Mathware & Soft Computing 14 (2007) 247-259

Mixed Integer Programming, General Concept Inclusions and Fuzzy Description Logics

U. Straccia[1] and F. Bobillo[2]
[1] ISTI - CNR, Pisa, Italy
[2] Dpt. of Computer Science and AI
University of Granada, Spain
straccia@isti.cnr.it fbobillo@decsai.ugr.es

Abstract

Fuzzy Description Logics (fuzzy DLs) have been proposed as a language to describe structured knowledge with vague concepts. In [19], 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.

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 [10] for details).

Fuzzy DLs extend classical DLs by allowing to deal with *fuzzy, vague and 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 [22], 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 (see [12] for an overview).

From a semantics point of view, most works rely on the Gödel conjunction and disjunction ($x \otimes y = \min(x, y)$, $x \oplus y = \max(x, y)$), but with Łukasiewicz negation ($\ominus x = 1 - x$), essentially the semantics of fuzzy sets operators proposed by Zadeh [22] (which we call here "Zadeh semantics"). Although Łukasiewcz full fuzzy logics have been widely studied [4], very few work address fuzzy DLs under Łukasiewcz semantics. Indeed, Hajek [5, 6] proposes a reasoning solution based on a reduction to Łukasiewicz propositional logic Ł, while Straccia [19] 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 [11, 17] 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.

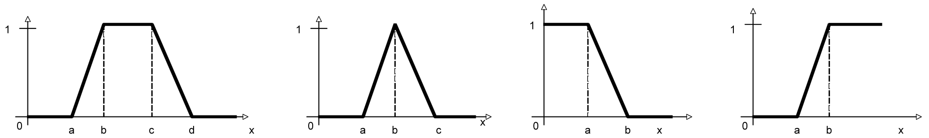
In this paper, we address this issue and develop a calculus for fuzzy $\mathcal{ALCF}(\mathbf{D})$ with GCIs under various semantics, which have been considered in the running system *fuzzyDL*, available from Straccia's home page. In the remainder, we proceed as follows. Section 2 describes the basics of fuzzy DLs, then Section 3 addresses the inference algorithm and finally Section 4 sets out some conclusions.

2 Fuzzy DLs basics

We next define fuzzy $\mathcal{ALCF}(D)$. We recall here the semantics given in [5, 6, 20].

Syntax. In fuzzy $\mathcal{ALCF}(D)$, we allow to reason with concrete fuzzy data types, using so-called concrete domains. We recall that $\mathcal{ALCF}(\mathbf{D})$ is the basic DL $\mathcal{ALC}$ [15] extended with functional roles (also called attributes or features) and concrete domains [13] allowing to deal with data types such as strings and integers. In fuzzy $\mathcal{ALCF}(\mathbf{D})$, however, concrete domains are fuzzy sets.

A *fuzzy data type theory* $\mathbf{D} = \langle \Delta_{\mathbf{D}}, \cdot_{\mathbf{D}} \rangle$ is such that $\cdot_{\mathbf{D}}$ assigns to every n-ary data type predicate an n-ary fuzzy relation over $\Delta_{\mathbf{D}}$. For instance, as for $\mathcal{ALCF}(\mathbf{D})$, the predicate $\leqslant_{18}$ may be a unary crisp predicate over the natural numbers denoting the set of integers smaller or equal to 18. On the other hand, concerning non-crisp fuzzy domain predicates, we recall that in fuzzy set theory and practice, there are many functions for specifying fuzzy set membership degrees. However, the 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 $\mathbb{Q}^+ \cup \{0\}$ (see Fig. 1). Using these functions, we may then



(a) (b) (c) (d)

Figure 1: (a) Trapezoidal function; (b) Triangular function; (c) L-function; (d) R-function

define, for instance, $\mathit{Young}\colon \mathit{Natural} \to [0,1]$ 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 $\mathit{Young}(x) = L(x; 10, 30)$.

We allow modifiers in fuzzy $\mathcal{ALCF}(\mathbf{D})$. 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 $\mathit{very}(x) = x^2$ and $\mathit{slightly}(x) = \sqrt{x}$. Modifiers have been considered, for instance, in [7, 21].

Now, let $\mathbf{A}$, $\mathbf{R}_A$, $\mathbf{R}_C$, $\mathbf{I}$, $\mathbf{I}_c$ 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. $\mathbf{R}_A$ also contains a non-empty subset $\mathbf{F}_a$ of *abstract feature names* (denoted r), while $\mathbf{R}_C$ contains a non-empty subset F_c of *concrete feature names* (denoted t). Features are functional roles.

The syntax of fuzzy $\mathcal{ALCF}(\mathbf{D})$ concepts is as follows:

$$C \quad := \quad \top \mid \; \bot \; \mid A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid m(C) \mid \forall R.C \mid \exists R.C \mid \forall T.D \mid \exists T.D\,.$$

Fuzzy axioms are defined as follows, similarly to [20]. An $\mathcal{ALCF}(\mathbf{D})$ *fuzzy knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ consists of a fuzzy TBox $\mathcal{T}$, and a fuzzy ABox $\mathcal{A}$.

