangular, left-shoulder and right-shoulder membership functions.
- *Concept modifiers* can be represented using linear hedges and triangular functions.
- Concrete features or *datatypes*, which can have a value with is an integer, a real or a string.

Another interesting feature is that the degrees of the fuzzy axioms may not only be numerical constants, but also variables, thus being able to deal with *unknown degrees of truth*.

b) *Fuzzy operators*: The user can specify three semantics: Zadeh, Łukasiewicz and classical logics. Classical logic makes the reasoner backwards compatible, whereas Zadeh semantics is included for compatibility with most of the papers about fuzzy DLs. By default, the operators will be interpreted according to the semantics chosen by the user, but the language allows to combine arbitrarily the supported

<sup>2</sup><https://javacc.dev.java.net>

<sup>3</sup><http://www.coin-or.org/projects/Cbc.xml>

fuzzy operators. Whenever a fuzzy conjunction, a disjunction or implication is used, the user can specify its particular semantics.

The following restriction should be remarked. If Gödel implication  $\Rightarrow_G$  appears in the fuzzy KB, then the reasoner restricts the calculus to finite models. In general, Gödel logic does not have the witnessed model property, i.e., there can exist fuzzy KBs which have an infinite model, but they do not have a witnessed model (see [11] for an example). However, due to the limited precision of computers, we will deal with a finite number of truth values, and in Gödel logic over a fixed finite subset of truth values all models (finite or infinite) are witnessed [11].

c) *Constraints*: The possible values of the variables in the KB may be restricted by specifying linear inequations, as well as by restricting variables to be binary or free. Clearly, the range of the variables can be defined by means of a pair of linear inequations. As a consequence, it is possible to represent (and to solve) MILP problems using *fuzzyDL*. We allow the following constraints, with  $\bowtie \in \{\geq, \leq, =\}$ :

Constraint	Semantics
(linear inequation)	$a_1 var_1 + \dots + a_k * var_k \bowtie n$
(binary var)	$var \in \{0, 1\}$
(free var)	$var \in (-\infty, \infty)$

For instance, we may encode that “Tim is taller than Tom” by specifying  $\langle tim: Tall, x_1 \rangle, \langle tom: \neg Tall, x_2 \rangle$  with the constraint  $x_1 + x_2 = 1$ .

d) *Query language*: *fuzzyDL* allows to compute several different types of queries. It does not only allow typical reasoning tasks such as the BDB, concept satisfiability and subsumption problems, but it also allows to optimize variables and to perform defuzzifications. The concrete syntax for the queries is the following:

Query	Semantics
(max-instance? a C)	$\sup\{n \mid \mathcal{K} \models \langle a : C, n \rangle\}$
(min-instance? a C)	$\inf\{n \mid \mathcal{K} \models \langle a : C, n \rangle\}$
(max-related? a b R)	$\sup\{n \mid \mathcal{K} \models \langle (a, b) : R, n \rangle\}$
(min-related? a b R)	$\inf\{n \mid \mathcal{K} \models \langle (a, b) : R, n \rangle\}$
(max-subs? C D)	$\sup\{n \mid \mathcal{K} \models \langle D \sqsubseteq C, n \rangle\}$
(min-subs? C D)	$\inf\{n \mid \mathcal{K} \models \langle D \sqsubseteq C, n \rangle\}$
(g-max-subs? C D)	$\sup\{n \mid \mathcal{K} \models \langle D \sqsubseteq_G C, n \rangle\}$
(g-min-subs? C D)	$\inf\{n \mid \mathcal{K} \models \langle D \sqsubseteq_G C, n \rangle\}$
(l-max-subs? C D)	$\sup\{n \mid \mathcal{K} \models \langle D \sqsubseteq_{\perp} C, n \rangle\}$
(l-min-subs? C D)	$\inf\{n \mid \mathcal{K} \models \langle D \sqsubseteq_{\perp} C, n \rangle\}$
(kd-max-subs? C D)	$\sup\{n \mid \mathcal{K} \models \langle D \sqsubseteq_{KD} C, n \rangle\}$
(kd-min-subs? C D)	$\inf\{n \mid \mathcal{K} \models \langle D \sqsubseteq_{KD} C, n \rangle\}$
(max-sat? C [a])	$\sup_{\mathcal{I}} \sup_{a \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(a)$
(min-sat? C [a])	$\inf_{\mathcal{I}} \sup_{a \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(a)$
(max-var? var)	$\sup\{\text{var} \mid \mathcal{K} \text{ is consistent}\}$
(min-var? var)	$\inf\{\text{var} \mid \mathcal{K} \text{ is consistent}\}$
(defuzzify-lom? C a t)	defuzzify $t$ using LOM
(defuzzify-som? C a t)	defuzzify $t$ using SOM
(defuzzify-mom? C a t)	defuzzify $t$ using MOM

