Abstract

Fuzzy Description Logics (fuzzy DLs) have been proposed as a language to describe structured knowledge with vague concepts. In [23], a solution based on Mixed Integer Linear Programming has been proposed to deal with fuzzy DLs under Łukasiewicz semantics in which typical membership functions, such as triangular and trapezoidal functions, can be explicitly represented in the language.

A major theoretical and computational limitation so far is the inability to deal with General Concept Inclusions (GCIs), which is an important feature of classical DLs. In this paper, we address this issue and develop a calculus for fuzzy DLs with GCIs under various semantics: classical logic, “Zadeh semantics”, and Łukasiewicz logic.

Keywords: Fuzzy Description Logics

1 Introduction

Description Logics (DLs) [1] 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, as the theoretical counterpart of OWL DL (the W3C standard for specifying ontologies, see [12] for details).

Fuzzy DLs [18, 22, 28, 30] extend classical DLs by allowing to deal with fuzzy/vague/imprecise concepts for which a clear and precise definition is not possible. The problem to deal with imprecise concepts has been addressed several decades ago by Zadeh [31], which gave birth in the meanwhile to the so-called fuzzy set and fuzzy logic theory and a huge number of real life applications exists. Despite the popularity of fuzzy set theory, relative little work has been carried out in extending DLs towards the representation of imprecise concepts, notwithstanding DLs can be considered as a quite natural candidate for such an extension [2, 5, 7, 8, 13, 16, 19, 18, 20, 21, 22, 23, 25, 27, 28, 30, 32].

From a semantics point of view, most works rely on the Gödel conjunction and disjunction $(x \land y = \min(x, y), x \lor y = \max(x, y))$, but with Łukasiewicz negation $(\neg x = 1 - x)$, essentially the semantics of fuzzy sets operators proposed by Zadeh [31] (which we call here “Zadeh semantics”), while very few work address the case of a fully Łukasiewicz compliant semantics. Indeed, Hajek [6, 5] proposes a reasoning solution based on a reduction to Łukasiewicz propositional logic Ł, while Straccia [23] proposes a reasoning solution, which is based on a mixture of tableau rules and Mixed Integer Linear Programming (MILP). The latter solution has been proposed to deal with fuzzy DLs with so-called fuzzy concrete domains, i.e. the possibility to represent in fuzzy DLs concepts with explicit membership functions such as triangular, trapezoidal, left-shoulder and right-shoulder functions.

A major theoretical, computational and practical limitation so far of fuzzy DLs is the inability to deal with General Concept Inclusions (GCIs), which is an important feature of classical DLs; e.g., GCIs are necessary to represent domain and range constraints. We note that [29, 20] present a solution for fuzzy DLs under “Zadeh semantics”, but which is hardly applicable in real world scenarios. The major problem relies on the fact that, informally, the algorithm generates a disjunction for each rational occurring in the knowledge base. This is clearly not feasible in practice.

In this paper, we address this issue and develop a calculus for fuzzy $ALC\mathcal{F}(D)$ with GCIs under various semantics, which have been considered in the running system $fuzzyDL$, available from Straccia’s home page.

In the remainder, we proceed as follows. We next describe the basics of fuzzy DLs, then address the inference algorithm and finally we conclude.
2 Fuzzy DLs basics

In this section, we define fuzzy ALCF(D). We recall here the semantics given in [6, 5, 25, 26].

Syntax. In fuzzy ALCF(D), we allow to reason with concrete fuzzy data types, using so-called concrete domains. We recall that ALCF(D) is the basic DL ALC [17] extended with functional roles (also called attributes or features) and concrete domains [14] allowing to deal with data types such as strings and integers. In fuzzy ALCF(D), however, concrete domains are fuzzy sets.

A fuzzy data type theory D = (ΔD, ⊢ D) is such that ⊢ D assigns to every n-ary data type predicate an n-ary fuzzy relation over ΔD. For instance, as for ALCF(D), the predicate ≤18 may be a unary crisp predicate over the natural numbers denoting the set of integers smaller or equal to 18. On the other hand, considering 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 triangular, the trapezoidal, the L-function (left-shoulder function), and the R-function (right-shoulder function) are simple, but most frequently used to specify membership degrees. The functions are defined over the set of non-negative rationals Q+ ∪ {0} (see Fig. 1). Using these functions, we may then define, for instance, Young: Natural → [0, 1] to be a fuzzy concrete predicate over the natural numbers denoting the degree of youngness of a person’s age. The concrete fuzzy predicate Young may be defined as Young(x) = L(x; 10, 30).

We allow modifiers in fuzzy ALCF(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 fα : [0, 1] → [0, 1]. For instance, we may define very(x) = x² and slightly(x) = √x. Modifiers have been considered, for instance, in [9, 28].

Now, let A, R_A, R_C, I, I_e and M be non-empty finite and pair-wise disjoint sets of concepts names (denoted A), abstract roles names (denoted R), i.e. binary predicates concrete roles names (denoted T), abstract individual names (denoted a), concrete individual names (denoted c) and modifiers (denoted m). Concepts may be seen as unary predicates, while roles may be seen as binary predicates. R_A also contains a non-empty subset F_a of abstract feature names (denoted τ), while R_C contains a non-empty subset F_c of concrete feature names (denoted t). Features are functional roles.