Let $n \in (0,1]$. A *fuzzy TBox* $\mathcal{T}$ is a finite set of *fuzzy concept inclusion axioms* $\langle C \sqsubseteq D, n \rangle$, where C, D are concepts. Informally, $\langle C \sqsubseteq D, n \rangle$ states that all instances of concept C are instances of concept D to degree n. For example, $\langle \mathit{Father} \sqsubseteq \mathit{Person} \sqcap \forall \mathit{hasChild}.\mathit{Person}, 1 \rangle$ (a father is a person and all his children are also persons). We write $C = D$ as a shorthand of the two axioms $\langle C \sqsubseteq D, 1 \rangle$ and $\langle D \sqsubseteq C, 1 \rangle$. For instance, $\mathit{Minor} = \mathit{Person} \sqcap \exists \mathit{age}.\leqslant_{18}$ defines a person whose age is less or equal to 18 (*age* is a concrete feature), i.e., it defines a minor, $\mathit{YoungPerson} = \mathit{Person} \sqcap \exists \mathit{age}.\mathit{Young}$ defines a young person, while we may define the concept of sports car as $\mathit{SportsCar} = \mathit{Car} \sqcap \exists \mathit{speed}.\mathit{very}(\mathit{High})$, 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 $\mathit{High}(x) = R(x; 80, 250)$ (*speed* is a concrete feature).

A *fuzzy ABox* $\mathcal{A}$ consists of a finite set of *fuzzy concept* and *fuzzy role assertion axioms* of the form $\langle a{:}C, n \rangle$, $\langle (a,b){:}R, n \rangle$ and $\langle (a,c){:}T, n \rangle$, 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 $\langle \alpha, n\rangle$ constrains the membership degree of α to be at least n. Hence, $\langle \mathit{jim}\!:\!\mathit{YoungPerson}, 0.2\rangle$ says that *jim* is a *YoungPerson* with degree at least 0.2, while $\langle (\mathit{jim}, \mathit{180})\!:\!\mathit{hasHeight}, 1\rangle$, 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 $\langle C \sqsubseteq D, n\rangle$ says that the subsumption degree between C and D is at least n.

Semantics. The semantics extends [18]. In $\mathcal{ALCF}(\mathbf{D})$ axioms, rather than being "classical" evaluated (being either true or false), they are "many-valued" evaluated taking a degree of truth in $[0, 1]$. In the following, we use $\otimes, \oplus, \ominus$ and $\Rightarrow$ in infix notation, in place of a t-norm, s-norm, negation function, and implication function, respectively.

A *fuzzy interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ relative to a fuzzy data type theory $\mathbf{D} = \langle \Delta_{\mathbf{D}}, \cdot_{\mathbf{D}}\rangle$ consists of a nonempty set $\Delta^{\mathcal{I}}$ (called the *domain*), disjoint from $\Delta_{\mathbf{D}}$, and of a *fuzzy interpretation function* $\cdot^{\mathcal{I}}$ that coincides with $\cdot_{\mathbf{D}}$ on every data value, data type, and fuzzy data type predicate, and it assigns (i) to each abstract concept C a function $C^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \to [0, 1]$; (ii) to each abstract role R a function $R^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0, 1]$; (iii) to each abstract feature r 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 concrete role T a function $R^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}} \to [0, 1]$; (v) to each concrete feature t a partial function $t^{\mathcal{I}}\colon \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}} \to [0, 1]$ such that for all $u \in \Delta^{\mathcal{I}}$ there is an unique $o \in \Delta_{\mathbf{D}}$ on which $t^{\mathcal{I}}(u, o)$ is defined; (vi) to each modifier m the modifier function $f_m\colon [0, 1] \to [0, 1]$; (vi) to each abstract individual a an element in $\Delta^{\mathcal{I}}$; (vii) to each concrete individual c an element in $\Delta_{\mathbf{D}}$.

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

$$
\begin{array}{rcl}
\bot^{\mathcal{I}}(x) & = & 0 \\
\top^{\mathcal{I}}(x) & = & 1 \\
(C_1 \sqcap C_2)^{\mathcal{I}}(x) & = & {C_1}^{\mathcal{I}}(x) \otimes {C_2}^{\mathcal{I}}(x) \\
(C_1 \sqcup C_2)^{\mathcal{I}}(x) & = & {C_1}^{\mathcal{I}}(x) \oplus {C_2}^{\mathcal{I}}(x) \\
(\neg C)^{\mathcal{I}}(x) & = & \ominus C^{\mathcal{I}}(x) \\
(m(C))^{\mathcal{I}}(x) & = & f_m(C^{\mathcal{I}}(x)) \\
(\forall R.C)^{\mathcal{I}}(x) & = & \inf_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \Rightarrow C^{\mathcal{I}}(y) \\
(\exists R.C)^{\mathcal{I}}(x) & = & \sup_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \otimes C^{\mathcal{I}}(y) \\
(\forall T.D)^{\mathcal{I}}(x) & = & \inf_{y \in \Delta_{\mathbf{D}}} T^{\mathcal{I}}(x, y) \Rightarrow D^{\mathcal{I}}(y) \\
(\exists T.D)^{\mathcal{I}}(x) & = & \sup_{y \in \Delta_{\mathbf{D}}} T^{\mathcal{I}}(x, y) \otimes D^{\mathcal{I}}(y)\,.
\end{array}
$$

Table 1: Fuzzy DL semantics.