It is worth to note that in concept satisfiability queries (max-sat? and min-sat?) it is possible the use an individual

$a$  instead of an arbitrary one. This will be used, for example, in the computation of the defuzzification methods, which will be explained in Section V-B together with an example.

e) *Output*: In most of the applications, the user is not only interested in the answer of the query, but also in the fuzzy model which satisfies it. However, not all elements of the model are relevant to the user, specially if he is working with large fuzzy KBs. *fuzzyDL* provides several statements which allow the user to restrict the elements of the fuzzy model for which the value will be shown. The current available statements allow to show:

- the value of the fillers of the concrete roles  $R_1 \dots R_n$  for every individual,
- the value of the fillers of the concrete roles  $R_1 \dots R_n$  for a particular individual  $a$ ,
- the value of the variables  $x_1 \dots x_n$ ,
- the value of the instances of the atomic concepts  $A_1 \dots A_n$ ,
- the membership degree of the individuals  $a_1 \dots a_n$  to any atomic concept,
- the language of the KB, ranging from  $\mathcal{ALC}$  to  $\mathcal{SHIF}(D)$ .

f) *Import of OWL ontologies*: At this moment, there does not exist an editor allowing to create fuzzy KBs in *fuzzyDL*. Up to now, the user needs to create a text file using the syntax of the language.

In order to make the representation of fuzzy KBs easier, *fuzzyDL* also allows the possibility of importing OWL Lite ontologies. These (crisp) ontologies are saved into a text file which the user can edit and extend, for example adding membership degrees to the fuzzy axioms or specifying a particular fuzzy operator for a complex concept.

## V. SOME EXAMPLES OF USE CASES

### A. Matchmaking

Assume, that a car seller sells a sedan car. A buyer is looking for a second hand passenger car. Both the buyer as well as the seller have preferences (restrictions). Our aim is to find the best agreement. The preferences are as follows. Concerning the buyer:

- 1) He does not want to pay more than 26000 euro (buyer reservation value).
- 2) If there is an alarm system in the car then he is completely satisfied with paying no more than 22300 euro, but he can go up to 22750 euro to a lesser degree of satisfaction.
- 3) He wants a driver insurance and either a theft insurance or a fire insurance.
- 4) He wants air conditioning and the external colour should be either black or grey.
- 5) Preferably the price is no more than 22000 euro, but he can go up to 24000 euro to a lesser degree of satisfaction.
- 6) The kilometer warranty is preferably at least 175000, but he may go down to 150000 to a lesser degree of satisfaction.

- 7) The weights of the preferences 2-6 are, (0.1, 0.2, 0.1, 0.2, 0.4). The higher the value the more important is the preference.

Concerning the seller:

- 1) He wants to sell no less than 22000 euro (seller reservation value)
- 2) If there is an navigator pack system in the car then he is completely satisfied with paying no less than 22750 euro, but he can go down to 22500 euro to a lesser degree of satisfaction.
- 3) Preferably the buyer buys the Insurance Plus package.
- 4) The kilometer warranty is preferably at most 100000, but he may go up to 125000 to a lesser degree of satisfaction.
- 5) The monthly warranty is preferably at most 60, but he may go up to 72 to a lesser degree of satisfaction.
- 6) If the color is black then the car has air conditioning.
- 7) The weights of the preferences 2-6 are, (0.3, 0.1, 0.3, 0.1, 0.2). The higher the value the more important is the preference.

We have also some background theory about the domain:

- 1) A sedan is a passenger car.
- 2) A satellite alarm system is an alarm system.
- 3) The navigator pack is a satellite alarm system with a GPS system.
- 4) The Insurance Plus package is a driver insurance together with a theft insurance.
- 5) The car colours are black or grey.