The syntax of fuzzy ALCF(D) concepts is as follows:

\[ C : = \top | \bot | A | C_1 \cap C_2 | C_1 \cup C_2 | \neg C | m(C) | \forall R.C | \exists R.C | \forall T_1, \ldots, T_n.D | \exists T_1, \ldots, T_n.D. \]

Concerning axioms and assertions, similarly to [26], we define fuzzy axioms as follows.

Let n ∈ {0, 1}. A fuzzy TBox T is a finite set of fuzzy concept inclusion axioms (C ⊑ D, n), where C, D are concepts. Informally, \((C \sqsubseteq D, 1)\) states that all instances of concept C are instances of concept D to degree n. We write C = D as a shorthand of the two axioms \((C \sqsubseteq D, 1)\) and \((D \sqsubseteq C, 1)\). For instance,

\[ \text{Minor} = \text{Person} \sqcap \exists \text{age}. \leq 18 \]

defines a person, whose age is less or equal to 18 (age is a concrete feature), i.e., it defines a minor,

\[ \text{YoungPerson} = \text{Person} \sqcap \exists \text{age}. \text{Young} \]

defines a young person, while

\[ \text{SportsCar} = \text{Car} \sqcap \exists \text{speed}. \text{very}(\text{High}) \]

we may define the concept of sports car, where very is a concept modifier, e.g. a linear hedge, and High is a fuzzy concrete predicate over the domain of speed expressed in kilometres per hour and may be defined as \(\text{High}(x) = R(x; 80, 250)\) (speed is a concrete feature).

A fuzzy ABox A consists of a finite set of fuzzy concept and fuzzy role assertion axioms of the form \((a : C, n)\), \((a, b) : R, n)\) and \((c : T, n)\), where a, b are abstract individual constants, c is a concrete individual, and C, R and T are a concept, an abstract role and a concrete role, respectively.

Informally, from a semantical point of view, a fuzzy axiom \((a, n)\) constrains the membership degree of a to be at least n. Hence, \((\text{Jim}: \text{YoungPerson}, 0.2)\) says that Jim is a YoungPerson with degree at least 0.2, while \((\text{Jim}, 180) : \text{hasHeight}, 1)\), where hasHeight is a concrete feature, says that Jim’s height is 180. On the other hand, a fuzzy concept inclusion axiom of the form \((C \sqsubseteq D, n)\) says that the subsumption degree between C and D is at least n.

An ALCF(D) fuzzy knowledge base \(K = (T, A)\) consists of a fuzzy TBox T, and a fuzzy ABox A.
The semantics 

The semantics extends [22]. The main idea is that concepts and roles are interpreted as fuzzy subsets of an interpretation’s domain. Therefore, $\text{ACCF}(D)$ axioms, rather than being satisfied (true) or unsatisfied (false) in an interpretation, become a degree of truth in $[0, 1]$. In the following, we use $\land, \lor, \neg$ and $\rightarrow$ in infix notation, in place of a t-norm, s-norm, negation function, and implication function. A fuzzy interpretation $I = (\Delta^I, \neg^I)$ relative to a fuzzy data type theory $D = (\Delta_D, \cdot_D)$ consists of a nonempty set $\Delta^I$ (called the domain), disjoint from $\Delta_D$, and of a fuzzy interpretation function $\mathcal{I}$ that coincides with $\cdot_D$ on every data value, data type, and fuzzy data type predicate, and it assigns $(i)$ to each abstract concept $C$ a function $C^I: \Delta^I \rightarrow [0, 1]$; $(ii)$ to each abstract role $R$ a function $R^I: \Delta^I \times \Delta^I \rightarrow [0, 1]$; $(iii)$ to each abstract feature $r$ a partial function $r^I: \Delta^I \rightarrow [0, 1]$ such that for all $u \in \Delta^I$ there is an unique $w \in \Delta^I$ on which $r^I(u, w)$ is defined; $(iv)$ to each concrete role $T$ a function $T^I: \Delta^I \rightarrow [0, 1]$; $(v)$ to each concrete feature $f$ a partial function $f^I: [0, 1] \rightarrow [0, 1]$; $(vi)$ to each abstract individual $a$ an element in $\Delta^I$; $(vii)$ to each concrete individual $c$ an element in $\Delta_D$.

The mapping $\mathcal{I}$ is extended to roles and complex concepts as specified in the Table 1 (where $x, y \in \Delta^I$ and $\delta \in \Delta_D$).

We comment briefly some points. The semantics of $\exists R.C$ $(\exists R.C)^I(d) = \sup_{y \in \Delta^I} R^I(x, y) \land C^I(y)$ is the result of viewing $\exists R.C$ as the open first order formula $\exists y. [F_R(x, y) \land F_C(y)]$ where $F$ is the obvious translation of roles and concepts into first-order logic (FOL). Similarly, $(\forall R.C)^I(x) = \inf_{y \in \Delta^I} R^I(x, y) \rightarrow C^I(y)$ is related to the open first order formula $\forall y. [F_R(x, y) \rightarrow F_C(y)]$. However, unlike the classical case, in general we do not have that $(\forall R.C)^I = (\neg\exists R.\neg C)^I$. For instance, it holds in Łukasiewicz logic, but not in Gödel logic. Also interesting is that (see [5]) the axiom $T \subseteq (\neg(\forall R.A) \rightarrow (\neg R.\neg A)$ has no classical model. However, in [5], it is shown that in Gödel logic it has no finite model, but has an infinite model.

Finally, the mapping $\mathcal{I}$ is extended to non-fuzzy axioms as specified in the following table (where $a, b$ are individuals):

- $(C \subseteq D)^I = \inf_{x \in \Delta^I} C^I(x) \rightarrow D^I(x)$
- $(a : C)^I = C^I(a^I)$
- $((a, b) : R)^I = R^I(a^I, b^I)$.

