Fuzzy Description Logic Programs

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

Abstract

Description Logic Programs (DLPs), which combine the expressive power of classical description logics and logic programs, are emerging as an important ontology description language paradigm. In this work, we present fuzzy DLPs, which extend DLPs by allowing the representation of vague/imprecise information.

Keywords: Fuzziness, Description Logics, Logic Programs

1 Introduction

Rule-based and object-oriented techniques are rapidly making their way into the infrastructure for representing and reasoning about the Semantic Web: combining these two paradigms emerges as an important objective.

Description Logic Programs (DLPs) [6, 7, 9, 10, 11, 20, 26], which combine the expressive power of classical *Description Logics* (DLs) and classical *Logic Programs* (LPs), are emerging as an important ontology description language paradigm. DLs capture the meaning of the most popular features of structured representation of knowledge, while LPs are powerful rule-based representation languages.

In this work, we present *fuzzy* DLPs, which is a novel extension of DLPs towards the representation of vague/imprecise information.

We proceed as follows. We first introduce the main notions related to fuzzy DLs and fuzzy LPs, and then show how both can be integrated, defining fuzzy DLPs in Section 3. Section 4 concludes and outlines future research.

2 Preliminaries

Fuzzy DLs. DLs [1] are a family of logics for representing structured knowledge. Each logic is identified by a name made of labels, which identify the operators allowed in that logic. Major DLs are the so-called logic $\mathcal{ALC}$ [35] ($\mathcal{A}$ttributive $\mathcal{L}$anguage with $\mathcal{C}$omplement) and is used as a reference language whenever new concepts are introduced in DLs, $\mathcal{SHOIN}(\mathtt{D})$, which is the logic behind the ontology description language OWL DL and $\mathcal{SHIF}(\mathtt{D})$, which is the logic behind OWL LITE, a slightly less expressive language than OWL DL (see [18, 21]).

Fuzzy DLs [37, 41] extend classical DLs by allowing to deal with *fuzzy/imprecise concepts*, like “Calla is a very large, long white flower on thick stalks”, allowing to deal with so-called *fuzzy* or *vague concepts*, like “creamy”, “dark”, “hot”, “large” and “thick”, for which a clear and precise definition is not possible (another issue relates to the representation of terms like “very”, which are called fuzzy concepts *modifiers*). While in classical DLs concepts denotes sets, in fuzzy DLs fuzzy concepts denote fuzzy sets [45].

Syntax. While the method we rely on in combining fuzzy DLs with fuzzy LPs does not depend on the particular fuzzy DL of choice, to make the paper self-contained, we shall use here fuzzy $\mathcal{ALC}(\mathtt{D})$ [38], which is fuzzy $\mathcal{ALC}$ [37] extended with explicit represent membership functions for modifiers (such as “very”) and vague concepts (such as “Young”) [38]. We refer to [41] for fuzzy OWL DL and related work on fuzzy DLs.

Fuzzy $\mathcal{ALC}(\mathtt{D})$ allows explicitly to represent membership functions in the language via fuzzy concrete domains. A *fuzzy concrete*

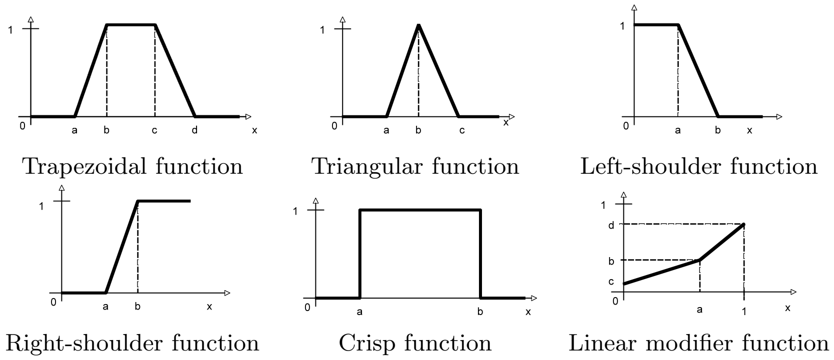

Figure 1: Membership functions and modifiers

domain (or simply *fuzzy domain*) is a pair $\langle \Delta_{\mathtt{D}}, \Phi_{\mathtt{D}} \rangle$, where $\Delta_{\mathtt{D}}$ is an interpretation domain and $\Phi_{\mathtt{D}}$ is the set of *fuzzy domain predicates* d with a predefined arity n and an interpretation $d^{\mathtt{D}}\colon \Delta_{\mathtt{D}}^n \to [0,1]$, which is a n-ary fuzzy relation over $\Delta_{\mathtt{D}}$. To the ease of presentation, we assume the fuzzy predicates have arity one, the domain is a subset of the rational numbers $\mathbb{Q}$ and the range is $[0,1]_{\mathbb{Q}} = [0,1] \cap \mathbb{Q}$. Concerning fuzzy predicates, there are many membership functions for fuzzy sets membership specification. However (see Figure 1), for $k_1 \leq a < b \leq c < d \leq k_2$ rational numbers, the *trapezoidal* $trz(a,b,c,d,[k_1,k_2])$, the *triangular* $tri(a,b,c,[k_1,k_2])$, the *left-shoulder* $ls(a,b,[k_1,k_2])$, the *right-shoulder* $rs(a,b,[k_1,k_2])$ and the *crisp function* $cr(a,b,[k_1,k_2])$ are simple, yet most frequently used to specify membership degrees and are those we are considering in this paper. To simplify the notation, we may omit the domain range, and write, e.g. $cr(a,b)$ in place of $cr(a,b,[k_1,k_2])$, whenever the domain range is not important. For instance, the concept "less than 18 year old" can be defined as a crisp concept $cr(0,18)$, while `Young`, denoting the degree of youngness of a person's age, may be defined as $\mathtt{Young} = ls(10,30)$. We also consider fuzzy modifiers in fuzzy $\mathcal{ALC}(\mathtt{D})$. Fuzzy modifiers, like `very`, `more_or_less` and `slightly`, apply to fuzzy sets to change their membership function. Formally, a *modifier* is a function $f_m\colon [0,1] \to [0,1]$. For instance, we may define $\mathtt{very}(x) = \mathtt{lm}(0.7, 0.49, 0, 1)$, while define $\mathtt{slightly}(x)$ as $\mathtt{lm}(0.7, 0.49, 1, 0)$, where $lm(a,b,c,d)$ is the *linear modifier* in Figure 1.