Now, the background theory can be encoded as:

$Sedan \sqsubseteq PassengerCar$   
 $SatelliteAlarm \sqsubseteq AlarmSystem$   
 $NavigatorPack = SatelliteAlarm \sqcap GPS\_system$   
 $InsurancePlus = DriverInsurance \sqcap TheftInsurance$   
 $\top \sqsubseteq ExColorBlack \sqcup ExColorGray$   
 $ExColorBlack \sqcap ExColorGray \sqsubseteq \perp$   
 $(fun\ HasAlarmSystem)$   
 $(fun\ HasAirConditioning)$   
 $(fun\ HasExColor)$   
 $(fun\ HasNavigator)$   
 $(fun\ HasMWarranty)$   
 $(fun\ HasPrice)$   
 $(fun\ HasKMWarranty)$

The buyer's preferences can be encoded as follows:

- 1)  $B = (PassengerCar \sqcap (\leq HasPrice\ 26000))$
- 2)  $B_1 = ((\exists HasAlarmSystem.AlarmSystem) \rightarrow (\exists HasPrice.L(22300, 22750)))$
- 3)  $B_2 = ((\exists HasInsurance.DriverInsurance) \sqcap ((\exists HasInsurance.TheftInsurance) \sqcup (\exists HasInsurance.FireInsurance)))$
- 4)  $B_3 = ((\exists HasAirConditioning.Airconditioning) \sqcap (\exists HasExColor.(ExColorBlack \sqcup ExColorGray)))$
- 5)  $B_4 = (\exists HasPrice.L(22000, 24000))$
- 6)  $B_5 = (\exists HasKMWarranty.R(15000, 175000))$
- 7)  $Buy = (B \sqcap ((0.1B_1) + (0.2B_2) + (0.1B_3) + (0.2B_4) + (0.4B_5)))$

Please note that the concept *Buy* collects all the buyer's preferences together in such a way that the higher is the maximal degree of satisfiability of *Buy* (i.e.,  $glb(\mathcal{K}, Buy)$ ), the more the buyer is satisfied.

The seller's preferences can be encoded as follows:

- 1)  $S = (Sedan \sqcap (\geq HasPrice\ 22000))$
- 2)  $S_1 = ((\exists HasNavigator.NavigatorPack) \rightarrow (\exists HasPrice.R(22500, 22750)))$
- 3)  $S_2 = (\exists HasInsurance.InsurancePlus)$
- 4)  $S_3 = (\exists HasKMWarranty.L(100000, 125000))$
- 5)  $S_4 = (\exists HasMWarranty.L(60, 72))$
- 6)  $S_5 = ((\exists HasExColor.ExColorBlack) \rightarrow (\exists HasAirConditioning.AirConditioning))$
- 7)  $Sell = (S \sqcap ((0.3S_1) + (0.1S_2) + (0.3S_3) + (0.1S_4) + (0.2S_5)))$

Similarly to the buyer case, the concept *Sell* collects all the seller's preferences together in such a way that the higher is the maximal degree of satisfiability of *Sell* (i.e.,  $glb(\mathcal{K}, Sell)$ ), the more the seller is satisfied.

Now, it is clear that the best agreement among the buyer and the seller is determined by the maximal degree of satisfiability of the conjunction  $Buy \sqcap Sell$ , i.e., we have to determine  $glb(\mathcal{K}, Buy \sqcap Sell)$ . In particular, we ask the *fuzzyDL* solver to answer to the query (by relying on Łukasiewicz conjunction, we have the guarantee that the solution is Pareto optimal [20])

(show-fillers *HasPrice HasKMWarranty HasMWarranty*)  
(max-sat? ( $Buy \sqcap_{\mathbb{L}} Sell$ ))

which determines  $glb(\mathcal{K}, Buy \sqcap_{\mathbb{L}} Sell)$  and displays also the price and the warranties of the agreement. In particular, we have that

$$\begin{aligned}
 glb(\mathcal{K}, Buy \sqcap_{\mathbb{L}} Sell) &= 0.7 \\
 HasPrice &= 22000.0 \\
 HasKMWarranty &= 175000.0 \\
 HasMWarranty &= 0.0 .
 \end{aligned}$$

So an optimal match (the Pareto optimal degree is 0.7625) would be an agreement with a price of 22500 euro, with 100000 kilometer warranty and a 60 month warranty.

### B. Fuzzy control

In this subsection we will describe how to use *fuzzyDL* to represent Mamdani fuzzy IF-THEN rules and to reason with them (differently from [1], which only allows to represent fuzzy controllers). The interesting thing is that it is not only possible to represent a fuzzy control problem, but also background knowledge related to it.