We comment briefly some points. The semantics of $\exists R.C$ $(\exists R.C)^{\mathcal{I}}(d) = \sup_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \otimes C^{\mathcal{I}}(y)$ is the result of viewing $\exists R.C$ as the open first order formula $\exists y.F_R(x, y) \otimes F_C(y)$ (where F is the obvious translation of roles and concepts into first-order logic (FOL)). Similarly, $(\forall R.C)^{\mathcal{I}}(x) = \inf_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \Rightarrow C^{\mathcal{I}}(y)$ 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 $(\forall R.C)^{\mathcal{I}} = (\neg\exists R.\neg C)^{\mathcal{I}}$. For instance, it holds in Łukasiewicz logic, but not in Gödel logic. Also interesting is that (see [5]) the axiom $\top \sqsubseteq \neg(\forall R.A) \sqcap (\neg\exists R.\neg A)$ 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 $\cdot^{\mathcal{I}}$ is extended to non-fuzzy axioms as specified in the following table (where a, b are individuals):

$$\begin{array}{rcl}
(C \sqsubseteq D)^{\mathcal{I}} & = & \inf_{x\in\Delta^{\mathcal{I}}} C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x) \\
(a{:}C)^{\mathcal{I}} & = & C^{\mathcal{I}}(a^{\mathcal{I}}) \\
((a,b){:}R)^{\mathcal{I}} & = & R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}})\ .
\end{array}$$

Note here that e.g. the semantics of a concept inclusion axiom $C \sqsubseteq D$ 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 $C \sqsubseteq D$ is viewed as $\forall x.C(x) \leqslant D(x)$. 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 $\mathcal{I}$, denoted $\mathcal{I} \models E$, is defined as follows: $\mathcal{I} \models \langle \alpha \geqslant n\rangle$, where α is a concept inclusion axiom, if and only if $\alpha^{\mathcal{I}} \geqslant n$. Similarly, $\mathcal{I} \models \langle \alpha \geqslant n\rangle$, where α is a concept or a role assertion axiom, if and only if $\alpha^{\mathcal{I}} \geqslant n$. We say that a concept C is *satisfiable* if and only if there is an interpretation $\mathcal{I}$ and an individual $x \in \Delta^{\mathcal{I}}$ such that $C^{\mathcal{I}}(x) > 0$.

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

A fuzzy axiom E is a *logical consequence* of a knowledge base $\mathcal{K}$ ($\mathcal{K} \models E$) if and only if every model of $\mathcal{K}$ 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. For example, using the previous definitions of *Minor* and *YoungPerson*, under Łukasiewicz logic we have that $\mathcal{K} \models \langle \mathit{Minor} \sqsubseteq \mathit{YoungPerson}, 0.6\rangle$ ([19]) and $KB \models \langle \mathit{YoungPerson} \sqsubseteq \mathit{Minor}, 0.4\rangle$.

Finally, given $\mathcal{K}$ and an axiom α of the form $C \sqsubseteq D$, $a{:}C$, $(a,b){:}R$ or $(a,c){:}T$, it is of interest to compute α's best lower degree value bound. The *greatest lower bound* of α w.r.t. $\mathcal{K}$ (denoted $\mathit{glb}(\mathcal{K},\alpha)$) is $\mathit{glb}(\mathcal{K},\alpha) = \sup\{n \mid \mathcal{K} \models \langle \alpha \geqslant n\rangle\}$, where $\sup \emptyset = 0$. Determining the *glb* is called the *Best Degree Bound* (BDB) problem. Another similar concept is the *best satisfiability bound* of a concept C and amounts to determine $\mathit{glb}(\mathcal{K}, C) = \sup_{\mathcal{I}} \sup_{x\in\Delta^{\mathcal{I}}}\{C^{\mathcal{I}}(x) \mid \mathcal{I} \models \mathcal{K}\}$. Essentially, among all models $\mathcal{I}$ of the knowledge base, we are determining the maximal degree of truth that the concept C may have over all individuals $x \in \Delta^{\mathcal{I}}$.

Example 2.1 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

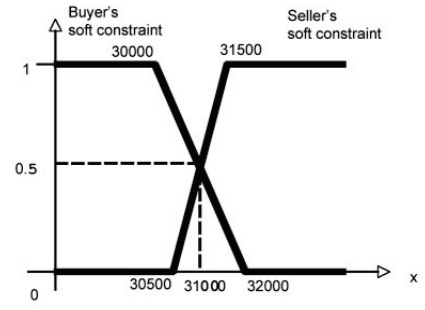


Figure 2: The soft price constraints.

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 \$32000. We may represent these statements by means of the axioms (see Figure 2): $\mathit{AudiTT} = \mathit{SportsCar} \sqcap \exists \mathit{hasPrice}.R(x; 30500, 31500)$ and $\mathit{Query} = \mathit{SportsCar} \sqcap \exists \mathit{hasPrice}.L(x; 30000, 32000)$, where *hasPrice* is a concrete feature (a car has only one price, which is a number). Then we may find out that the highest degree to which the concept $C = \mathit{AudiTT} \sqcap \mathit{Query}$ is satisfiable is 0.5 (the *possibility* that the Audi TT and the query matches is 0.5). That is, $glb(\mathcal{K}, C) = 0.5$ and corresponds to the point where both requests intersects (i.e., the car may be sold at \$31000).