Note here that e.g. the semantics of a concept inclusion axiom $C \subseteq D$ is derived directly from its FOL translation, which is of the form $\forall x. F_C(x) \rightarrow F_D(x)$. This definition is clearly different from the approaches in which $C \subseteq D$ is viewed as $\forall x. C(x) \leq D(x)$. This latter approach has the effect that the subsumption relationship is a boolean relationship, while the in former approach subsumption is determined up to a degree in $[0, 1]$.

The notion of satisfaction of a fuzzy axiom $E$ by a fuzzy interpretation $I$, denoted $I \models E$, is defined as follows: $I \models (\alpha \geq n)$, where $\alpha$ is a concept inclusion axiom, iff $\alpha^I \geq n$. Similarly, $I \models (\alpha \geq n)$, where $\alpha$ is a concept or a role assertion axiom, iff $\alpha^I \geq n$. We say that a concept $C$ is satisfiable iff there is an interpretation $I$ and an individual $x \in \Delta^I$ such that $C^I(x) > 0$.

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

A fuzzy axiom $E$ is a logical consequence of a knowledge base $K$, denoted $K \models E$ iff every model of $K$ satisfies $E$.

The interesting point is that according to our semantics, e.g., a minor is a young person to a degree and this relationship is obtained without explicitly mentioning it.

Example 2.1 Consider the knowledge base $K$ with definitions (1) and (2). Then under Łukasiewicz logic we have that (see [24])

$$K \models \langle \text{Minor} \subseteq \text{YoungPerson}, 0.6 \rangle$$

$$K \models \langle \text{YoungPerson} \subseteq \text{Minor}, 0.4 \rangle.$$

Finally, given $K$ and an axiom $\alpha$ of the form $C \subseteq D, a : C, (a, b) : R$ or $(a, c) : T$, it is of interest to compute $\alpha$’s best lower degree value bound. The greatest lower bound of $\alpha$ w.r.t. $K$ (denoted $\text{glb}(K, \alpha)$) is

$$\text{glb}(K, \alpha) = \sup \{ n \mid K \models (\alpha \geq n) \},$$

where $\sup \emptyset = 0$. Determining the $\text{glb}$ is called the Best Degree Bound (BDB) problem. For instance, the consequences in Example 2.1 are the best possible degree bounds. Another similar concept is the best satisfiability bound of a concept $C$ and amounts to determine

$$\text{glb}(K, C) = \sup_{x \in \Delta^I} \sup \{ C^I(x) \mid \forall x^I \models K \}.$$

Essentially, among all models $I$ of the knowledge base, we are determining the maximal degree of truth that the concept $C$ may have over all individuals $x \in \Delta^I$.

Example 2.2 Assume, that a car seller sells an Audi TT for $31500$, as from the catalog price. A buyer is looking for a sports car, but wants to pay no more than around $30000$. In classical DLs no agreement can be found. The problem relies on the crisp condition on the seller’s and the buyer’s price. A more fine grained approach would be (and usually happens in negotiation) to consider prices as concrete fuzzy sets instead. For instance, the seller may consider optimal to sell above $31500$, but can go down to $30500$. The buyer prefers to spend less than $30000$, but can go up to
The procedure differs from the one presented in [23] in how to deal with fuzzy concrete domains and modifiers.

Łukasiewicz semantics. We leave out, for reasons of space, pushing in the usual manner negation on front of concept names, modifiers and concrete predicate names only. With $\text{nff}(C)$ we denote the NNF of concept $C$.

The basic idea behind our reasoning algorithm is as follows. Consider $\mathcal{K} = \langle T, \mathcal{A} \rangle$. In order to solve the BDB problem, we combine appropriate DL tableau rules with methods developed in the context of Many-Valued Logics (MVLs) [4]. In order to determine e.g. $\text{glb}(K, a:C)$, we consider an expression of the form $(a:_C, 1 - x)$ (informally, $(a:C \leq x)$, where $x$ is a $[0,1]$-valued variable. Then we construct a tableau for $\mathcal{K} = \langle T, \mathcal{A} \cup \{\{(a:_C, \neg x))\}\}$ in which the application of satisfiability preserving rules generates new fuzzy assertion axioms together with inequations over $[0,1]$-valued variables. These inequations have to hold in order to respect the semantics of the DL constructors. Finally, in order to determine the greatest lower bound, we minimize the original variable $x$ such that all constraints are satisfied \footnote{Informally, suppose the minimal value is $\bar{n}$. We will know then that for any interpretation $I$ satisfying the knowledge base such that $(a:C)^I < \bar{n}$, the starting set is unsatisfiable and, thus, $(a:C)^I \geq \bar{n}$ has to hold. Which means that $\text{glb}(K, (a:C)) = \bar{n}$.}. Similarly, for $C \subseteq D$, we can compute $\text{glb}(K, C \subseteq D)$ as the minimal value of $x$ such that $\mathcal{K} = \langle T, \mathcal{A} \cup \{(a:C \cap \neg D, 1 - x))\}$ is satisfiable, where $a$ is new abstract individual. Therefore, the BDB problem can be reduced to minimal satisfiability problem of a KB. Finally, concerning the satisfiability bound problem, $\text{glb}(K, C)$ is determined by the maximal value of $x$ such that $\langle T, \mathcal{A} \cup \{(a:C, x))\}$ is satisfiable.