In order to represent a fuzzy control problem we proceed as follows. Firstly, for each variable of the system, we define a concrete feature representing it and specify its range (a subset of the reals  $[k_1, k_2]$ ). Then, we define the different linguistic labels, which will be used to describe the value of these variables, using triangular fuzzy concepts. Next, we define a concept resuming the rule base. It is well known, that a set  $\{r_1, \dots, r_p\}$  of Mamdani rules  $r_i$  of the form (1) can be transformed in a concept *Mamd* of the form  $\bigoplus_{i=\{1, \dots, p\}} (A_{i1} \otimes \dots \otimes A_{in} \otimes B_i)$  [10].

In general,  $\oplus$  is assumed to be the Gödel t-conorm and  $\otimes$  the Gödel t-norm, but we recall that *fuzzyDL* allows to use other fuzzy operators. Then, we represent the input of the system as fuzzy assertions involving an individual *fc*.

For the sake of concrete illustration, consider an adaption of the simple example available at [http://en.wikipedia.org/wiki/Fuzzy\\_system](http://en.wikipedia.org/wiki/Fuzzy_system). It considers the design of a fuzzy controller for an anti-lock braking system. The encoding in *fuzzyDL* is shown in Table III. The system has two input variables, *temperature* and *pressure*, which are used to compute an output value which will adjust the *throttle*.

- *temperature* has 5 labels associated: *cold*, *cool*, *nominal*, *warm* and *hot*.
- *pressure* has 5 labels: *weak*, *low*, *ok*, *strong* and *high*.
- *throttle* has 7 labels: *N3* (large negative), *N2* (medium negative), *N1* (small negative), *Z* (zero), *P1* (small positive), *P2* (medium positive) and *P3* (large positive).

The rules of the system are the following:

- 1) IF temperature IS cool AND pressure IS weak, THEN throttle is P3.
- 2) IF temperature IS cool AND pressure IS low, THEN throttle is P2.
- 3) IF temperature IS cool AND pressure IS ok, THEN throttle is Z.
- 4) IF temperature IS cool AND pressure IS strong, THEN throttle is N2.

The linguistic labels of the variables are defined using fuzzy concrete concepts. For example, *TempCool*, the label representing that the temperature is cool, can be defined as  $\text{triangular}(480, 500, 520)$ , *ThrottleP2* as  $\text{triangular}(10, 15, 20)$  and *ThrottleZ* as  $\text{triangular}(15, 20, 25)$ .

Now assume that the temperature is 489.6, which (in abstract syntax) is represented using the assertion  $\langle fc : (= \text{temperature } 489.6) \rangle$ . Then,  $\mu_{\text{cool}}(489.6) = \text{triangular}(480, 500, 520)(489.6) = 0.48$ . We also assume that the pressure is  $p$  and that  $\mu_{\text{low}}(p) = 0.57$ ,  $\mu_{\text{ok}}(p) = 0.25$  and  $\mu_{\text{weak}}(p) = \mu_{\text{strong}}(p) = \mu_{\text{high}}(p) = 0$ . That is, the temperature is in the *cool* state, the pressure is in the *low* and *ok* states, in such a way that only rules 2 and 3 are fired.

Rule 2 is fired with a degree  $\min\{0.48, 0.57\} = 0.48$ , so *throttle* is *P2* with degree 0.48. Rule 3 is fired with a degree  $\min\{0.48, 0.25\} = 0.25$ , so *throttle* is *Z* with degree 0.25. These two values for *throttle* are aggregated using the maximum. Then, defuzzification step is performed. It can be verified that  $LOM = 17.6$ ,  $MOM = 15$  and  $SOM = 12.4$ .

For a concrete feature  $t$  representing an output variable of the system, LOM is implemented in the following steps.

- 1) Compute the maximum degree  $\gamma$  of satisfiability of concept *Mamd* (using individual  $fc$ ), that is, we determine (max-sat? *Mamd*  $fc$ ).
- 2) Ensure that  $fc$  belongs to *Mamd* with degree  $\gamma$ . i.e., we add the assertion  $\langle fc:\text{Mamd}, \gamma \rangle$  to the KB.
- 3) Maximize the value of the (internal) variable representing the value of  $t$ .

The SOM can be computed in a similar way, but minimizing the value of the (internal) variable representing the value of  $t$ . Finally, MOF can be computed as  $(LOM + SOM)/2$ .

## VI. CONCLUSIONS

The aim of this paper was to present salient features of the *fuzzyDL* system (in terms of syntax and semantics), which

is an expressive reasoner for fuzzy DLs. We have shown that *fuzzyDL* significantly extends fuzzy *SHIF* by allowing several additional features. We have also shown two use cases, namely logic-based matchmaking and fuzzy control, which are not supported by any other fuzzy DL system so far (see below).