3 Reasoning algorithm

Our procedure is inspired on [19]. We present here the method for reasoning in $\mathcal{ALCF}$ with GCI's under Łukasiewicz semantics: $\alpha \otimes \beta = \max\{\alpha + \beta - 1, 0\}, \alpha \oplus \beta = \min\{\alpha + \beta, 1\}, \ominus \alpha = 1 - \alpha$ and $\alpha \Rightarrow \beta = \min\{1, 1 - \alpha + \beta\}$. We leave out, for reasons of space, how to deal with fuzzy concrete domains and modifiers. The interested reader may consult [19]. We note however, that the procedure differs from the one presented in [19] in such a way that it will be better suited to deal with GCIs. Also note that $(\forall R.C)^{\mathcal{I}} = (\neg \exists R.\neg C)^{\mathcal{I}}$ and $(\exists R.C)^{\mathcal{I}} = (\neg \forall R.\neg C)^{\mathcal{I}}$. This allows 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 $\mathit{nnf}(C)$ we denote the NNF of concept C.

The basic idea of our reasoning algorithm is as follows. Consider $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$. In order to solve the BDB problem, we combine appropriate DL tableaux rules with methods developed in the context of *Many-Valued Logics* (MVLs) [3]. In order to determine e.g. $glb(\mathcal{K}, a{:}C)$, we consider an expression of the form $\langle a{:}\neg C, 1 - x \rangle$ (informally, $\langle a{:}C \leqslant x \rangle$), where x is a $[0, 1]$-valued variable. Then we construct a tableaux for $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \cup \{\langle a{:}\neg C, \ominus x\rangle\}\rangle$ in which the application of satisfiability preserving rules generates new fuzzy assertion axioms together with *inequations* over $[0,1]$-valued variables. These inequations have to hold in order to respect the semantics of the DL constructors. Finally, to determine the greatest lower bound, we *minimize* the original variable x such that all constraints are satisfied [1]. Similarly, for $C \sqsubseteq D$, we can compute $glb(\mathcal{K}, C \sqsubseteq D)$ as the minimal value of x such that $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \cup \{\langle a{:}C \sqcap \neg D, 1-x\rangle\}\rangle$ 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, $glb(\mathcal{K}, C)$ is determined by the maximal value of x such that $\langle \mathcal{T}, \mathcal{A} \cup \{\langle a{:}C, x\rangle\}\rangle$ is satisfiable.

In Łukasiewicz logic we end up with a *bounded Mixed Integer Linear Program* (bMILP) optimization problem [14]. 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 $\langle \alpha, l\rangle$, where l is a linear expression over variables in $\mathbf{V}, \mathbf{W}$ and real values.

Similar to crisp DLs, our tableaux 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 extends the tableau presented in [9], and is inspired by the one presented in [16, 17].

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

1. $\mathcal{L}(s, \bot) = 0$ and $\mathcal{L}(s, \top) = 1$ for all $s \in \mathbf{S}$,
2. If $\mathcal{L}(s, \neg A) \geqslant n$, then $\mathcal{L}(s, A) \leqslant \ominus n$.
3. If $\mathcal{L}(s, C \sqcap D) \geqslant n$, then $\mathcal{L}(s, C) \geqslant m_1$, $\mathcal{L}(s, D) \geqslant m_2$ and $n = m_1 \otimes m_2$, for some m_1 and m_2.
4. If $\mathcal{L}(s, C \sqcup D) \geqslant n$, then $\mathcal{L}(s, C) \geqslant m_1$, $\mathcal{L}(s, D) \geqslant m_2$ and $n = m_1 \oplus m_2$, for some m_1 and m_2.
5. If $\mathcal{L}(s, \forall R.C) \geqslant n$, then $\mathcal{E}(R, \langle s, t\rangle) \leqslant \mathcal{L}(t, C) + 1 - n$ for all $t \in \mathbf{S}$.
6. If $\mathcal{L}(s, \exists R.C) \geqslant n$, then there exists $t \in \mathbf{S}$ such that $\mathcal{E}(R, \langle s, t\rangle) \geqslant m_1$, $\mathcal{L}(t, C) \geqslant m_2$ and $n = m_1 \otimes m_2$, for some m_1 and m_2.
7. If $\langle C \sqsubseteq D, n\rangle \in \mathcal{T}$, then $\mathcal{L}(s, C) \leqslant \mathcal{L}(s, D) + 1 - n$, for all $s \in \mathbf{S}$.
8. If $\langle a{:}C, n\rangle \in \mathcal{A}$, then $\mathcal{L}(\mathcal{V}(a), C) \geqslant n$.

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

9. If $\langle (a,b){:}R, n\rangle \in \mathcal{A}$, then $\mathcal{E}(R, \langle \mathcal{V}(a), \mathcal{V}(b)\rangle) \geqslant n$.
10. If $\langle (a,c){:}T, n\rangle \in \mathcal{A}$, then $\mathcal{E}(T, \langle \mathcal{V}(a), \mathcal{V}(c)\rangle) \geqslant n$.