Under Łukasiewicz logic we end up with a bounded Mixed Integer Linear Program (bMILP) optimization problem [15]. Interestingly, as for the MVL case, the tableaux we are generating contains one branch only and, thus, just one bMILP problem has to be solved.

Now, let $\mathbf{V}$ be a new alphabet of variables $x$ ranging over $[0,1]$, $\mathbf{W}$ be a new alphabet of 0-1 variables $y$. We extend fuzzy assertions to the form $(\alpha, l)$, where $l$ is a linear expression over variables in $\mathbf{V}, \mathbf{W}$ and real values.

Similar to crisp DLs, our tableau algorithm checks the satisfiability of a fuzzy KB by trying to build a fuzzy tableau,
from which it is immediate either to build a model in case KB is satisfiable or to detect that the KB is unsatisfiable. The fuzzy tableau we present here can be seen as an extension of the tableau presented in [11], and is inspired by the one presented in [19, 20].

Given $K = \langle T, A \rangle$, let $R_K$ be the set of roles occurring in $K$ and let $\text{sub}(K)$ be the set of named concepts appearing in $K$. A fuzzy tableau $T$ for $K$ is a quadruple $(S, L, E, V)$ such that: $S$ is a set of elements, $L : S \times \text{sub}(K) \to [0, 1]$ maps each element and concept, to a membership degree (the degree of the element being an instance of the concept), and $E : R_K \times (S \times S) \to [0, 1]$ maps each role of $R_K$ and pair of elements to the membership degree of the pair being an instance of the role, and $V : I_A \to S$ maps individuals occurring in $A$ to elements in $S$. For all $s, t \in S$, $C, D \in \text{sub}(K)$, and $R \in R_K$, $T$ has to satisfy:

1. $L(s, \bot) = 0$ and $L(s, \top) = 1$ for all $s \in S$.
2. If $L(s, \neg A) \geq n$, then $L(s, A) \leq -n$.
3. If $L(s, C \cap D) \geq n$, then $L(s, C) \geq m_1$, $L(s, D) \geq m_2$ and $n = m_1 \land m_2$, for some $m_1$ and $m_2$.
4. If $L(s, C \cup D) \geq n$, then $L(s, C) \geq m_1$, $L(s, D) \geq m_2$ and $n = m_1 \lor m_2$, for some $m_1$ and $m_2$.
5. If $L(s, \forall R.C) \geq n$, then $E(R, (s, t)) \leq L(t, C) + 1 - n$ for all $t \in S$.
6. If $L(s, \exists R.C) \geq n$, then there exists $t \in S$ such that $E(R, (s, t)) \geq m_1$, $L(t, C) \geq m_2$ and $n = m_1 \lor m_2$, for some $m_1$ and $m_2$.
7. If $(C \subseteq D, n) \in T$, then $L(s, C) \subseteq L(s, D) + 1 - n$, for all $s \in S$.
8. If $(a, C, n) \in A$, then $L(V(a), C) \geq n$.
9. If $((a, b) \cdot R, n) \in A$, then $E(R, (V(a), V(b))) \geq n$.
10. If $((a, c) \cdot T, n) \in A$, then $E(T, (V(a), V(c))) \geq n$.

Proposition 3.1 $K = \langle T, A \rangle$ is satisfiable if there exists a fuzzy tableau for $K$.

Proof: [Sketch] For the if direction if $T = (S, L, E, V)$ is a fuzzy tableau for $K$, we can construct a fuzzy interpretation $I = (\Delta^T, \sqsubseteq^T)$ that is a model of $A$ and $T$ as follows:

- $\Delta^T = S$, $a^T = V(a)$, $a$ occurs in $A$
- $\sqsubseteq^T(s) = L(s, \top)$, $\bot^T(s) = L(s, \bot)$, for all $s \in S$
- $A^T(s) = L(a, A),$ for all $s \in S$
- $R^T(s, t) = E(R, (s, t))$ for all $(s, t) \in S \times S$

To prove that $I$ is a model of $A$ and $T$, we can show by induction on the structure of concepts that $L(s, C) \geq n$ implies $C^T(s) \geq n$ for all $s \in S$. Together with properties 7–10, this implies that $I$ is a model of $T$, and that it satisfies each fuzzy assertion in $A$.

For the converse, we know from [5, 6] that if $\mathcal{K}$ has a model then it has a witnessed model. That is, $I = (\Delta^T, \sqsubseteq^T)$ is a witnessed model of $\mathcal{K}$, if for all $x \in \Delta^T$ there is $y \in \Delta^T$ such that

$$(\exists R.C)^T(x) = R^T(x, y) \land C^T(y)$$

and there is $x \in \Delta^T$ such that

$$(C \subseteq D)^T = C^T(x) \rightarrow D^T(x).$$

So, let $I$ be a witnessed model of $\mathcal{K}$. Then a fuzzy tableau $T = (S, L, E, V)$ for $K$ can be defined as follows:

$S = \Delta^T, E(R, (s, t)) = R^T(s, t), L(s, C) = C^T(s), V(a) = a^T$

It can be verified that $T$ is a fuzzy tableau for $K$.

Now, in order to decide the satisfiability of $\mathcal{K} = \langle T, A \rangle$ a procedure that constructs a fuzzy tableau $T$ for $K$ has to be determined. Like the tableau algorithm presented in [20], our algorithm works on completion-forests since an ABox might contain several individuals with arbitrary roles connecting them. Due to the presence of general or cyclic terminology $T$, the termination of the algorithm has to be ensured. This is done by providing a blocking condition for rule applications.

