Fuzzy $\mathcal{ALC}$ with Fuzzy Concrete Domains

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

Abstract

We present a fuzzy description logic where the representation of concept membership functions and fuzzy modifiers is allowed, together with a inference procedure based on a mixture of a tableaux and bounded mixed integer programming.

1 Introduction

Description Logics (DLs) [1] play an important role in the context of the *Semantic Web* as they are essentially the theoretical counterpart of the *Web Ontology Language OWL DL* [5], a state of the art language to specify ontologies. However, DLs becomes less suitable in domains where concepts have not a precise definition. For instance, in a flower ontology we may encounter the problem of representing concepts like "Candia is a creamy white rose with dark pink edges to the petals" and "Calla is a very large, long white flower on thick stalks". As it becomes apparent such concepts hardly can be encoded into DLs, as they involve so-called *fuzzy* or *vague concepts*, like "creamy", "dark", "large" and "thick". The problem to deal with *imprecision* has been addressed several decades ago by Zadeh, which gave bird in the meanwhile to the so-called *fuzzy set and fuzzy logic theory* (see, e.g. [4] for an in-depth study of fuzzy logic). Unfortunately, despite the popularity of fuzzy logic theory, relative little work has been carried out involving fuzzy DLs [3, 7, 8, 11, 12].

We present a fuzzy version of $\mathcal{ALC}(\mathtt{D})$. Main features are that we allow the explicit representation of typical concept membership functions (fuzzy concrete domains) and fuzzy modifiers (similarly to [11, 3]). We present a novel inference procedure based on a mixture of tableaux rules and bounded Mixed Integer Programming (bMIP). In the following, we present fuzzy $\mathcal{ALC}(\mathtt{D})$ and a reasoning procedure. An extended version of this work can be found in [9].

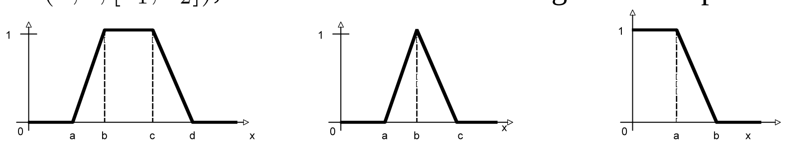
2 Fuzzy $\mathcal{ALC}$ with fuzzy domains

A *fuzzy set* A w.r.t. a universe X is characterized by a *membership function* $\mu_A \colon X \to [0, 1]$, or simply $A(x) \in [0, 1]$. $A(x)$ gives us an estimation of the belonging of x

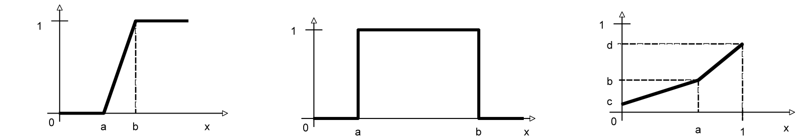
to A. In fuzzy logics, the degree of membership $A(x)$ is regarded as the *degree of truth* of the statement "*x is* A". Accordingly, in our fuzzy DL, a concept C will be interpreted as a fuzzy set and, thus, concepts become *imprecise*; and, consequently, e.g. the statement "a is an instance of concept C", will have a truth-value in $[0, 1]$ given by the membership degree $C(a)$. In our fuzzy variant of $\mathcal{ALC}(\mathtt{D})$, unlike the classical case (see [6]), concrete domains are considered as fuzzy sets. A *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 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 $[0, 1] \cap \mathbb{Q}$ (in the following, whenever we write $[0, 1]$, we mean $[0, 1] \cap \mathbb{Q}$). For instance, we may define the predicate $\leq_{18}$ as an unary predicate over the natural numbers denoting the set of integers smaller or equal to 18. On the other hand, Young may be a fuzzy predicate denoting the degree of youngness of a person's age over the domain range $[0, 150]$ with

$$\mathtt{Young}(x) = \mathtt{ls}(10, 30, [0, 150]) ,$$

where $ls(a, b, [0, 150])$ is a left shoulder function with shape defined as in the figure below. Concerning fuzzy predicates, there are many membership functions for fuzzy sets membership specification. However, (see figure below), 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 function* $ls(a, b, [k_1, k_2])$, the *right-shoulder function* $rs(a, b, [k_1, k_2])$ 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.