### A. Related work

There exist some other reasoners for fuzzy DLs (in chronological order):

- *Fire* [22] implements a tableau algorithm for fuzzy *SHIN* restricted to the Zadeh semantics. An interesting feature is its graphical interface, although users need to deal directly with the syntax of the language for the representation of the fuzzy KB.
- *GURDL* [8] supports an extension of *ALC* with an abstract and more general notion of uncertainty. The reasoning algorithm is also based on a mixture of tableau rules and the resolution of a set of inequations. Moreover, they propose some interesting techniques of optimization and study the applicability of some techniques used in the crisp case.
- *DeLorean* [4] reduces reasoning in fuzzy *SHOIN* under Zadeh semantics to reasoning in crisp *SHOIN*, considering some optimizations. As a consequence, it allows to reuse classical languages and resources (editors, tools, reasoners ...).
- *GERDS* [9] implements a resolution algorithm for fuzzy *ALC* under Łukasiewicz logic, extended with role negation, top role and bottom role.
- *YADLR* [15] is a recent implementation of a resolution-based algorithm for Łukasiewicz logic. The authors have claimed that all existing implementations require that the degrees of all assertions in the KB are numerical constants and that *YADLR* is the first reasoner allowing also variables, but *fuzzyDL* has supported this since its first version (published in Internet on June 2006).

While *Fire* and *DeLorean* allow the use of some DL constructs which *fuzzyDL* (still) do not support (cardinality restrictions and, in the case of *DeLorean*, also nominals) and *GURDL* supports a more general representation of uncertainty, not being limited to fuzzy logic, none of them support most of the already discussed features of *fuzzyDL*, especially in what regards expressivity of the representation. That are, explicit fuzzy sets, concepts modifiers, data types, defuzzification ..., which we consider as the most important features for fuzzy DL languages.

### B. Future work

*fuzzyDL* reasoner is in constant update. Future work will include the extension of the expressivity of the logic (especially the implementation of the algorithms to reason with a product family of fuzzy operators [5] and with fuzzy role hierarchies [26]), the development of a graphical interface for fuzzy ontologies representation and the implementation of some optimization techniques to reduce the running time.

```

% Concrete features
(fun temperature)
(fun pressure)
(fun throttle)

% Labels for the variables
TempCool = triangular(480,500,520),...
PressLow = triangular(30,40,45), PressOK = triangular(40,48.6,57.2),...
ThrottleP2 = triangular(10,15,20), ThrottleZ = triangular(15,20,25),...

% Encoding of Mamdami Rules
Rule1 = (( $\exists$ temperature.TempCool)  $\sqcap_G$  ( $\exists$ pressure.PressWeak)  $\sqcap_G$  ( $\exists$ throttle.ThrottleP3))
Rule2 = (( $\exists$ temperature.TempCool)  $\sqcap_G$  ( $\exists$ pressure.PressLow)  $\sqcap_G$  ( $\exists$ throttle.ThrottleP2))
Rule3 = (( $\exists$ temperature.TempCool)  $\sqcap_G$  ( $\exists$ pressure.PressOK)  $\sqcap_G$  ( $\exists$ throttle.ThrottleZ))
Rule4 = (( $\exists$ temperature.TempCool)  $\sqcap_G$  ( $\exists$ pressure.PressStrong)  $\sqcap_G$  ( $\exists$ throttle.ThrottleN2))

% Encoding of Mamdami Rule base
Mamd = (Rule1  $\sqcup_G$  Rule2  $\sqcup_G$  Rule3)

% Input to the controller
fc:(= temperature 489.6)
fc:(= pressure 42.15)

% Output of the controller
(defuzzify-lom? Mamd fc throttle)

```

TABLE III  
FRAGMENT OF AN EXAMPLE OF FUZZY KB FOR FUZZY CONTROL.

#### ACKNOWLEDGMENT

The research of F. Bobillo has been partially supported by a FPU scholarship and project TIN2006-15041-C04-01, both from the Spanish Ministerio de Educación y Ciencia.