Let $K = \langle T, A \rangle$ be a fuzzy KB. A completion-forest $F$ for $K$ is a collection of trees whose distinguished roots are arbitrarily connected by edges.

Each node $v$ is labelled with a sequence $L(v)$ of expressions of the form $\langle C, l \rangle$, where $C \in \text{sub}(K)$, and $l$ is either a rational, a variable $x$, or a negated variable, i.e. of the form $1 - x$, where $x$ is a variable (the intuition here is that $v$ is an instance of $C$ to degree equal or greater than of the evaluation of $l$).

Each edge $\langle v, w \rangle$ is labelled with a sequence $L((v, w))$ of expressions of the form $\langle R, l \rangle$, where $R \in R_K$ are roles occurring in $K$ (the intuition here is that $\langle v, w \rangle$ is an instance of $R$ to degree equal or greater than of the evaluation of $l$).

The forest has associated a set $C_F$ of constraints of the form $c \leq c', c = c', x_i \in [0, 1], y_i \in \{0, 1\}$, on the variables occurring the node labels and edge labels. $c, c'$ are linear expressions.

If nodes $v$ and $w$ are connected by an edge $\langle v, w \rangle$ with $\langle R, l \rangle$ occurring in $L((v, w))$, then $w$ is called an $R_l$-successor of $v$ and $w$ is called an $R_l$-predecessor of $w$.

A node $v$ is an $R$-successor (resp. $R$-predecessor) of $w$ if it is an $R_l$-successor (resp. $R_l$-predecessor) of $w$ for some role $R$. As usual, ancestor is the transitive closure of predecesor.

We say that two non-root nodes $v$ and $w$ are equivalent, denoted $L(v) \approx L(w)$, if $L(v) = [(C_1, l_1), \ldots, (C_n, l_k)]$, $L(w) = [(C_1, l'_1), \ldots, (C_n, l'_k)]$, and for all $1 \leq i \leq k$, either both $l_i$ and $l'_i$ are variables, or both $l_i$ and $l'_i$ are negated variables or both $l_i$ and $l'_i$ are the same rational in $[0, 1]$ (the intuition here is that $v$ and $w$ share the same properties).
A node $v$ is directly blocked iff none of its ancestors are blocked, it is not a root node, and it has an ancestor $w$ such that $L(v) \equiv L(w)$. In this case, we say $w$ directly blocks $v$. A node $v$ is blocked iff it is directly blocked or if one of its predecessor is blocked (the intuition here is that we need not further to apply rules to node $v$, as an equivalent predecessor node $w$ of $v$ exists).

The algorithm initializes a forest $\mathcal{F}$ to contain (i) a root node $v_0$, for each individual $a_i$ occurring in $A$, labelled with $L(v_0^i)$ such that $L(v_0^i)$ contains $(C_i, n)$ for each fuzzy assertion $(a_i; C_i, n) \in A$, and (ii) an edge $(v_0, v_1)$, for each fuzzy assertion $\langle a_i, a_j; R_{ij}, n \rangle \in A$, labelled with $L((v_0^i, v_0^j))$ such that $L((v_0^i, v_0^j))$ contains $(R_{ij}, n)$. $\mathcal{F}$ is then expanded by repeatedly applying the completion rules described below. The completion-forest is complete when none of the completion rules are applicable. Then, the bMILP problem on the set of constraints $\mathcal{C}_F$ is solved.

We also need a technical definition involving feature roles (see [14]). Let $\mathcal{F}$ be forest, $r$ an abstract feature such that we have two edges $\langle v, w \rangle$ and $\langle v, w \rangle$ such that $(r, l_1)$ and $(r, l_2)$ occur in $\mathcal{L}(v)$ and $\mathcal{L}(v, w)$, respectively (informally, $\mathcal{F}$ contains $\langle (v, w), r, l_1 \rangle$ and $\langle (v, w), r, l_2 \rangle$). Then we call such a pair a fork. As $r$ is a function, such a fork means that $w_1$ and $w_2$ have to be interpreted as the same individual. Such a fork can be deleted by adding both $L((v, w_0))$ to $\mathcal{L}(v, w_0)$ and $\mathcal{L}(w_2)$ to $\mathcal{L}(w_1)$, and then deleting node $w_2$. A similar argument applies to concrete feature roles. At the beginning, we remove the forks from the initial forest. We assume that forks are eliminated as soon as they appear (as part of a rule application) with the proviso that newly generated nodes are replaced by older ones and not vice-versa.

We also assume a fixed rule application strategy as e.g. the order of rules below, such that the rule for $(\exists)$ is applied as last. Also, all expressions in node labels are processed according to the order they are introduced into the $\mathcal{F}$.

With $x_C$ we denote the variable associated to the atomic assertion $C$ of the form $a; A$ or $(a, b); R$. $x_C$ will take the truth value associated to $a$, while with $x$, we denote the variable associated to the concrete individual $c$. The rules are the following:

(A). If $\langle A, l \rangle \in L(v)$ then $C_F = C_F \cup \{x:v; A \geq l \} \cup \{x:v; A \in [0, l]\}$.

(\bar{A}). If $\langle \sim A, l \rangle \in L(v)$ then $C_F = C_F \cup \{x:v; A \leq l \} \cup \{x:v; A \in [0, l]\}$.

(R). If $\langle R, l \rangle \in L(v, w)$ then $C_F = C_F \cup \{x(v, w); R \geq l \} \cup \{x(v, w); R \in [0, l]\}$.