Proposition 3.1 *$\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ is satisfiable if and only if there exists a fuzzy tableau for $\mathcal{K}$.*

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

$$\begin{array}{l}\Delta^{\mathcal{I}} = \mathbf{S}, \quad a^{\mathcal{I}} = \mathcal{V}(a), \ a \text{ occurs in } \mathcal{A} \\ \top^{\mathcal{I}}(s) = \mathcal{L}(s, \top), \bot^{\mathcal{I}}(s) = \mathcal{L}(s, \bot), \text{ for all } s \in \mathbf{S} \\ A^{\mathcal{I}}(s) = \mathcal{L}(s, A), \text{ for all } s \in \mathbf{S} \\ R^{\mathcal{I}}(s,t) = \mathcal{E}(R, \langle s, t\rangle) \text{ for all } \langle s, t\rangle \in \mathbf{S} \times \mathbf{S}\end{array}$$

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

For the converse, we know from [5, 6] that if $\mathcal{K}$ has a model then it has a *witnessed model.* That is, $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a witnessed model of $\mathcal{K}$, if for all $x \in \Delta^{\mathcal{I}}$ there is $y \in \Delta^{\mathcal{I}}$ such that $(\exists R.C)^{\mathcal{I}}(x) = R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y)$ and there is $x \in \Delta^{\mathcal{I}}$ such that $(C \sqsubseteq D)^{\mathcal{I}} = C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x)$. So, let $\mathcal{I}$ be a witnessed model of $\mathcal{K}$. Then a fuzzy tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{V})$ for $\mathcal{K}$ can be defined as follows:

$$\mathbf{S}{=}\ \Delta^{\mathcal{I}},\ \mathcal{E}(R, \langle s,t\rangle) = R^{\mathcal{I}}(s,t),\ \mathcal{L}(s, C) = C^{\mathcal{I}}(s),\ \mathcal{V}(a) = a^{\mathcal{I}}$$

It can be verified that T is a fuzzy tableau for $\mathcal{K}$. □

Now, in order to decide the satisfiability of $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ a procedure that constructs a fuzzy tableau T for $\mathcal{K}$ has to be determined. Like the tableaux algorithm presented in [17], 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 $\mathcal{T}$, the termination of the algorithm has to be ensured. This is done by providing a *blocking* condition for rule applications.

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

Each node v is labelled with a sequence $\mathcal{L}(v)$ of expressions of the form $\langle C, l\rangle$, where $C \in sub(\mathcal{K})$, 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 $\mathcal{L}(\langle v, w\rangle)$ of expressions of the form $\langle R, l\rangle$, where $R \in \mathbf{R}_{\mathcal{K}}$ are roles occurring in $\mathcal{K}$(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 $\mathcal{C}_{\mathcal{F}}$ of constraints of the form $c \leqslant c', c = c', x_i \in [0,1], y_i \in \{0,1\}$, 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 $\mathcal{L}(\langle v, w\rangle)$, 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 *predecessor*.

We say that two non-root nodes v and w are *equivalent*, denoted $\mathcal{L}(v) \approx \mathcal{L}(w)$, if $\mathcal{L}(v) = [\langle C_1, l_1\rangle, \ldots, \langle C_n, l_k\rangle]$, $\mathcal{L}(w) = [\langle C_1, l'_1\rangle, \ldots, \langle C_n, l'_k\rangle]$, and for all $1 \leqslant i \leqslant k$, 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* if and only if none of its ancestors are blocked, it is not a root node, and it has an ancestor w such that $\mathcal{L}(v) \approx \mathcal{L}(w)$. In this case, we say w directly blocks v. A node v is *blocked* if and only if 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^i, for each individual a_i occurring in $\mathcal{A}$, labelled with $\mathcal{L}(v_0^i)$ such that $\mathcal{L}(v_0^i)$ contains $\langle C_i, n\rangle$ for each fuzzy assertion $\langle a_i{:}C_i, n\rangle \in \mathcal{A}$, and (ii) an edge $\langle v_0^i, v_0^j\rangle$, for each fuzzy assertion $\langle (a_i, a_j){:}R_i, n\rangle \in \mathcal{A}$, labelled with $\mathcal{L}(\langle v_0^i, v_0^j\rangle)$ such that $\mathcal{L}(\langle v_0^i, v_0^j\rangle)$ contains $\langle R_i, n\rangle$. $\mathcal{F}$ is then expanded by repeatedly applying the completion rules described below. The completion-forest is complete when none of the completion rules are applicable. Then, the bMILP problem on the set of constraints $\mathcal{C}_{\mathcal{F}}$ is solved.

We also need a technical definition involving feature roles (see [13]). Let $\mathcal{F}$ be forest, r an abstract feature such that we have two edges $\langle v, w_1\rangle$ and $\langle v, w_1\rangle$ such that $\langle r, l_1\rangle$ and $\langle r, l_2\rangle$ occur in $\mathcal{L}(\langle v, w_1\rangle)$ and $\mathcal{L}(\langle v, w_2\rangle)$, respectively (informally, $\mathcal{F}$ contains $\langle (v, w_1){:}r, l_1\rangle$ and $\langle (v, w_2){:}r, l_2\rangle$). 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 $\mathcal{L}(\langle v, w_2\rangle)$ to $\mathcal{L}(\langle v, w_1\rangle)$ and $\mathcal{L}(w_2)$ to $\mathcal{L}(w_1)$, and then deleting node w_2. 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_α we denote the variable associated to the *atomic assertion* α of the form $a{:}A$ or $(a, b){:}R$. 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:

(A). If $\langle A, l\rangle \in \mathcal{L}(v)$ then $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:A} \geqslant l\} \cup \{x_{v:A} \in [0, 1]\}$.