Trapezoidal function   Triangular function   Left-shoulder function



Right-shoulder function   Crisp function   Linear modifier function

Fuzzy modifiers [3, 11] like very, more_or_less and slightly, apply to fuzzy sets to change their membership function and allow, e.g. to express concepts like very(High), moreOrLess(Ripe) and slightly(Nice). 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 slightly(x) as $\mathtt{lm}(0.7, 0.49, 1, 0)$, where $lm(a, b, c, d)$ is the *linear modifier* in the figure. For the purpose of this paper, we will assume that modifiers are a *linear combination* of two linear functions as depicted in the figure, which covers usual cases.

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 individual names* (denoted a), *concrete individual 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 following syntactic rules (d is a unary fuzzy predicate):

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

A *TBox* $\mathcal{T}$ consists of a finite set of *terminological axioms* of the form $A \sqsubseteq C$ (A is sub-concept of C) or $A = C$ (A is defined as the concept C), where A is a concept name and C is concept. We allow the definition of modifier names and concrete predicates names to appear in the TBox and concept expressions. For instance, $\mathtt{very} = \mathtt{lm}(0.7, 0.49, 0, 1) \in \mathcal{T}$ dictates that $\mathtt{very}$ is an abbreviation for $\mathtt{lm}(0.7, 0.49, 0, 1)$, while $\mathtt{Young} = \mathtt{ls}(10, 30, [0, 150]) \in \mathcal{T}$ dictates that $\mathtt{Young}$ is an abbreviation for $\mathtt{ls}(10, 30, [0, 150])$.

We also assume that no concept A appears more than once on the left hand side of a terminological axiom and that no cyclic definitions are present in $\mathcal{T}$. Note that in classical DLs, usually terminological axioms are of the form $C \sqsubseteq D$, where C and D are concepts. While from a semantics point of view it is easy to consider them as well (see [10]), we have not yet found a calculus to deal with such axioms.

Using axioms we may define the concept of a minor and young person as

$$\begin{array}{l} \mathtt{Minor} = \mathtt{Person} \sqcap \exists \mathtt{age}.\leq_{18} \\ \leq_{18} = \mathtt{cr}(0, 18, [0, 150]) \\ \\ \mathtt{YoungPerson} = \mathtt{Person} \sqcap \exists \mathtt{age}.\mathtt{Young} \\ \mathtt{Young} = \mathtt{ls}(10, 30, [0, 150]) \end{array}$$

A *concept-*, *role- assertion axiom* and an *individual (in)equality axiom* has the form $a{:}\,C, (a, b){:}\,R, a \approx b$ and $a \not\approx b$, respectively, where a, b are abstract individuals. For $n \in [0, 1]$, an *ABox* $\mathcal{A}$ is a finite set of *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. An ABox $\mathcal{A}$ may also contain a finite set of individual (in)equality axioms $a \approx b$ and $a \not\approx b$, respectively. 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 $\mathcal{A}$.

From a semantics point of view, we extend fuzzy $\mathcal{ALC}$ [7]. A *fuzzy interpretation* $\mathcal{I}$ w.r.t. 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 $C^{\mathcal{I}}{:}\, \Delta^{\mathcal{I}} \rightarrow [0, 1]$; (ii) to each abstract role $R \in \mathtt{R}_a$ a function $R^{\mathcal{I}}{:}\, \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0, 1]$; (iii) to each abstract feature $r \in \mathtt{F}_a$ a partial function $r^{\mathcal{I}}{:}\, \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0, 1]$ 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 individual $a \in \mathtt{I}_a$

an element in $\Delta^{\mathcal{I}}$; (v) to each concrete individual $c \in \mathtt{I}_c$ 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 corresponding function $f_m\colon [0,1] \to [0,1]$; (ix) to each unary concrete predicate d the corresponding fuzzy relation $d^{\mathtt{D}}\colon \Delta_{\mathtt{D}} \to [0,1]$ and to $\neg d$ the negation of $d^{\mathtt{D}}$. The mapping $\cdot^{\mathcal{I}}$ is extended to concepts and roles as follows (where $u \in \Delta^{\mathcal{I}}$): $\top^{\mathcal{I}}(u) = 1$, $\bot^{\mathcal{I}}(u) = 0$,