#### REFERENCES

- [1] G. Acampora and V. Loia. Using FML and fuzzy technology in adaptive ambient intelligence environments. *International Journal of Computational Intelligence Research* 1(2):171–182, 2005.
- [2] F. Baader, D. Calvanese, D. McGuinness, D. Nardi and P. F. Patel-Schneider, editors. *The description logic handbook: Theory, implementation, and applications*. Cambridge University Press, 2003.
- [3] T. Berners-Lee, J. Hendler and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.
- [4] F. Bobillo, M. Delgado and J. Gómez-Romero. Optimizing the crisp representation of the fuzzy description logic *SROIQ*. In *Proceedings of the 3rd ISWC Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2007)*. CEUR Workshop Proceedings, 2007.
- [5] F. Bobillo and U. Straccia. A fuzzy description logic with product t-norm. In *Proceedings of the 16th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007)*, pp. 652–657, 2006.
- [6] S. Calegari and D. Ciucci. Fuzzy ontology, fuzzy description logics and fuzzy-OWL. In *Proceedings of the 7th International Workshop on Fuzzy Logic and Applications (WILF 2007)*, volume 4578 of *Lecture Notes in Computer Science*, pp. 118–126. Springer Verlag, 2007.
- [7] C. Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence* 19:17–37, 1982.
- [8] V. Haarslev, H. I. Pai and N. Shiri. Optimizing tableau reasoning in *ALC* extended with uncertainty. In *Proceedings of the 2007 International Workshop on Description Logics (DL-2007)*, pp. 307–314, 2007.
- [9] H. Habiballa. Resolution strategies for fuzzy description logic. In *Proceedings of the 5th Conference of the European Society for Fuzzy Logic and Technology (EUSFLAT 2007)*, vol. 2, pp. 27–36, 2007.
- [10] P. Hájek. *Metamathematics of fuzzy logic*. Kluwer, 1998.
- [11] P. Hájek. Making fuzzy description logics more general. *Fuzzy Sets and Systems*, 154 (1), 1–15, 2005.
- [12] S. Hölldobler, H.-P. Störr and T. D. Khang. The fuzzy description logic *ALCFH* with hedge algebras as concept modifiers. *Journal of Advanced Computational Intelligence*, 7(3):294–305, 2003.
- [13] I. Horrocks and P. F. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. *Journal of Web Semantics*, 1(4):345–357, 2004.
- [14] G. Klir and B. Yuan. *Fuzzy sets and fuzzy logic: Theory and applications*. Prentice-Hall, 1995.
- [15] S. Konstantopoulos and G. Apostolikas. Fuzzy-DL reasoning over unknown fuzzy degrees. In *Proceedings of the 3rd International Workshop on Semantic Web and Web Semantics (SWWS 07)*, 2007.
- [16] T. Lukasiewicz and U. Straccia. An overview of uncertainty and vagueness in description logics for the semantic web. Technical Report INFOSYS RR-1843-06-07, Technische Universität Wien, 2006.
- [17] C. Lutz. Description logics with concrete domains - A survey. In *Advances in Modal Logics Volume 4*. King's College, 2003.
- [18] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies* 7(1):1–13, 1975.
- [19] D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview, 2004. Retrieved October 1st, 2007, from <http://www.w3.org/TR/owl-features>.
- [20] A. Ragone, U. Straccia, F. Bobillo, T. Di Noia, E. Di Sciascio and F. M. Donini. A fuzzy description logic based approach to matchmaking in e-marketplaces. Submitted.
- [21] H. Salkin and M. Kamlesh. *Foundations of integer programming*. North-Holland, 1988.
- [22] G. Stoilos, N. Simou, G. Stamou and S. Kollias. Uncertainty and the semantic web. *IEEE Intelligent Systems* 21(5): 84–87, 2006.
- [23] U. Straccia. Reasoning within fuzzy description logics. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.
- [24] U. Straccia. Description logics with fuzzy concrete domains. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pp. 559–567. AUA Press, 2005.
- [25] U. Straccia. A fuzzy description logic for the semantic web. In E. Sanchez, editor, *Fuzzy Logic and the Semantic Web*, Capturing Intelligence, chapter 4, pp. 73–90. Elsevier, 2006.
- [26] U. Straccia. Reasoning in *L-SHLF*: an expressive fuzzy description logic under Łukasiewicz semantics. Technical report ISTI-018/2007, ISTI - CNR, 2007.
- [27] U. Straccia and F. Bobillo. Mixed integer programming, general concept inclusions and fuzzy description logics. In *Mathware & Soft Computing* 14(3):247–259, 2007.
- [28] C. Tresp and R. Molitor. A description logic for vague knowledge. In *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, 1998.
- [29] J. Yen. Generalizing term subsumption languages to fuzzy logic. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, pp. 472–477, 1991.
- [30] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.