Now, let $\mathtt{C}$, $\mathtt{R}_a$, $\mathtt{R}_c$, $\mathtt{I}_a$, $\mathtt{I}_c$ and $\mathtt{M}$ 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 constant names* (denoted a), *concrete constant names* (denoted c) and *modifiers* (denoted m). $\mathtt{R}_a$ contains a non-empty subset $\mathtt{F}_a$ of *abstract feature names* (denoted r), while $\mathtt{R}_c$ contains a non-empty subset $\mathtt{F}_c$ of *concrete feature names* (denoted t). Features are functional roles. The set of fuzzy $\mathcal{ALC}(\mathtt{D})$ *concepts* is defined by the syntactic rules (d is a unary fuzzy predicate) in Figure 2. A *TBox* $\mathcal{T}$ consists of a finite set of *terminological axioms* of the form $C_1 \sqsubseteq C_2$ (C_1 is sub-concept of C_2) or $A = C$ (A is defined as the concept C), where A is a concept name and C is concept. Using axioms we may define the concepts of a minor and young person as

$$\mathtt{Minor} = \mathtt{Person} \sqcap \exists \mathtt{age}.\leq_{18} \quad (1)$$
$$\mathtt{YoungPerson} = \mathtt{Person} \sqcap \exists \mathtt{age}.\mathtt{Young} \quad (2)$$

We also allow to formulate statements about constants. A *concept-*, *role- assertion axiom* and an *constant (in)equality axiom* has the form $a\colon C$ (a is an instance of C), $(a,b)\colon R$ (a is related to b via R), $a \approx b$ (a and b are equal) and $a \not\approx b$, respectively, where a, b are abstract constants. For $n \in [0,1]_{\mathbb{Q}}$, an *ABox* $\mathcal{A}$ consists of a finite set of constant (in)equality axioms, and *fuzzy concept* and *fuzzy role assertion axioms* of the form $\langle \alpha, n \rangle$, where α is a concept or role assertion. Informally, $\langle \alpha, n \rangle$ constrains the truth degree of α to be greater or equal to n. A fuzzy $\mathcal{ALC}(\mathtt{D})$ *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$.

Semantics. We recall here the main notions related to fuzzy DLs (for more on fuzzy DLs,

$$
\begin{array}{lll}
C & \longrightarrow & \top \mid \bot \mid A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \forall R.C \mid \exists R.C \mid \forall T.D \mid \exists T.D \mid m(C) \\
D & \rightarrow & d \mid \neg d \\
m & \rightarrow & \mathtt{lm}(\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d}) \\
d & \rightarrow & \mathtt{trz}(\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d},[\mathtt{k}_1,\mathtt{k}_2]) \mid \mathtt{tri}(\mathtt{a},\mathtt{b},\mathtt{c},[\mathtt{k}_1,\mathtt{k}_2]) \mid \mathtt{ls}(\mathtt{a},\mathtt{b},[\mathtt{k}_1,\mathtt{k}_2]) \mid \mathtt{rs}(\mathtt{a},\mathtt{b},[\mathtt{k}_1,\mathtt{k}_2]) \mid \mathtt{cr}(\mathtt{a},\mathtt{b},[\mathtt{k}_1,\mathtt{k}_2])
\end{array}
$$

Figure 2: $\mathcal{ALC}(\mathtt{D})$ concepts

	Lukasiewicz Logic	Gödel Logic	Product Logic	"Zadeh semantics"
$\neg x$	$1 - x$	if $x = 0$ then 1 else 0	if $x = 0$ then 1 else 0	$1 - x$
$x \wedge y$	$\max(x + y - 1, 0)$	$\min(x, y)$	$x \cdot y$	$\min(x, y)$
$x \vee y$	$\min(x + y, 1)$	$\max(x, y)$	$x + y - x \cdot y$	$\max(x, y)$
$x \Rightarrow y$	if $x \leq y$ then 1 else $1 - x + y$	if $x \leq y$ then 1 else y	if $x \leq y$ then 1 else y/x	$\max(1 - x, y)$

Table 1: Typical connective interpretation.