$$
\begin{array}{lcl}
(C_1 \sqcap C_2)^{\mathcal{I}}(u) & = & \min\{{C_1}^{\mathcal{I}}(u), {C_2}^{\mathcal{I}}(u)\} \\
(C_1 \sqcup C_2)^{\mathcal{I}}(u) & = & \max\{{C_1}^{\mathcal{I}}(u), {C_2}^{\mathcal{I}}(u)\} \\
(\neg C)^{\mathcal{I}}(u) & = & 1 - C^{\mathcal{I}}(u) \\
(m(C))^{\mathcal{I}}(u) & = & f_m(C^{\mathcal{I}}(u)) \\
(\forall R.C)^{\mathcal{I}}(u) & = & \inf_{w \in \Delta^{\mathcal{I}}} \max\{1 - R^{\mathcal{I}}(u,w), C^{\mathcal{I}}(w)\} \\
(\exists R.C)^{\mathcal{I}}(u) & = & \sup_{w \in \Delta^{\mathcal{I}}} \min\{R^{\mathcal{I}}(u,w), C^{\mathcal{I}}(w)\} \\
(\forall T.D)^{\mathcal{I}}(u) & = & \inf_{o \in \Delta_{\mathtt{D}}} \max\{1 - T^{\mathcal{I}}(u,o), D^{\mathcal{I}}(o)\} \\
(\exists T.D)^{\mathcal{I}}(u) & = & \sup_{o \in \Delta_{\mathtt{D}}} \min\{T^{\mathcal{I}}(u,o), D^{\mathcal{I}}(o)\}\ .
\end{array}
$$

Note that due to the restrictions on the chosen fuzzy functions, we do have that $(\forall R.C)^{\mathcal{I}} = (\neg\exists R.\neg C)^{\mathcal{I}}$. This will allow us to transform concept expressions into a semantically equivalent *Negation Normal Form* (NNF), *which is obtained by pushing in the usual manner negation on front of concept names, modifiers and concrete predicate names only*. With $\mathtt{nnf}(C)$ we denote the NNF of concept C. 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:

$$
\begin{array}{lll}
I \models A \sqsubseteq C & \text{iff} & \text{for all } u \in \Delta^{\mathcal{I}}, A^{\mathcal{I}}(u) \leq C^{\mathcal{I}}(u) \\
I \models A = C & \text{iff} & \text{for all } u \in \Delta^{\mathcal{I}}, A^{\mathcal{I}}(u) = C^{\mathcal{I}}(u) \\
I \models \langle \alpha, n \rangle & \text{iff} & \alpha^{\mathcal{I}} \geq n \\
\mathcal{I} \models a \approx b & \text{iff} & a^{\mathcal{I}} = b^{\mathcal{I}} \\
\mathcal{I} \models a \not\approx b & \text{iff} & a^{\mathcal{I}} \neq b^{\mathcal{I}}
\end{array}
$$

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 write $\mathcal{K} \models \langle A \sqsubseteq B, n \rangle$ iff for every model $\mathcal{I}$ of $\mathcal{K}$, $[\inf_{u \in \Delta^{\mathcal{I}}} A^{\mathcal{I}}(u) \Rightarrow B^{\mathcal{I}}(u)] \geq n$. Finally, given $\mathcal{K}$ and an axiom α 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. As $\mathcal{K} \models \langle \alpha, n \rangle$ iff $glb(\mathcal{K}, \alpha) \geq n$, the BDB problem is the major problem we have to consider in fuzzy $\mathcal{ALC}(\mathtt{D})$, which we address in the next section.

Example 1 *Consider the following simplified excerpt of a knowledge base about cars (*`speed` *is a concrete feature):*

$$
\begin{array}{l}
\mathtt{SportsCar} = \exists \mathtt{speed}.\mathtt{very}(\mathtt{High}),\quad \mathtt{High} = \mathtt{rs}(80, 250, [0, 400]) \\
\mathtt{very} = \mathtt{lm}(0.7, 0.49, 0, 1),\quad \leq_{170} = \mathtt{cr}(0, 170, [0, 400]) \\
\geq_{350} = \mathtt{cr}(350, 400, [0, 400]),\quad =_{243} = \mathtt{cr}(243, 243, [0, 400])
\end{array}
$$

$$
\langle \mathtt{mg\_mgb}\colon \exists \mathtt{speed}.\leq_{170}, 1 \rangle,\ \langle \mathtt{ferrari\_enzo}\colon \exists \mathtt{speed}.\geq_{350}, 1 \rangle,\ \langle \mathtt{audi\_tt}\colon \exists \mathtt{speed}. =_{243}, 1 \rangle\}
$$