$(\bar{A})$. If $\langle \neg A, l\rangle \in \mathcal{L}(v)$ then $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:A} \leqslant 1 - l\} \cup \{x_{v:A} \in [0, 1]\}$.

(R). If $\langle R, l\rangle \in \mathcal{L}(\langle v, w\rangle)$ then $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{(v,w):R} \geqslant l\} \cup \{x_{(v,w):R} \in [0, 1]\}$.

$(\bot)$. If $\langle \bot, l\rangle \in \mathcal{L}(v)$ then $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{l = 0\}$.

$(\sqcap)$. If $\langle C \sqcap D, l\rangle \in \mathcal{L}(v)$ then append $\langle C, x_1\rangle$ and $\langle D, x_2\rangle$ to $\mathcal{L}(v)$, and $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{y \leqslant 1 - l, x_i \leqslant 1 - y, x_1 + x_2 = l + 1 - y, x_i \in [0, 1], y \in \{0, 1\}\}$, where x_i, y are new variables.

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

($\forall$). If $\langle \forall R.C, l_1\rangle \in \mathcal{L}(v)$, $\langle R, l_2\rangle \in \mathcal{L}(\langle v, w\rangle)$ and the rule has not been already applied to this pair then append $\langle C, x\rangle$ to $\mathcal{L}(w)$ and $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x \geqslant l_1 + l_2 - 1, x \leqslant y, l_1 + l_2 - 1 \leqslant y, l_1 + l_2 \geqslant y, x \in [0,1], y \in \{0,1\}\}$, where x, y are new variables. The case for concrete roles is similar.

($\sqsubseteq$). If $\langle C \sqsubseteq D, n\rangle \in \mathcal{T}$ and v is a node to which this rule has not yet been applied then append $\langle nnf(\neg C), 1 - x_1\rangle$ and $\langle D, x_2\rangle$ to $\mathcal{L}(v)$, and $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_1 \leqslant x_2 + 1 - n\}$.

($\exists$). If $\langle \exists R.C, l\rangle \in \mathcal{L}(v)$ and v is not blocked then create a new node w and append $\langle R, x_1\rangle$ to $\mathcal{L}(\langle v, w\rangle)$ and $\langle C, x_2\rangle$ to $\mathcal{L}(w)$, and $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{y \leqslant 1 - l, x_i \leqslant 1 - y, x_1 + x_2 = l + 1 - y, x_i \in [0,1], y \in \{0,1\}\}$, where x_i, y are new variables. The case for concrete roles is similar.

Let's see an example.

Example 3.1 *Consider $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$, where $\mathcal{T} = \{\langle \exists R.C \sqsubseteq D, 1\rangle\}$ and $\mathcal{A} = \{\langle (a,b) : R, 0.7\rangle, \langle b{:}C, 0.8\rangle\}$. Let us show that glb($\mathcal{K}, a{:}D$) = 0.5. To this end, we have to determine the minimal value for x such that $\langle \mathcal{T}, \mathcal{A} \cup \{\langle a{:}\neg D, 1 - x\rangle\}\rangle$ is satisfiable. To start with, we construct a forest $\mathcal{F}$ with two root nodes a and b (one for each individual in $\mathcal{A}$). We process first $\langle (a,b){:}R, 0.7\rangle$, then $\langle b{:}C, 0.8\rangle$ and finally $\langle a{:}\neg D, 1 - x\rangle$. Therefore, we set $\mathcal{L}(a) = [\langle C, 0.8\rangle, \langle \neg D, 1{-}x\rangle]$, $\mathcal{L}(\langle a, b\rangle) = [\langle R, 0.7\rangle]$ and $\mathcal{C}_\mathcal{F} = \{x \in [0,1]\}$.*

We first process $\langle R, 0.7\rangle \in \mathcal{L}(\langle a, b\rangle)$, apply rule ($R$) and, thus, add $x_{(a,b):R} \geqslant 0.7$ and $x_{(a,b):R} \in [0,1]$ to $\mathcal{C}_\mathcal{F}$. Then we process $\langle C, 0.8\rangle \in \mathcal{L}(a)$, apply rule ($A$) and, thus, add $x_{a:C} \geqslant 0.8$ and $x_{a:C} \in [0,1]$ to $\mathcal{C}_\mathcal{F}$. We next process $\langle \neg D, 1 - x\rangle \in \mathcal{L}(a)$, apply rule ($A$) and, thus, add $x_{a:D} \leqslant x$ and $x_{a:D} \in [0,1]$ to $\mathcal{C}_\mathcal{F}$.