see [37, 41]). The main idea is that an assertion $a\colon C$, rather being interpreted as either true or false, will be mapped into a truth value $c \in [0,1]_{\mathbb{Q}}$. The intended meaning is that c indicates to which extend 'a is a C'. Similarly for role names. Formally, a *fuzzy interpretation* $\mathcal{I}$ with respect to a concrete domain $\mathtt{D}$ is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a non empty set $\Delta^{\mathcal{I}}$ (called the *domain*), disjoint from $\Delta_{\mathtt{D}}$, and of a *fuzzy interpretation function* $\cdot^{\mathcal{I}}$ that assigns (i) to each abstract concept $C \in \mathtt{C}$ a function $C^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \to [0,1]$; ($ii$) to each abstract role $R \in \mathtt{R}_a$ a function $R^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$; ($iii$) to each abstract feature $r \in \mathtt{F}_a$ a partial function $r^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$ such that for all $u \in \Delta^{\mathcal{I}}$ there is an unique $w \in \Delta^{\mathcal{I}}$ on which $r^{\mathcal{I}}(u,w)$ is defined; (iv) to each abstract constant $a \in \mathtt{I}_a$ an element in $\Delta^{\mathcal{I}}$; (v) to each concrete constant $c \in \mathtt{I}_c$ an element in $\Delta_{\mathtt{D}}$; (vi) to each concrete role $T \in \mathtt{R}_c$ a function $T^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta_{\mathtt{D}} \to [0,1]$; ($vii$) to each concrete feature $t \in \mathtt{F}_c$ a partial function $t^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta_{\mathtt{D}} \to [0,1]$ such that for all $u \in \Delta^{\mathcal{I}}$ there is an unique $o \in \Delta_{\mathtt{D}}$ on which $t^{\mathcal{I}}(u,o)$ is defined; ($viii$) to each modifier $m \in \mathtt{M}$ the function $f_m\colon [0,1] \to [0,1]$; ($ix$) to each unary concrete predicate d the fuzzy relation $d^{\mathtt{D}}\colon \Delta_{\mathtt{D}} \to [0,1]$ and to $\neg d$ the negation of $d^{\mathtt{D}}$. To extend the interpretation function to complex concepts, we use so-called *t-norms* (interpreting conjunction), *s-norms* (interpreting disjunction), *negation function* (interpreting negation), and *implication function* (interpreting implication) [17]. In Table 1 we report most used combinations of norms.

The mapping $\cdot^{\mathcal{I}}$ is then extended to concepts and roles as follows (where $u \in \Delta^{\mathcal{I}}$): $\top^{\mathcal{I}}(u) = 1$, $\bot^{\mathcal{I}}(u) = 0$,

$$
\begin{array}{rcl}
(C_1 \sqcap C_2)^{\mathcal{I}}(u) & = & {C_1}^{\mathcal{I}}(u) \wedge {C_2}^{\mathcal{I}}(u) \\
(C_1 \sqcup C_2)^{\mathcal{I}}(u) & = & {C_1}^{\mathcal{I}}(u) \vee {C_2}^{\mathcal{I}}(u) \\
(\neg C)^{\mathcal{I}}(u) & = & \neg C^{\mathcal{I}}(u) \\
(m(C))^{\mathcal{I}}(u) & = & f_m(C^{\mathcal{I}}(u)) \\
(\forall R.C)^{\mathcal{I}}(u) & = & \inf_{w \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(u,w) \Rightarrow C^{\mathcal{I}}(w) \\
(\exists R.C)^{\mathcal{I}}(u) & = & \sup_{w \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(u,w) \wedge C^{\mathcal{I}}(w) \\
(\forall T.D)^{\mathcal{I}}(u) & = & \inf_{o \in \Delta_{\mathtt{D}}} T^{\mathcal{I}}(u,o) \Rightarrow D^{\mathcal{I}}(o) \\
(\exists T.D)^{\mathcal{I}}(u) & = & \sup_{o \in \Delta_{\mathtt{D}}} T^{\mathcal{I}}(u,o) \wedge D^{\mathcal{I}}(o) \; .
\end{array}
$$

The mapping $\cdot^{\mathcal{I}}$ is extended to assertion axioms as follows (where $a, b \in \mathtt{I}_a$): $(a\colon C)^{\mathcal{I}} = C^{\mathcal{I}}(a^{\mathcal{I}})$ and $((a,b)\colon R)^{\mathcal{I}} = R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}})$. The notion of *satisfiability* of a fuzzy axiom E by a fuzzy interpretation $\mathcal{I}$, denoted $I \models E$, is defined as follows: $I \models C_1 \sqsubseteq C_2$ iff for all $u \in \Delta^{\mathcal{I}}, {C_1}^{\mathcal{I}}(u) \leq {C_2}^{\mathcal{I}}(u)$; $I \models A = C$ iff for all $u \in \Delta^{\mathcal{I}}, A^{\mathcal{I}}(u) = C^{\mathcal{I}}(u)$; $I \models \langle \alpha, n \rangle$ iff $\alpha^{\mathcal{I}} \geq n$; $\mathcal{I} \models a \approx b$ iff $a^{\mathcal{I}} = b^{\mathcal{I}}$; and $\mathcal{I} \models a \not\approx b$ iff $a^{\mathcal{I}} \neq b^{\mathcal{I}}$. The notion of *satisfiability* (is *model*) of a knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ and *entailment* of an assertional axiom is straightforward. Concerning terminological axioms, we also introduce degrees of subsumption. We say that $\mathcal{K}$ *entails* $C_1 \sqsubseteq C_2$ to degree $n \in [0,1]$, denoted $\mathcal{K} \models \langle C_1 \sqsubseteq C_2, n \rangle$ iff for every model $\mathcal{I}$ of $\mathcal{K}$, $[\inf_{u \in \Delta^{\mathcal{I}}} {C_1}^{\mathcal{I}}(u) \Rightarrow {C_2}^{\mathcal{I}}(u)] \geq n$.