Then

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

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* $\mathtt{Minor} = \mathtt{Person} \sqcap \exists \mathtt{age}.\leq_{18}$ *and* $\mathtt{YoungPerson} = \mathtt{Person} \sqcap \exists \mathtt{age}.\mathtt{Young}$*, where* $\leq_{18} = \mathtt{cr}(0, 18, [0, 150])$*,* $\mathtt{Young} = \mathtt{ls}(10, 30, [0, 150])$ *and* age *is a concrete feature. Then* $\mathcal{K} \models \langle \mathtt{Minor} \sqsubseteq \mathtt{YoungPerson}, 0.5\rangle$*.*

Example 3 *Consider the following simplified excerpt of a computer store:*

$$\mathtt{Computer} \sqsubseteq \exists \mathtt{hasPrice}.\mathtt{Price}),\ \langle \mathtt{c1}\colon \exists \mathtt{hasPrice}.=_{995}, 1\rangle,\ \langle \mathtt{c2}\colon \exists \mathtt{hasPrice}.=_{1010}, 1\rangle$$

where hasPrice *is a concrete feature. Suppose a customer is looking for a computer whose price is in the range* $[900, 1000]$*. In a classical DL retrieval system just* c1 *is retrieved, leaving unfortunately* c2 *out. However, if the system internally defines*

$$\mathtt{PriceRange} = \mathtt{trz}(800, 900, 1000, 1100, [0, 5000])$$

establishing that the price boundaries of the customer are not crisp anymore (as almost all store owners do), then

$$\begin{array}{lcl} \mathcal{K} & \models & \langle \mathtt{c1}\colon \exists \mathtt{hasPrice}.\mathtt{PriceRange}, 1\rangle \\ \mathcal{K} & \models & \langle \mathtt{c2}\colon \exists \mathtt{hasPrice}.\mathtt{PriceRange}, 0.9\rangle \end{array}$$

and, thus, the "fuzzy" retrieval gives us a ranking of the items c1 *and* c2*, in decreasing order of relevance.*

3 Reasoning

A more detailed description of the reasoning algorithm can be found in [9]. Consider $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$. In order to solve the BDB problem, we combine appropriate DL completion rules with methods developed in the context of *Many-Valued Logics* (MVLs) [2]. The basic idea is as follows. In order to determine e.g. $glb(\mathcal{K}, a\colon C)$, we consider an expression of the form $\langle a\colon \neg C, 1 - x\rangle$ (informally, $\langle a\colon C \leq x\rangle$), where x is a $[0, 1]$-valued variable. Then we construct a tableaux for $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \cup \{\langle a\colon \neg C, 1 - x\rangle\}\rangle$ in which the application of satisfiability preserving rules generates new assertion axioms together with *inequations* over $[0, 1]$-valued variables. These inequations have to be hold in order to respect the semantics of the DL constructors. Finally, in order to determine the greatest lower bound, we *minimize* the original variable x such that all constraints are satisfied. In this paper, we limited the choice of the semantics of concept constructors, modifiers and fuzzy predicates in such a way that we end up with a *bounded Mixed Integer Program* (bMIP) optimization problem. Interestingly, as for the MVL case, the tableaux we are generating contains *one* branch only and, thus, just *one* bMIP problem has to be solved.