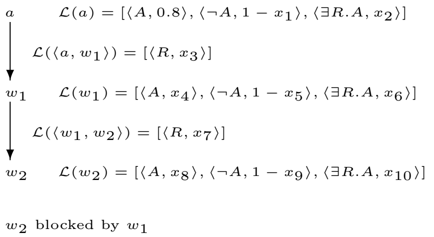
Now, we apply ($\sqsubseteq$) to a and $\langle \exists R.C \sqsubseteq D, 1\rangle$, and, thus, we append $\langle \forall R.\neg C, 1 - x_1\rangle$ and $\langle D, x_2\rangle$ to $\mathcal{L}(a)$, and we add $x_1 \leqslant x_2$ and $x_1, x_2 \in [0,1]$ to $\mathcal{C}_\mathcal{F}$. Similarly, we apply ($\sqsubseteq$) to b: we append $\langle \forall R.\neg C, 1 - x_3\rangle$ and $\langle D, x_4\rangle$ to $\mathcal{L}(b)$, and we add $x_3 \leqslant x_4$ and $x_3, x_4 \in [0,1]$ to $\mathcal{C}_\mathcal{F}$.

Next we process $\langle \forall R.\neg C, 1 - x_1\rangle \in \mathcal{L}(a)$, apply rule ($\forall$) to it and $\langle R, 0.7\rangle \in \mathcal{L}(\langle a, b\rangle)$ and, thus, we append $\langle \neg C, x_5\rangle$ to $\mathcal{L}(b)$, and we add $\{x_5 \geqslant 1 - x_1 + 0.7 - 1, x_5 \leqslant y, 1 - x_1 - 0.3 \leqslant y, 1 - x_1 + 0.7 \geqslant y, x_5 \in [0,1], y \in \{0,1\}\}$ to $\mathcal{C}_\mathcal{F}$. We next process $\langle D, x_2\rangle \in \mathcal{L}(a)$, apply rule ($A$) and, thus, add $x_{a:D} \geqslant x_2$ and $x_{a:D} \in [0,1]$ to $\mathcal{C}_\mathcal{F}$. Next we process $\langle \forall R.\neg C, 1 - x_1\rangle \in \mathcal{L}(b)$, but no rule is applicable to it. Finally, we process $\langle D, x_4\rangle \in \mathcal{L}(b)$, apply rule ($A$) and, thus, add $x_{b:D} \geqslant x_4$ and $x_{b:D} \in [0,1]$ to $\mathcal{C}_\mathcal{F}$.

Now the forest $\mathcal{F}$ is complete as no more rule is applicable and we consider the set of inequations $\mathcal{C}_\mathcal{F}$. It remains to solve the bMILP problem on $\mathcal{C}_\mathcal{F}$. Indeed, it holds that glb($\mathcal{K}, a{:}D$) = $\min x.\mathcal{C}_\mathcal{F}$. 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.

Example 3.2 *Consider $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$, where $\mathcal{T} = \{\langle A \sqsubseteq \exists R.A, 1\rangle\}$ and $\mathcal{A} = \{\langle a{:}A, 0.8\rangle\}$. Let's show that $\mathcal{K}$ is satisfiable. The complete forest $\mathcal{F}$ in Figure 3 shows the computation.*


a
$\mathcal{L}(a) = [\langle A, 0.8\rangle, \langle \neg A, 1 - x_1\rangle, \langle \exists R.A, x_2\rangle]$
$\mathcal{L}(\langle a, w_1\rangle) = [\langle R, x_3\rangle]$
w_1
$\mathcal{L}(w_1) = [\langle A, x_4\rangle, \langle \neg A, 1 - x_5\rangle, \langle \exists R.A, x_6\rangle]$
$\mathcal{L}(\langle w_1, w_2\rangle) = [\langle R, x_7\rangle]$
w_2
$\mathcal{L}(w_2) = [\langle A, x_8\rangle, \langle \neg A, 1 - x_9\rangle, \langle \exists R.A, x_{10}\rangle]$
w_2 blocked by w_1

$$\begin{aligned}\mathcal{C}_{\mathcal{F}} = \{&x_i \in [0,1], y_i \in \{0,1\}, x_{a:A} \geqslant 0.8, x_1 \leqslant x_2,\\ &x_{a:A} \geqslant x_1, y_1 \leqslant 1 - x_2, x_3 \leqslant 1 - y_1, x_4 \leqslant 1 - y_1,\\ &x_3 + x_4 = x_2 + 1 - y_1, x_{(a,\, w_1):R} \geqslant x_3, x_{w_1:A} \geqslant x_4,\\ &x_5 \leqslant x_6, x_{w_1:A} \leqslant x_5, y_2 \leqslant 1 - x_6, x_7 \leqslant 1 - y_2,\\ &x_8 \leqslant 1 - y_2, x_7 + x_8 = x_6 + 1 - y_2,\\ &x_{(w_1,\, w_2):R} \geqslant x_7, x_{w_2:A} \geqslant x_8, x_9 \leqslant x_{10}, x_{w_2:A} \leqslant x_9\}\end{aligned}$$

Figure 3: Complete forest for Example 3.2.