Example 1 ([41]) *Consider the following simplified excerpt from a knowledge base about cars:*

$$
\begin{array}{l}
\mathtt{SportsCar} = \exists \mathtt{speed.very}(\mathtt{High}), \\
\langle \mathtt{mg\_mgb}\colon \exists \mathtt{speed}.\leq_{170}, 1 \rangle \\
\langle \mathtt{ferrari\_enzo}\colon \exists \mathtt{speed}.>_{350}, 1 \rangle, \\
\langle \mathtt{audi\_tt}\colon \exists \mathtt{speed}.=_{243}, 1 \rangle
\end{array}
$$

speed *is a concrete feature. The fuzzy domain predicate* High *has membership function* High $= rs(80, 250)$*. It can be shown that*

$$\mathcal{K} \models \langle \texttt{mg\_mgb}\colon \neg\texttt{SportsCar}, 0.72\rangle$$
$$\mathcal{K} \models \langle \texttt{ferrari\_enzo}\colon \texttt{SportsCar}, 1\rangle$$
$$\mathcal{K} \models \langle \texttt{audi\_tt}\colon \texttt{SportsCar}, 0.92\rangle \ .$$

Note how the maximal speed limit of the mg_mgb *car* (≤ 170) *induces an upper limit,* $0.28 = 1 - 0.72$*, on the membership degree of being* mg_mgb *a* SportsCar*.*

Example 2 *Consider* $\mathcal{K}$ *with terminological axioms* (1) *and* (2)*. Then under Zadeh logic* $\mathcal{K} \models \langle \texttt{Minor} \sqsubseteq \texttt{YoungPerson}, 0.5\rangle$ *holds.*

Finally, given $\mathcal{K}$ and an axiom α, it is of interest to compute its best lower degree bound. The *greatest lower bound* of α w.r.t. $\mathcal{K}$, denoted $glb(\mathcal{K}, \alpha)$, is $glb(\mathcal{K}, \alpha) = \sup\{n\colon \mathcal{K} \models \langle \alpha, n\rangle\}$, where $\sup \emptyset = 0$. Determining the *glb* is called the *Best Degree Bound* (BDB) problem. For instance, the entailments in Examples 1 and 2 are the best possible degree bounds. Note that, $\mathcal{K} \models \langle \alpha, n\rangle$ iff $glb(\mathcal{K}, \alpha) \geq n$. Therefore, the BDB problem is the major problem we have to consider in fuzzy $\mathcal{ALC}(\texttt{D})$.

Fuzzy LPs. The management of imprecision in logic programming has attracted the attention of many researchers and numerous frameworks have been proposed. Essentially, they differ in the underlying truth space (e.g. *Fuzzy set theory* [2, 8, 22, 24, 36, 43, 44], *Multi-valued logic* [3, 4, 5, 13, 14, 23, 25, 27, 28, 29, 30, 31, 32, 34, 33, 40, 39]), and how imprecision values, associated to rules and facts, are managed.

Syntax. We consider here a very general form of the rules [39, 40]:

$$A \leftarrow f(B_1, ..., B_n) \ , \qquad (3)$$

where $f \in \mathcal{F}$ is an n-ary computable monotone function $f : [0,1]^n_{\mathbb{Q}} \to [0,1]_{\mathbb{Q}}$ and B_i are atoms. Each rule may have a different f. An example of rule is

$$s \leftarrow \min(p, q) \cdot \max(\neg r, 0.7) + v \ ,$$

where p, q, r, s and v are atoms. Computationally, given an assignment I of values to the B_i, the value of A is computed by stating that A is at least as true as $f(I(B_1), ..., I(B_n))$. The form of the rules is sufficiently expressive to encompass all approaches to fuzzy logic programming we are aware of. We assume that the standard functions $\wedge$ (meet) and $\vee$ (join) belong to $\mathcal{F}$. Notably, $\wedge$ and $\vee$ are both monotone. We call $f \in \mathcal{F}$ a *truth combination function*, or simply *combination function* [1]. We recall that an *atom*, denoted A, is an expression of the form $P(t_1, \ldots, t_n)$, where P is an n-ary predicate symbol and all t_i are terms, i.e. a *constant* or a *variable.* A *generalized normal logic program*, or simply *normal logic program*, denoted with $\mathcal{P}$, is a finite set of rules. The *Herbrand universe* $H_{\mathcal{P}}$ of $\mathcal{P}$ is the set of constants appearing in $\mathcal{P}$. If there is no constant symbol in $\mathcal{P}$ then consider $H_{\mathcal{P}} = \{a\}$, where a is an arbitrary chosen constant. The *Herbrand base* $B_{\mathcal{P}}$ of $\mathcal{P}$ is the set of ground instantiations of atoms appearing in $\mathcal{P}$ (ground instantiations are obtained by replacing all variable symbols with constants of the Herbrand universe). Given $\mathcal{P}$, the generalized normal logic program $\mathcal{P}^*$ is constructed as follows: (*i*) set $\mathcal{P}^*$ to the set of all ground instantiations of rules in $\mathcal{P}$; (*ii*) if an atom A is not head of any rule in $\mathcal{P}^*$, then add the rule $A \leftarrow 0$ to $\mathcal{P}^*$ (it is a standard practice in logic programming to consider such atoms as *false*); (*iii*) replace several rules in $\mathcal{P}^*$ having same head, $A \leftarrow \varphi_1$, $A \leftarrow \varphi_2$, ... with $A \leftarrow \varphi_1 \vee \varphi_2 \vee \ldots$ (recall that $\vee$ is the join operator of the truth lattice in infix notation). Note that in $\mathcal{P}^*$, each atom appears in the head of *exactly one* rule.