A general MIP problem consists in minimizing a linear function w.r.t. a set of constraints that are linear inequations in which rational and integer variables can occur. In our case, the variables are bounded. More precisely, let $\mathbf{x} = \langle x_1, \ldots, x_k \rangle$ and $\mathbf{y} = \langle y_1, \ldots, y_m \rangle$ be variables over $\mathbb{Q}$, over the integers and let A, B be integer matrices and h an integer vector. The variables in $\mathbf{y}$ are called *control variables*. Let $f(\mathbf{x}, \mathbf{y})$ be an $k+m$-ary linear function. Then the *general MIP problem* is to find $\bar{\mathbf{x}} \in \mathbb{Q}^k, \bar{\mathbf{y}} \in \mathbb{Z}^m$ such that $f(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = \min\{f(\mathbf{x}, \mathbf{y})\colon A\mathbf{x} + B\mathbf{y} \geq h\}$. The general case can be restricted to what concerns the paper as we can deal with *bounded* MIP (bMIP). That is, the rational variables range over a given interval, while the integer variables ranges over $\{0, 1\}$. Furthermore, we say that $M \subseteq [0, 1]^k$ is *bMIP-representable* iff there is a bMIP (A, B, h) with k real and m 0-1 variables such that $M = \{\mathbf{x}\colon \exists \mathbf{y} \in \{0, 1\}^m$ such that $A\mathbf{x} + B\mathbf{y} \geq h\}$. In particular, we require that 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 bMIP-representable. It is easily verified that all fuzzy predicates, modifiers and DL constructors are bMIP representable (see [9]). **The BDB problem.** We start with a pre-processing steps in which each $A \sqsubseteq C$ can be replaced with $A = C \sqcap A^*$, where A^* is a new concept name, then substitute concept names with their definitions and finally transform each concept into NNF. This last operations does not affect the semantics due to the restrictions we made on the fuzzy constructors. Notice that negation may appear on front of modifiers in the from $\neg m(C)$, where C is a complex concept. Now, let $\mathtt{V}$ be a new alphabet of variables x ranging over $[0, 1]$, $\mathtt{W}$ be a new alphabet of 0-1 variables y. We extend fuzzy assertions to the form $\langle \alpha, l\rangle$, where l is a linear expression over variables in $\mathtt{V}, \mathtt{W}$ and real values. A *linear constraint* is of the form $l \geq l'$ or $l \leq l'$, where l, l' are linear expressions over variables in $\mathtt{V}, \mathtt{W}$ and rational values. The satisfiability notion of linear constraints is immediate. A *constraint set* S is a set of terminological axioms, fuzzy assertion axioms, (in)equality axioms and linear constraints. $\mathcal{I}$ *satisfies* S iff $\mathcal{I}$ satisfies all elements of it. With S_0 we denote the constraint set $S_0 = \mathcal{T} \cup \mathcal{A}$. We will see later how to determine the satisfiability of a constraint set. In the following, we assume that S_0 is satisfiable, otherwise $glb(\mathcal{K}, \alpha) = 1$. As in [7], concerning fuzzy role assertions, we have that $\mathcal{K} \models \langle (a, b)\colon R, n\rangle$ iff $\langle (a, b)\colon R, m\rangle \in \mathcal{A}$ with $m \geq n$. Therefore, $glb(\mathcal{K}, (a, b)\colon R)) = \max\{n\colon \langle R(a, b), n\rangle \in \mathcal{A}\}$. So we do not consider this case further. Now, let us determine $glb(\mathcal{K}, a\colon C)$. As anticipated,

$$glb(\mathcal{K}, a\colon C) = \min_x \text{ such that } S = S_0 \cup \{\langle a\colon \neg C, 1 - x\rangle\} \text{ satisfiable} \ .$$

Similarly, for a terminological axiom $A \sqsubseteq B$,

$$glb(\mathcal{K}, A \sqsubseteq B) = \min_x \text{ such that } S = S_0 \cup \{\langle a\colon A \sqcap \neg B, 1 - x\rangle\}\} \text{ satisfiable} \ ,$$

where a is new abstract individual. Therefore, the BDB problem can be reduced to minimal satisfiability problem. **The Satisfiability problem.** Our satisfiability checking calculus is based on a set of constraint propagation rules transforming a set S of constraints into "simpler" satisfiability preserving constraint sets S_i until either S_i contains a *clash* or no rule can be further be applied to S_i. If S_i contains a clash then S_i and, thus S is immediately not satisfiable. Otherwise, we apply a bMIP oracle to solve the set of linear constraints in S_i to determine either the satisfiability of the set or the minimal value for a given variable x, making S_i satisfiable. We assume that a constraint set S is reflexive, symmetric and transitively closed concerning the equality axioms. S contains a *clash* iff either $\langle a\colon \bot, n\rangle \in S$ with $n > 0$, or $\{a \approx b,\, a \not\approx b\} \subseteq S$. The rules follow easily from the bMIP representations. *Each rule instantiation is applied at most once*. Before we can formulate the rules we need a technical definition involving feature roles (see [6]). Let S be a constraint set, r an abstract feature and both $\langle (a, b_1)\colon r, l_1\rangle$ and $\langle (a, b_2)\colon r, l_2\rangle$ occur in S. Then we call such a pair a *fork*. As r is a function, such a fork means that b_1 and b_2 have to be interpreted as the same individual. A fork $\langle (a, b_1)\colon r, l_1\rangle$, $\langle (a, b_2)\colon r, l_2\rangle$ can be deleted by replacing all occurrences of b_2 in S by b_1. A similar argument applies to concrete feature roles. At the beginning, we remove the forks from S_0. We assume that forks are eliminated as soon as they appear (as part of a rule application) with the proviso that newly generated individuals are replaced by older ones and not vice-versa. With x_α we denote the variable associated to the *atomic assertion* α of the form $a\colon A$ or $(a, b)\colon R$. x_α will take the truth value associated to α, while with x_c we denote the variable associated to the concrete individual c. The rules are the following:

RA. If $\langle \alpha, l\rangle \in S_i$ and α is an atomic assertion of the form $a\colon A$ or $(a, b)\colon R$ then $S_{i+1} = S_i \cup \{x_\alpha \geq l\}$.

R$\bar{A}$. If $\langle a\colon \neg A, l\rangle \in S_i$ then $S_{i+1} = S_i \cup \{x_{a\colon A} \leq 1 - l\}$.

R$\sqcap$. If $\langle a\colon C \sqcap D, l\rangle \in S_i$ then $S_{i+1} = S_i \cup \{\langle a\colon C, l\rangle, \langle a\colon D, l\rangle\}$.

R$\sqcup$. If $\langle a\colon C \sqcup D, l\rangle \in S_i$ then $S_{i+1} = S_i \cup \{\langle a\colon C, x_1\rangle, \langle a\colon D, x_2\rangle, x_1 + x_2 = l, x_1 \leq y, x_2 \leq 1 - y, x_i \in [0, 1], y \in \{0, 1\}\}$, where x_i is a new variable, y is a new control variable.

R$\exists$. If $\langle a\colon \exists R.C, l\rangle \in S_i$ then $S_{i+1} = S_i \cup \{\langle (a, b)\colon R, l\rangle, \langle b\colon C, l\rangle\}$, where b is a new abstract individual. The case for concrete roles is similar.

R$\forall$. If $\{\langle a\colon \forall R.C, l_1\rangle, \langle (a, b)\colon R, l_2\rangle\} \subseteq S_i$ then $S_{i+1} = S_i \cup \{\langle a\colon C, x\rangle, x + y \geq l_1, x \leq 1 - y, l_1 + l_2 \leq 2 - y, x \in [0, 1], y \in \{0, 1\}\}$, where x is a new variable and y is anew control variable. The case for concrete roles is similar.

Rm. If $\langle a\colon m(C), l\rangle \in S_i$ then $S_{i+1} = S_i \cup \gamma(a\colon C, l)$, where the set $\gamma(a\colon C, l)$ is obtained from the bMIP representation of $g(m)$ as follows: replace in $g(m)$ all occurrences of x_2 with l. Then resolve for x_1 and replace all occurrences of the form $x_1 \geq l'$ with $\langle a\colon C, l'\rangle$, while replace all occurrences the form $x_1 \leq l'$ with $\langle a\colon \mathtt{nnf}(\neg C), 1 - l'\rangle$.

R$\bar{m}$. The case $\langle a\colon \neg m(C), l\rangle \in S_i$ is similar to rule **R**m, where we use the bMIP representation of $\bar{g}(m)$ in place of $g(m)$

Rd. If $\langle c\colon d, l\rangle \in S_i$ then $S_{i+1} = S_i \cup \gamma(c\colon d, l)$, where the set $\gamma(c\colon d, l)$ is obtained from the bMIP representation of $g(d)$ by replacing all occurrences of x_2 with l and x_1 with x_c.

R$\bar{d}$. The case $\langle c\colon \neg d, l\rangle \in S_i$ is similar to rule **R**d, where we use the bMIP representation of $\bar{g}(d)$ in place of $g(d)$.

Note that *an unique branch* is generated in the tableaux of S_0. Some comments for the **R**$\sqcup$ rule. By reasoning by case, for $y = 0$, we have $x_1 = 0, x_2 \leq 1, x_2 = l$, while for $y = 1$, we have $x_2 = 0, x_1 \leq 1, x_1 = l$. Therefore, the control variable y simulates the two branchings of the disjunction. A similar argument applies to the other rules.