As we can notice, node w_2 is blocked by node w_1, meaning that w_1 and w_2 share the same properties. In order to build a model, we replace all occurrences of w_2 in $\mathcal{C}_{\mathcal{F}}$ with w_1 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 $\mathcal{K}$, the tableau algorithm terminates.*

Proof: [Sketch] Termination is a result of the properties of the expansion rules, as in the classical case [9]. 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 $sub(\mathcal{K})$, 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. □

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

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

$$\begin{aligned}\mathbf{S} &= \{v \mid v \text{ is a node in } \mathcal{F}, \text{ and } v \text{ is not blocked}\},\\ \mathcal{L}(v, \bot) &= 0, \text{ if } v \in \mathbf{S},\\ \mathcal{L}(v, \top) &= 1, \text{ if } v \in \mathbf{S},\\ \mathcal{L}(v, A) &= \bar{x}_{v:A}, \text{ if } v \text{ in } \mathcal{F} \text{ not blocked},\\ \mathcal{E}(R, \langle v, w\rangle) &= \bar{x}_{(v,w):R}, \text{ if } v, w \text{ in } \mathcal{F} \text{ not blocked}\\ \mathcal{E}(R, \langle v, w\rangle) &= \bar{x}_{(v,w'):R}, \text{ if } v \text{ in } \mathcal{F} \text{ not blocked, } w \text{ blocks } w'\\ \mathcal{V}(a_i) &= v_0^i, \text{ where } v_0^i \text{ is a root node}\end{aligned}$$

It can be shown that T is a fuzzy tableau for $\mathcal{K}$. □

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

Proof: [Sketch] Let $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{V})$ be a fuzzy tableau for $\mathcal{K}$. Using T, we can trigger the application of the expansion rules such that they yield a completion-forest $\mathcal{F}$ that is complete. Using $\mathcal{L}$ and $\mathcal{E}$ we can find a solution to $\mathcal{C}_{\mathcal{F}}$. □

4 Conclusions

In this work we presented a reasoning algorithm for $\mathcal{ALCF}(D)$ with general concept inclusions and explicit membership functions ($\mathcal{ALCF}$ is a guarded logic [2]), under Łukasiewicz semantics. Clearly, by using [19] the result applies also to the usually used "Zadeh semantics" of fuzzy DLs, and of course also for the classical boolean variant of $\mathcal{ALCF}$.

The result can be extended to more expressive fuzzy DLs, such as $\mathcal{SHIF}(D)$ and $\mathcal{SHOIN}(D)$, which are the DLs behind the web ontology description languages OWL-DL and OWL-Lite, by adapting our blocking condition similarly as done in [8]. The description of these blocking conditions will be the subject of an extend paper. So far, an implementation for fuzzy $\mathcal{SHIF}(D)$ can be found from Straccia's home page.

Acknowledgment

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

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] E. Grädel, C. Hirsch and M. Otto. Back and forth between guarded and modal logics. *ACM Transactions on Computational Logic*, 3(3):418–463, 2002.

[3] R. Hähnle. Advanced many-valued logics. In D. M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic, 2nd Edition*, volume 2. Kluwer, 2001.

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

[5] P. Hájek. Making fuzzy description logics more general. *Fuzzy Sets and Systems*, 154 (1), 1–15, 2005.

[6] P. Hájek. What does mathematical fuzzy logic offer to description logic? In Elie Sanchez, editor, *Capturing Intelligence: Fuzzy Logic and the Semantic Web*. Elsevier, 2006.

[7] S. Hölldobler, H.-P. Störr and T. D. Khang. The fuzzy description logic ALC_{FH} with hedge algebras as concept modifiers. *Journal of Advanced Computational Intelligence*, 7(3):294–305, 2003.

[8] I. Horrocks, U. Sattler and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.

[9] I. Horrocks, U. Sattler and S. Tobies. Reasoning with individuals for the description logic $\mathcal{SHIQ}$. In D. MacAllester, editor, *Proceedings of the 17th International Conference on Automated Deduction (CADE-17)*, volume 1831 in Lecture Notes in Artificial Intelligence, pages 482–496, 2000. Springer Verlag.

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

[11] Y. Li, B. Xu, J. Lu and D. Kang. Discrete tableau algorithms for SHI. In *Proceeedings of the International Workshop on Description Logics (DL-06)*. CEUR, 2006.

[12] T. Lukasiewicz, U. Straccia. An overview of uncertainty and vagueness in description logics for the semantic web. Technical Report INFSYS RR-1843-06-07, Institut für Informationssysteme, Technische Universität Wien (2006)

[13] C. Lutz. Description logics with concrete domains - A survey. In *Advances in Modal Logics Volume 4*. King's College Publications, 2003.

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

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

[16] G. Stoilos, G. Stamou, V. Tzouvaras, J. Z. Pan and Ian Horrocks. A Fuzzy Description Logic for Multimedia Knowledge Representation. In *Proceedings of the International Workshop on Multimedia and the Semantic Web*, 2005.

[17] G. Stoilos, U. Straccia, G. Stamou and J. Z. Pan. General concept inclusions in fuzzy description logics. In *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI-06)*, pages 457–461. IOS Press, 2006.

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

[19] U. Straccia. Description logics with fuzzy concrete domains. In F. Bachus and T. Jaakkola, editors, *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pages 559–567, 2005. AUAI Press.

[20] 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.

[21] C. Tresp and R. Molitor. A description logic for vague knowledge. In *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, 1998.

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