Semantics. An *interpretation* I *of a logic program* is a mapping from atoms to members of $[0,1]_{\mathbb{Q}}$. I is extended from atoms to the interpretation of rule bodies as follows: $I(f(B_1, \ldots, B_n)) = f(I(B_1), \ldots, I(B_n))$. The ordering $\leq$ is extended from $[0,1]_{\mathbb{Q}}$ to the set of all interpretations point-wise: (*i*) $I_1 \leq I_2$ iff $I_1(A) \leq I_2(A)$, for every ground atom A. With $\mathtt{I}_\perp$ we denote the bottom interpretation under $\leq$ (it maps any atom into 0).

An interpretation I is a *model* of a logic program $\mathcal{P}$, denoted by $I \models \mathcal{P}$, iff for all $A \leftarrow \varphi \in \mathcal{P}^*$, $I(\varphi) \leq I(A)$ holds. The semantics of a logic program $\mathcal{P}$ is determined by the least

[1]Due to lack of space, we do not deal with non-monotonic negation here, though we can managed is as in [39].

model of $\mathcal{P}$, $M_{\mathcal{P}} = \min\{I : I \models \mathcal{P}\}$. The *existence and uniqueness* of $M_{\mathcal{P}}$ is guaranteed by the fixed-point characterization, by means of the *immediate consequence operator* $\Phi_{\mathcal{P}}$. For an interpretation I, for any ground atom A, $\Phi_{\mathcal{P}}(I)(A) = I(\varphi)$, where $A \leftarrow \varphi \in \mathcal{P}^*$. We can show that the function $\Phi_{\mathcal{P}}$ is monotone, the set of fixed-points of $\Phi_{\mathcal{P}}$ is a complete lattice and, thus, $\Phi_{\mathcal{P}}$ has a least fixed-point and I is a model of a program $\mathcal{P}$ iff I is a fixed-point of $\Phi_{\mathcal{P}}$. Therefore, the minimal model of $\mathcal{P}$ coincides with the least fixed-point of $\Phi_{\mathcal{P}}$, which can be computed in the usual way by iterating $\Phi_{\mathcal{P}}$ over $\mathtt{I}_{\perp}$ [39, 40].

Example 3 ([44]) *In [44],* Fuzzy Logic Programming *is proposed, where rules have the form $A \leftarrow f(A_1, ..., A_n)$ for some specific f. [44] is just a special case of our framework. As an illustrative example consider the following scenario. Assume that we have the following facts, represented in the tables below. There are hotels and conferences, their locations and the distance among locations.*

HasLocationH	
HotelID	HasLocationH
h1	hl1
h2	hl2
⋮	⋮

HasLocationC	
ConferenceID	HasLocationC
c1	cl1
c2	cl2
⋮	⋮

Distance		
HasLocationH	HasLocationC	Distance
hl1	cl1	300
hl1	cl2	500
hl2	cl1	750
hl2	cl2	750
⋮	⋮	

Now, suppose that our query is to find hotels close to the conference venue, labeled c1. *We may formulate our query as the rule:*

$$\begin{array}{lll} \mathtt{Query}(h) & \leftarrow & \min(\\ & & \mathtt{HasLocationH}(h, hl), \\ & & \mathtt{HasLocationC}(\mathtt{c1}, cl), \\ & & \mathtt{Distance}(hl, cl, d), \mathtt{Close}(d)) \end{array}$$

where $\mathtt{Close}(x)$ *is defined as* $\mathtt{Close}(x) = \max(0, 1 - x/1000)$. *As a result to that query we get a ranked list of hotels as shown in the table below.*

Result List	
HotelID	Closeness degree
h1	0.7
h2	0.25
⋮	⋮

3 Fuzzy DLPs

In this section we introduce fuzzy *Description Logic Programs* (fuzzy DLPs), which are a combination of fuzzy DLs with fuzzy LPs. In the classical semantics setting, there are mainly three approaches (see, [12, 15], for an overview), the so-called axiom-based approach (e.g. [20, 26]) and the DL-log approach (e.g., [7, 9, 10]) and the autoepistemic approach (e.g., [6, 11]). We are not going to discuss in this section these approaches. The interested reader may see [42]. We just point out that in this paper we follow the DL-log approach, in which rules may not modify the extension of concepts and DL atoms and roles appearing the body of a rule act as procedural calls to the DL component.

We would like to note that the unique combination of DL and LPs for the management of imprecision we are aware of is [42]. The major problems behind [42] rely on the computational part. Indeed, [42] requires that the so-called annotation terms (see [23]) are grounded, which makes the approach hardly feasible in practice. We do not have here such restrictions.