A constraint set S' obtained from rule applications to S is a *completion* of S iff no more rule can be applied to S'. It can be shown that the rules are satisfiability preserving and a completion is obtained after a finite number of rule applications. Furthermore, consider $\mathcal{K}\langle \mathcal{T}, \mathcal{A}\rangle$ and let α be a concept assertion axiom $a\colon C$ or a terminological axiom $A \sqsubseteq B$. Then in finite time we can determine $glb(\mathcal{K}, \alpha)$ as the minimal value of x such that the completion of $S = \mathcal{T} \cup \mathcal{A} \cup \{\langle \alpha', 1 - x\rangle\}$ is satisfiable, where (i) $\alpha' = a\colon \neg C$ if $\alpha = a\colon C$, (ii) $\alpha' = a\colon A \sqcap \neg B$ if $\alpha = A \sqsubseteq B$.

An example of computation can be found in [9].

4 Conclusions, related work and outlook

We have presented a fuzzy DL showing that its representation capabilities go beyond current approaches to fuzzy DLs. We recall that the first work on fuzzy DLs is due to Yen ([12]) who considered a sub-language of $\mathcal{ALC}$, $\mathcal{FL}^-$. It already informally talks about the use of modifiers and fuzzy concrete domains. Tresp ([11]) considered fuzzy $\mathcal{ALC}$ extended with a special form of modifiers. $\min$, $\max$ and $1 - x$ membership functions has been considered and complete reasoning algorithm testing the subsumption relationship has been presented. Similar to our approach, a linear programming oracle is needed. Straccia ([7]) considers fuzzy assertion axioms in fuzzy $\mathcal{ALC}$, concept modifiers are not allowed however. He also introduced the BDB problem and provided a sound and complete reasoning algorithm based on completion rules ([8] provides a translation of fuzzy $\mathcal{ALC}$ into classical $\mathcal{ALC}$). In the same spirit [3] extend Straccia's fuzzy $\mathcal{ALC}$ with slightly more enhanced concept modifiers. A sound and complete reasoning algorithm for the graded subsumption problem is presented. Finally, [10] extends $\mathcal{ALC}(\mathtt{D})$ to OWL DL. However, no reasoning algorithm is given. Future work involves the extension of fuzzy $\mathcal{ALC}(\mathtt{D})$ to $\mathcal{SHOIN}(\mathtt{D})$, the theoretical counterpart of OWL DL. Another direction is in extending fuzzy DLs with *fuzzy quantifiers*, where $\forall$ and $\exists$ are replaced with fuzzy quantifiers like `most`, `some`, `usually` and the like. This allows to define concepts like $\mathtt{TopCustomer} = \mathtt{Customer} \sqcap (\mathtt{Usually})\mathtt{buys}.\mathtt{ExpensiveItem}$, $\mathtt{ExpensiveItem} = \mathtt{Item} \sqcap \exists \mathtt{price}.\mathtt{High}$. Fuzzy quantifiers can be applied to inclusion axioms as well, allowing to express, e.g. $(\mathtt{Most})$ $\mathtt{Bird} \sqsubseteq \mathtt{FlyingObject}$. Here the fuzzy quantifier `Most` replaces the classical universal quantifier $\forall$ assumed in the inclusion axioms expressing that "most birds fly".

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] R. Hähnle. Advanced many-valued logics. In *Handbook of Philosophical Logic, 2nd Edition*, Volume 2. Kluwer, 2001.

[3] S. Hölldobler, H. Störr, and T. D. Khang. The Subsumption Problem of the Fuzzy Description Logic ALC_{FH}. In IPMU-04.

[4] Hájek, Petr. *Metamathematics of Fuzzy Logic*. Kluwer, 1998.

[5] 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.

[6] C. Lutz. Description logics with concrete domains—a survey. In *Advances in Modal Logics Volume 4*. 2003.

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

[8] U. Straccia. Transforming fuzzy description logics into classical description logics. In LNCS 3229, 2004. Springer Verlag.

[9] U. Straccia. Fuzzy Description Logics with Concrete Domains. Technical Report 2005-TR-03, ISTI-CNR, Pisa, Italy, 2005.

[10] Umberto Straccia. Towards a fuzzy description logic for the semantic web (preliminary report). In *European Semantic Web Conference (ESWC-05)*, 2005. To appear.

[11] C. Tresp and R. Molitor. A description logic for vague knowledge. In ECAI-98.

[12] J. Yen. Generalizing term subsumption languages to fuzzy logic. In IJCAI-91.