(\exists). If $\langle C \cap D, l \rangle \in L(v)$ then append $\langle C, x_1 \rangle$ and $\langle D, x_2 \rangle$ to $L(v)$, and $C_F = C_F \cup \{y \leq 1 - l, x_1 \leq 1 - y, x_1 + x_2 = l + 1 - y, x_1 \in [0, 1], y \in [0, 1]\}$, where $x_1, y$ are new variables.

(\forall). If $\langle C \cap D, l \rangle \in L(v)$ then append $\langle C, x_1 \rangle$ and $\langle D, x_2 \rangle$ to $L(v)$, and $C_F = C_F \cup \{x_1 + x_2 = l, x_1 \in [0, 1]\}$, where $x_1$ are new variables.

(\forall f C. C \cap D, l \rangle \in L(v), \langle R, l \rangle \in L(<v, w>)$ and the rule has not been already applied to this pair then append $\langle C, x \rangle$ to $L(w)$ and $C_F = C_F \cup \{x \geq l, x \in [0, 1], y \in [0, 1]\}$, where $x, y$ are new variables. The case for concrete rules is similar.

(\exists). If $\langle C \cap D, n \rangle \in T$ and $v$ is a node to which this rule has not yet been applied then append $\langle \sim C, 1 - x_1 \rangle$ and $\langle D, x_2 \rangle$ to $L(v)$, and $C_F = C_F \cup \{x_1 \leq x_2 + 1 - n \}$.

(\forall f C. C \cap D, l \rangle \in L(v)$ and $v$ is not blocked then create a new node $w$ and append $\langle R, x_1 \rangle$ to $L(v, w)$ and $\langle C, x_2 \rangle$ to $L(w)$, and $C_F = C_F \cup \{y \leq l - x_1, y_1 \leq x - y, x_1 + x_2 = l + 1 - y, x_1 \in [0, 1], y \in [0, 1]\}$, where $x_1, y$ are new variables. The case for concrete rules is similar.

Let’s see an example.

Example 3.1 Consider $K = (T, A)$, where $T = \{\exists R.C \subseteq D, 1\}$ and $A = \{(a, b), R_0, 0.7, (b, C, 0.8)\}$. Let us show that $\text{gb}(K, a; D) = 0.5$. To this end, we have to determine the minimal value for such that $(T, A, \cup \{a; (a, D, 1 - x)\})$ is satisfactory.

To start with, we construct a forest $\mathcal{F}$ with two nodes $a$ and $b$ (one for each individual in $A$). We process first $\langle a, b \rangle; R_0, 0.7$, then $\langle b, C, 0.8 \rangle$ and finally $\langle a; D, 1 - x \rangle$. Therefore, we set $L(a) = \{[(C, 0.8), (\sim D, 1 - x)], L(a, b) = [[(R, 0.7)]\}$ and $C_F = \{x \in [0, 1]\}$.

We first process $\langle R, 0.7 \rangle \in L(a, b)$, apply rule $(R)$ and, thus, add $x(a, b); R \geq 0.7$ and $x(a, b); R \in [0, 1]$ to $C_F$.

Then we process $\langle C, 0.8 \rangle \in L(a)$, apply rule $(A)$ and, thus, add $x(a; C) \geq 0.8$ and $x(a; C) \in [0, 1]$ to $C_F$.

We next process $\langle \sim D, 1 - x \rangle \in L(a)$, apply rule $(A)$ and, thus, add $x_D \leq x$ and $x_D \in [0, 1]$ to $C_F$.

Now, we apply $(\exists)$ to $a$ and $(\exists R.C \subseteq D, 1)$, and, thus, we append $\langle \sim R, \sim C, 1 - x_1 \rangle$ and $\langle D, x_2 \rangle$ to $L(a)$, and we add $x_1 \leq x_2$ and $x_1, x_2 \in [0, 1]$ to $C_F$.

Similarly, we apply $(\exists)$ to $b$: we append $\langle \sim R, \sim C, 1 - x_3 \rangle$ and $\langle D, x_4 \rangle$ to $L(b)$, and we add $x_3 \leq x_4$ and $x_3 \in [0, 1]$ to $C_F$.

Next we process $\langle \sim R, \sim C, 1 - x_1 \rangle \in L(a)$, apply rule $(\forall f)$ to it and, $R, 0.7 \rangle \in L(a)$) and, thus, we append $\langle \sim C, x_2 \rangle$ to $L(b)$, and we add $x_2 \geq 1 - x_1 + 0.7 - 1, x_2 \leq y, y_1 = 1 - x_3 - 0.3 \leq y, x_1 + x_2 = 1 + 1 - y, x_1 \in [0, 1], y \in [0, 1]$ to $C_F$.

We next process $\langle D, x_2 \rangle \in L(a)$, apply rule $(A)$ and, thus, add $x_D \leq x_2$ and $x_D \in [0, 1]$ to $C_F$.

Next we process $\langle \sim R, \sim C, 1 - x_1 \rangle \in L(b)$, but no rule is applicable to it. Finally, we process $\langle D, x_4 \rangle \in L(b)$, apply rule $(A)$ and, thus, add $x_D \geq x_4$ and $x_D \in [0, 1]$ to $C_F$.

Now the forest $\mathcal{F}$ is complete as no more rule is applicable and we consider the set of inequations $C_F$. It remains to solve the bMILP problem on $C_F$. Indeed, it holds that $\text{gb}(K, a; D) = \min x.C_F$. It can be verified that this value is 0.5.