Syntax. We assume that the description logic component and the rules component share the same alphabet of constants. Rules are as for fuzzy LPs except that now atoms and roles may appear in the rule body. We assume that no rule head atom belongs to the DL signature. For ease the readability, in case of ambiguity, DL predicates will have a DL superscript in the rules. Note that in [9] a concept inclusion may appear in the body of the rule. We will not deal with this feature. A *fuzzy Description Logic Program* (fuzzy DLP) is a tuple $\mathcal{DP} = \langle \mathcal{K}, \mathcal{P} \rangle$, where $\mathcal{K}$ is a fuzzy DL knowledge base and $\mathcal{P}$ is a fuzzy logic program. For instance, the following is a fuzzy DLP:

$$\begin{array}{lll} \mathtt{LowCarPrize}(x) & \leftarrow & \min(\mathtt{made\_by}(x, y), \\ & & \mathtt{ChineseCarCompany}^{DL}(y)), \\ & & \mathtt{has\_prize}(x, z), \mathtt{LowPrize}^{DL}(z) \\ \mathtt{made\_by}(x, y) & \leftarrow & \mathtt{makes}^{DL}(y, x), \\ & & \\ \mathtt{LowPrize} & = & \mathtt{ls}(5.000, 15.000) \\ & & \\ \mathtt{ChineseCarCompany} & = & (\exists \mathtt{has\_location.China}) \sqcap \\ & = & (\exists \mathtt{makes.Car}) \end{array}$$

with meaning: a chinese car company is located in china, makes cars, which are sold as low prize cars. Low prize is defined as a fuzzy concept with left-shoulder membership function.

Semantics. We recall that in the DL-log approach, a DL atom appearing in a rule body acts as a query to the underlying DL knowledge base (see [9]). So, consider a fuzzy DLP $\mathcal{DP} = \langle \mathcal{K}, \mathcal{P} \rangle$. The *Herbrand universe* of $\mathcal{P}$, denoted $H_{\mathcal{P}}$ is the set of constants appearing in $\mathcal{DP}$ (if no such constant symbol exists, $H_{\mathcal{P}} = \{c\}$ for an arbitrary constant symbol c from the alphabet of constants). The *Herbrand base* of $\mathcal{P}$, denoted $\mathcal{B}_{\mathcal{P}}$, is the set of all ground atoms built up from the non-DL predicates and the Herbrand universe of $\mathcal{P}$. Then, the definition of $\mathcal{P}^*$ is as for fuzzy LPs. An *interpretation* I w.r.t. $\mathcal{DP}$ is a function $I\colon B_{\mathcal{P}} \to [0,1]_{\mathbb{Q}}$ mapping non-DL atoms into $[0,1]_{\mathbb{Q}}$. We say that I is a *model* of a $\mathcal{DP} = \langle \mathcal{K}, \mathcal{P} \rangle$ iff $I^{\mathcal{K}} \models_{\mathcal{K}} \mathcal{P}$, where

1. $I^{\mathcal{K}} \models \mathcal{P}$ iff for all $A \leftarrow \varphi \in \mathcal{P}^*$, $I^{\mathcal{K}}(\varphi) \leq I^{\mathcal{K}}(A)$;
2. $I^{\mathcal{K}}(f(A_1, ..., A_n)) = f(I^{\mathcal{K}}(A_1), ..., I^{\mathcal{K}}(A_n)$;
3. $I^{\mathcal{K}}(P(t_1, \ldots, t_n)) = I(P(t_1, \ldots, t_n))$ for all ground non-DL atoms $P(t_1, \ldots, t_n)$;
4. $I^{\mathcal{K}}(A(a)) = glb(\mathcal{K}, a\colon A)$ for all ground DL atoms $A(a)$;
5. $I^{\mathcal{K}}(R(a,b)) = glb(\mathcal{K}, (a,b)\colon R)$ for all ground DL roles $R(a,b)$.

Note how in Points 4. and 5. the interpretation of a DL-atom and role depends on the DL-component only. Finally, we say that $\mathcal{DP} = \langle \mathcal{K}, \mathcal{P} \rangle$ *entails* a ground atom A, denoted $\mathcal{DP} \models A$, iff $I \models A$ whenever $I \models \mathcal{DP}$.

For instance, assume that together with the $\mathcal{DP}$ about low prize cars we have the following instances, where `l1` and `l2` are located in China and `car1` and `car2` are cars.

CarCompany	
CarCompany	has_location
c1	l1
c2	l2
⋮	⋮

Makes	
CarCompany	makes
c1	car1
c2	car2
⋮	⋮

If the prizes are as in the left table below then the degree of the car prizes is depicted in the right table below. Note that due to the definition of chinese car companies, `c1` and `c2` are chinese car companies.

Prize	
Car	prize
car1	10.000
car2	7.500
⋮	⋮

LowPrizeCar	
Car	LowPrizeDegree
car1	0.5
car2	0.75
⋮	⋮

Interestingly, it is possible to adapt the standard results of Datalog to our case, which say that a satisfiable description logic program $\mathcal{DP}$ has a minimal model $M_{\mathcal{DP}}$ and entailment (logical consequence) can be reduced to model checking in this minimal model.

Proposition 1 *Let $\mathcal{DP} = \langle \mathcal{K}, \mathcal{P} \rangle$ be a fuzzy DLP. If $\mathcal{DP}$ is satisfiable, then there exists a unique model $M_{\mathcal{DP}}$ such that $M_{\mathcal{DP}} \leq I$ for all models I of $\mathcal{DP}$. Furthermore, for any ground atom A, $\mathcal{DP} \models A$ iff $M_{\mathcal{DP}} \models A$.*

The minimal model can be computed as the least fixed-point of the following monotone operator. Let $\mathcal{DP} = \langle \mathcal{K}, \mathcal{P} \rangle$ be a fuzzy DLP. Define the operator $T_{\mathcal{DP}}$ on interpretations as follows: for every interpretation I, for all ground atoms $A \in B_{\mathcal{P}}$, given $A \leftarrow \varphi \in \mathcal{P}^*$

$$T_{\mathcal{DP}}(I)(A) = I^{\mathcal{K}}(\varphi) \ .$$

Then it can easily be shown that $T_{\mathcal{DP}}$ is monotone, i.e. $I \leq I'$ implies $T_{\mathcal{DP}}(I) \leq T_{\mathcal{DP}}(I')$, and, thus, by the Knaster-Tarski Theorem $T_{\mathcal{DP}}$ has a least fixed-point, which can be computed as a fixed-point iteration of $T_{\mathcal{DP}}$ starting with $\mathtt{I}_{\perp}$.

Reasoning. From a reasoning point of view, to solve the entailment problem we proceed as follows. Given $\mathcal{DP} = \langle \mathcal{K}, \mathcal{P} \rangle$, we first compute for all DL atoms $A(a)$ occurring in $\mathcal{P}^*$, the greatest truth lower bound, i.e. $n_{A(a)} = glb(\mathcal{K}, a\colon A)$. Then we add the rule $A(a) \leftarrow n_{A(a)}$ to $\mathcal{P}$, establishing that the truth degree of $A(a)$ is at least $n_{A(a)}$ (similarly for roles). Finally, we can rely on a theorem prover for fuzzy LPs only either using a usual bottom-up computation or a top-down computation for logic programs [3, 39, 40, 44]. Of course, one has to be sure that both computations, for the fuzzy DL component and for the fuzzy LP component, are supported. With respect to the logic presented in this paper, we need the reasoning algorithm described in [38] for fuzzy DLs component [2], while we have to use [39, 40]

[2]However, sub-concept specification in terminological axioms are of the form $A \sqsubseteq C$ only, where A is a concept name and neither cyclic definitions are allowed nor may there be more than one definition per concept name A.

for the fuzzy LP component.

We conclude by mentioning that by relying on [39], the whole framework extends to fuzzy description normal logic programs as well (non-monotone negation is allowed in the logic programming component).

4 Conclusions

We integrated the management of imprecision into a highly expressive family of representation languages, called fuzzy Description Logic Programs, resulting from the combination of fuzzy Description Logics and fuzzy Logic Programs. We defined syntax, semantics, declarative and fixed-point semantics of fuzzy DLPs. We also detailed how query answering can be performed by relying on the combination of currently known algorithms, without any significant additional effort.

Our motivation is inspired by its application in the Semantic Web, in which both aspects of structured and rule-based representation of knowledge are becoming of interest [16, 19].

There are some appealing research directions. At first, it would certainly be of interest to investigate about reasoning algorithm for fuzzy description logic programs under the so-called axiomatic approach. Currently, very few is known about that. Secondly, while there is a huge literature about fuzzy logic programming and many-valued programming in general, very little is known in comparison about fuzzy DLs. This area may deserve more attention.

References

[1] 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.

[2] T. H. Cao. Annotated fuzzy logic programs. *Fuzzy Sets and Systems*, 113(2):277–298, 2000.

[3] C. Viegas Damásio, J. Medina, and M. Ojeda Aciego. A tabulation proof procedure for residuated logic programming. In *Proc. of the 6th European Conf. on Artificial Intelligence (ECAI-04)*, 2004.

[4] C. Viegas Damásio, J. Medina, and M. Ojeda Aciego. Termination results for sorted multi-adjoint logic programs. In *Proc. of the 10th Int. Conf. on Information Processing and Managment of Uncertainty in Knowledge-Based Systems, (IPMU-04)*, pages 1879–1886, 2004.

[5] C. Viegas Damásio and L. Moniz Pereira. Antitonic logic programs. In *Proc. of the 6th European Conf. on Logic Programming and Nonmonotonic Reasoning (LPNMR-01)*, number 2173 in Lecture Notes in Computer Science. Springer-Verlag, 2001.

[6] F. M. Donini, M. Lenzerini, D. Nardi, W. Nutt, and A. Schaerf. An epistemic operator for description logics. *Artificial Intelligence*, 100(1-2):225–274, 1998.

[7] F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. AL-log: Integrating datalog and description logics. *Journal of Intelligent Information Systems*, 10(3):227–252, 1998.

[8] R. Ebrahim. Fuzzy logic programming. *Fuzzy Sets and Systems*, 117(2):215–230, 2001.

[9] T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Combining answer set programming with description logics for the semantic web. In *Proc. of the 9th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR-04)*. AAAI Press, 2004.

[10] T. Eiter, T. Lukasiewicz, R. Schindlauer, and H. Tompits. Well-founded semantics for description logic programs in the semantic web. In *Proc. RuleML 2004 Workshop, Int. Semantic Web Conference*, LNCS 3323, pages 81–97. Springer Verlag, 2004.

[11] A. Lopatenko E. Franconi, G. Kuper and L. Serafini. A robust logical and computational characterisation of peer-to-peer database systems. In *Proc. of the VLDB Int. Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P-03)*, 2004.

[12] Pan *et al.* Specification of coordination of rule and ontology languages. Technical report, Knowledgeweb Network of Excellence, EU-IST-2004-507482, 2004. Deliverable D2.5.1.

[13] M. C. Fitting. Fixpoint semantics for logic programming - a survey. *Theoretical Computer Science*, 21(3):25–51, 2002.

[14] M. Fitting. A Kripke-Kleene-semantics for general logic programs. *Journal of Logic Programming*, 2:295–312, 1985.