Example 3.2 illustrates the behaviour on cyclic terminologies and shows how a potential infinite cyclic computation is blocked.
\[ P(a, w) = [(A, 0.8), (\neg A, 1 - x_1), (\exists R.A, x_2)] \]

**Example 3.2** Consider \( K = \langle T, A \rangle \), where \( T = \{ \{ A \subseteq \exists R.A \} \} \) and \( A = \{ (a: A, 0.8) \} \). Let’s show that \( K \) is satisfiable. The complete forest \( F \) in Figure 3 shows the computation.

As we can notice, node \( w_2 \) is blocked by node \( w_1 \), meaning that \( w_2 \) and \( w_1 \) share the same properties. In order to build a model, we replace all occurrences of \( w \) with \( \bar{w} \) and then we find a solution to the inequalities. Our blocking condition is based on the fact that, in case of cyclic definitions, as the calculus is deterministic, the same sequence of expressions has to be generated within a cycle.

**Proposition 3.2 (Termination)** For each KB \( K \), the tableau algorithm terminates.

**Proof:** [Sketch] Termination is a result of the properties of the expansion rules, as in the classical case [11]. More precisely we have the following observations. (i) The expansion rules never remove nodes from the tree (except forks at the beginning) or concepts from node labels or change the edge labels. (ii) Successors are only generated by the rule \( \exists \). For any node and for each concept these rules are applied at most once. (iii) Since nodes are labelled with nonempty sequences of \( \text{sub}(K) \), obviously there is a finite number of possible labelling for a pair of nodes. Thus, the blocking condition will be applied in any path of the tree and consequently any path will have a finite length. \( \square \)

**Proposition 3.3 (Soundness)** If the expansion rules can be applied to a KB \( K \) such that they yield a complete completion-forest \( F \) such that \( C_F \) has a solution, then \( K \) has a fuzzy tableau for \( K \).

**Proof:** [Sketch] Let \( F \) be a complete completion-forest constructed by the tableau algorithm for \( K \). By hypothesis, \( C_F \) has a solution. If \( x \) is a variable occurring in \( C_F \), with \( \bar{x} \) we denote the value of \( x \) in this solution. If the variable \( x \) does not occur in \( C_F \) then \( \bar{x} = 0 \) is assumed. A fuzzy tableau \( T = (S, L, E, V) \) can be defined as follows:

\[
S = \{ v \mid v \text{ is a node in } F, \text{ and } v \text{ is not blocked} \},
\]

\[
\mathcal{L}(v, \bot) = 0, \text{ if } v \in S,
\]

\[
\mathcal{L}(v, \top) = 1, \text{ if } v \in S,
\]

\[
\mathcal{L}(v, A) = \mathcal{L}(v, \bar{A}), \text{ if } v \text{ in } F \text{ not blocked},
\]

\[
\mathcal{E}(R, \{ v, w \}) = \mathcal{E}(R, \{ w, v \}); R, \text{ if } v, w \text{ in } F \text{ not blocked}
\]

\[
\mathcal{V}(a) = w^0, \text{ where } w^0 \text{ is a root node}
\]

It can be shown that \( T \) is a fuzzy tableau for \( K \). \( \square \)

**Proposition 3.4 (Completeness)** Consider a KB \( K \). If \( K \) has a fuzzy tableau, then the expansion rules can be applied in such a way that the tableaux algorithm yields a complete completion-forest \( F \) such that \( C_F \) has a solution.

**Proof:** [Sketch] Let \( T = (S, L, E, V) \) be a fuzzy tableau for \( K \). Using \( T \), we can trigger the application of the expansion rules such that they yield a complete forest \( F \) that is complete. Using \( L \) and \( E \) we can find a solution to \( C_F \). \( \square \)

4 Conclusions

In this work we presented a reasoning algorithm for \( ALC.F(D) \) with general concept inclusions and explicit membership functions (\( ALC.F \) is a guarded logic [3]), under Łukasiewicz semantics. Clearly, by using [23] the result applies also to the usually used “Zadeh semantics” of fuzzy DLs, and of course also for the classical boolean variant of \( ALC.F \).

The result can be extended to more expressive fuzzy DLs, such as \( SHIF(D) \) and \( SHOIN(D) \), which are the DLs behind the web ontology description languages OWL-DL and OWL-Lite, by adapting our blocking condition similarly as done in [10]. The description of these blocking conditions will be the subject of an extend paper. So far, an implementation for fuzzy \( SHIF(D) \) can be found from Straccia’s home page.

Acknowledgement

Fernando Bobillo holds a FPU scholarship from the Spanish Ministerio de Educación y Ciencia.

References


I. Horrocks, P. F. Patel-Schneider and F. van Harmelen. Foundations of Integer Pro-

I. Horrocks, U. Sattler and S. Tobies. Reasoning with indi-

D. Sánchez and A. G.B. Tettamanzi. Fuzzy quantification in

description logic ALC. In R. Khosla, R. J. Howlett, and
L. C. Jain, editors, Proceedings of the 9th International Conference,
Knowledge-Based Intelligent Information and Engineering Systems (KES-05), volume 3684 of Lecture
Notes in Computer Science, pages 896–902. Springer Verlag,
2005.

I. Horrocks, P. F. Patel-Schneider and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontol-

description logic ALCN. In R. Khosla, R. J. Howlett, and
L. C. Jain, editors, Proceedings of the 9th International Conference,
Knowledge-Based Intelligent Information and Engineering Systems (KES-05), volume 3684 of Lecture
Notes in Computer Science, pages 896–902. Springer Verlag,
2005.

C. Lutz. Description logics with concrete domains - A sur-
vey. In Advances in Modal Logics Volume 4. King’s College

H. Salkin and M. Kamlesh. Foundations of Integer Pro-

D. Sánchez and A. G.B. Tettamanzi. Fuzzy quantification in
fuzzy description logics. In E. Sanchez, editor, Capturing
Intelligence: Fuzzy Logic and the Semantic Web, Elsevier,
2006.

M. Schmidt-Schauß and G. Smolka. Attributive concept de-
scriptions with complements. Artificial Intelligence, 48:1–

Logic and Applications, volume 2955 in Lecture Notes in

E. Grädel, C. Hirsch and M. Otto. Back and forth between
guarded and modal logics. ACM Transactions on Computa-

R. Hähnle. Advanced many-valued logics. In D. M. Gabb-
bay and F. Guenthner, editors, Handbook of Philosophical


P. Hájek. What does mathematical fuzzy logic offer to de-
scription logic? In Elie Sanchez, editor, Capturing Intelli-

S. Hölldobler, N. H. Nga and T. D. Khang. The fuzzy de-
scription logic ALCFH. In Proceedings of the Interna-
tional Workshop on Description Logics (DL-05), 2005.

S. Hölldobler, H.-P. Störr and T. D. Khang. The fuzzy de-
scription logic ALCFH with hedge algebras as concept modifiers. Journal of Advanced Computational Intelligence,

S. Hölldobler, H.-P. Störr and T. D. Khang. The subsump-
tion problem of the fuzzy description logic ALCFH. In
Proceedings of the 10th International Conference on In-
formation Processing and Management of Uncertainty in


I. Horrocks, U. Sattler and S. Tobies. Reasoning with indi-
viduals for the description logic SHIQ. In D. MacAllester,
etitor, Proceedings of the 17th International Conference on
Automated Deduction (CADE-17), volume 1831 in Lecture
Notes in Artificial Intelligence, pages 482–496, Germany,

I. Horrocks, P. F. Patel-Schneider and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontol-

description logic ALCN. In R. Khosla, R. J. Howlett, and
L. C. Jain, editors, Proceedings of the 9th International Conference,
Knowledge-Based Intelligent Information and Engineering Systems (KES-05), volume 3684 of Lecture
Notes in Computer Science, pages 896–902. Springer Verlag,
2005.

C. Lutz. Description logics with concrete domains - A sur-
vey. In Advances in Modal Logics Volume 4. King’s College

H. Salkin and M. Kamlesh. Foundations of Integer Pro-

D. Sánchez and A. G.B. Tettamanzi. Fuzzy quantification in
fuzzy description logics. In E. Sanchez, editor, Capturing
Intelligence: Fuzzy Logic and the Semantic Web, Elsevier,
2006.

M. Schmidt-Schauß and G. Smolka. Attributive concept de-
scriptions with complements. Artificial Intelligence, 48:1–

G. Stoilos, G. Stamou, V. Tzouvaras, J. Z. Pan and I. Hor-
rocks. The fuzzy description logic f-SHIN. Proceedings of

G. Stoilos, G. Stamou, V. Tzouvaras, J. Z. Pan and lan Hor-
rocks. A Fuzzy Description Logic for Multimedia Knowledge

concept inclusions in fuzzy description logics. In Proceed-
ings of the 17th European Conference on Artificial Intelli-

U. Straccia. A framework for the retrieval of multimedia
objects based on four-valued fuzzy description logics. In
F. Crestani and G. Pasi, editors, Soft Computing in Infor-
mation Retrieval: Techniques and Applications, pages 332–

U. Straccia. Reasoning within fuzzy description logics. Journal of Artificial Intelligence Research, 14:137–166,

U. Straccia. Description logics with fuzzy concrete do-
mains. In F. Bachus and T. Jaakkola, editors, Proceedings of the 21st Conference on Uncertainty in Artificial Intelli-
AUAI Press.

U. Straccia. Fuzzy description logics with concrete do-
e Tecnologie dell’Informazione, Consiglio Nazionale delle
Ricerche, Pisa, Italy, 2005.

U. Straccia. Towards a fuzzy description logic for the se-
mantic web (preliminary report). In 2nd European Semantic
Web Conference (ESWC-05), volume 3532 in Lecture Notes
Springer Verlag.

U. Straccia. A fuzzy description logic for the semantic web. In E. Sanchez, editor, Fuzzy Logic and the Semantic Web,
Capturing Intelligence, chapter 4, pages 73–90. Elsevier,
2006.

U. Straccia. Uncertainty and description logic programs
over lattices. In E. Sanchez, editor, Fuzzy Logic and the Se-
mantic Web, Capturing Intelligence, chapter 7, pages 115–

C. Tresp and R. Molitor. A description logic for vague
knowledge. In Proceedings of the 15th European Confer-
ence on Artificial Intelligence (ECAI-98), 1998.

J. L. Y. Li, B. Xu and D. Kang. Discrete tableau algorithms
for SHI. In Proceedings of the International Workshop on

J. Yen. Generalizing term subsumption languages to fuzzy
logic. In Proc. of the 12th International Joint Conference on

L. Zadeh. Fuzzy sets. Information and Control, 8(3):338–

model for searching in semantic portals. In WWW’05: Proc.
of the 14th international conference on World Wide Web,