[15] E. Franconi and S. Tessaris. Rules and queries with ontologies: a unified logical framework. In *Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR-04)*, 2004.

[16] B. N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: combining logic programs with description logic. In *Proc. of the 12th Int. Conf. on World Wide Web*, pages 48–57. ACM Press, 2003.

[17] P. Hájek. *Metamathematics of Fuzzy Logic*. Kluwer, 1998.

[18] I. Horrocks and P. Patel-Schneider. Reducing OWL entailment to description logic satisfiability. *Journal of Web Semantics*, 2004.

[19] I. Horrocks and P. F. Patel-Schneider. Three theses of representation in the semantic web. In *Proc. of the 12th Int. Conf. on World Wide Web*, pages 39–47. ACM Press, 2003.

[20] I. Horrocks and P. F. Patel-Schneider. A proposal for an OWL rules language. In *Proc. of the 13th Int. World Wide Web Conf. (WWW-04)*. ACM, 2004.

[21] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[22] M. Ishizuka and N. Kanai. Prolog-ELF: incorporating fuzzy logic. In *Proc. of the 9th Int. Joint Conf. on Artificial Intelligence (IJCAI-85)*, pages 701–703, Los Angeles, CA, 1985.

[23] M. Kifer and V.S. Subrahmanian. Theory of generalized annotated logic programming and its applications. *Journal of Logic Programming*, 12:335–367, 1992.

[24] S. Krajči, R. Lencses, and P. Vojtáš. A comparison of fuzzy and annotated logic programming. *Fuzzy Sets and Systems*, 144:173–192, 2004.

[25] Laks V.S. Lakshmanan and N. Shiri. A parametric approach to deductive databases with uncertainty. *IEEE Transactions on Knowledge and Data Engineering*, 13(4):554–570, 2001.

[26] A. Y. Levy and M.-C. Rousset. Combining horn rules and description logics in CARIN. *Artificial Intelligence*, 104:165–209, 1998.

[27] Y. Loyer and U. Straccia. The approximate well-founded semantics for logic programs with uncertainty. In *28th Int. Symp. on Mathematical Foundations of Computer Science (MFCS-2003)*, LNCS 2747, pages 541–550, 2003. Springer-Verlag.

[28] Y. Loyer and U. Straccia. Default knowledge in logic programs with uncertainty. In *Proc. of the 19th Int. Conf. on Logic Programming (ICLP-03)*, LNCS 2916, pages 466–480, 2003. Springer Verlag.

[29] Y. Loyer and U. Straccia. Epistemic foundation of the well-founded semantics over bilattices. In *29th Int. Symp. on Mathematical Foundations of Computer Science (MFCS-2004)*, LNCS 3153, pages 513–524, 2004. Springer Verlag.

[30] Y. Loyer and U. Straccia. Any-world assumptions in logic programming. *Theoretical Computer Science*, 342(2-3):351–381, 2005.

[31] C. Mateis. Extending disjunctive logic programming by t-norms. In *Proc. of the 5th Int. Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-99)*, LNCS 1730, pages 290–304. Springer-Verlag, 1999.

[32] C. Mateis. Quantitative disjunctive logic programming: Semantics and computation. *AI Communications*, 13:225–248, 2000.

[33] J. Medina, M. Ojeda-Aciego, and P. Vojtáš. Multi-adjoint logic programming with continuous semantics. In *Proc. of the 6th Int. Conf. on Logic Programming and Nonmonotonic Reasoning (LPNMR-01)*, LNAI 2173, pages 351–364. Springer Verlag, 2001.

[34] J. Medina, M. Ojeda-Aciego, and P. Vojtáš. A procedural semantics for multi-adjoint logic programming. In *Proc. of the10th Portuguese Conf. on Artificial Intelligence on Progress in Artificial Intelligence, Knowledge Extraction, Multi-agent Systems, Logic Programming and Constraint Solving*, pages 290–297. Springer-Verlag, 2001.

[35] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.

[36] E. Y. Shapiro. Logic programs with uncertainties: A tool for implementing rule-based systems. In *Proc. of the 8th Int. Joint Conf. on Artificial Intelligence (IJCAI-83)*, pages 529–532, 1983.

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

[38] U. Straccia. Description logics with fuzzy concrete domains. *21st Conf. on Uncertainty in Artificial Intelligence (UAI-05)*, pages 559–567, Edinburgh, Scotland, 2005. AUAI Press.

[39] U. Straccia. Query answering in normal logic programs under uncertainty. In *8th European Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU-05)*, LNCS 3571, pages 687–700, 2005. Springer Verlag.

[40] U. Straccia. Uncertainty management in logic programming: Simple and effective top-down query answering. *9th Int. Conf. on Knowledge-Based & Intelligent Information & Engineering Systems (KES-05), Part II*, LNCS 3682, pages 753–760, 2005. Springer Verlag.

[41] U. Straccia. A fuzzy description logic for the semantic web. In Elie Sanchez, editor, *Capturing Intelligence: Fuzzy Logic and the Semantic Web*. Elsevier, 2006.

[42] U. Straccia. Uncertainty and description logic programs over lattices. In Elie Sanchez, editor, *Capturing Intelligence: Fuzzy Logic and the Semantic Web*. Elsevier, 2006.

[43] M.H. van Emden. Quantitative deduction and its fixpoint theory. *Journal of Logic Programming*, 4(1):37–53, 1986.

[44] P. Vojtáš. Fuzzy logic programming. *Fuzzy Sets and Systems*, 124:361–370, 2004.

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