A Fuzzy Description Logic with Product T-norm

Fernando Bobillo and Umberto Straccia

***Abstract*—*Fuzzy Description Logics* (fuzzy DLs) have been proposed as a language to describe structured knowledge with vague concepts. It is well known that the choice of the fuzzy operators may determine some logical properties. However, up to date the study of fuzzy DLs has been restricted to the Łukasiewicz logic and the "Zadeh semantics". In this work, we propose a novel semantics combining the common product t-norm with the standard negation. We show some interesting properties of the logic and propose a reasoning algorithm based on a mixture of tableaux rules and the reduction to Mixed Integer Quadratically Constrained Programming.**

I. 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 [7] for details).

Fuzzy DLs (see [9] for a survey) 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 [18], 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.

From a semantics point of view, most works rely on the semantics of fuzzy sets operators proposed by Zadeh [18] (which we call here "Zadeh semantics"): Gödel conjunction and disjunction ($x \wedge y = \min(x, y)$, $x \vee y = \max(x, y)$), Łukasiewicz negation ($\neg x = 1 - x$) and Kleene-Diene implication ($x \rightarrow y = \max\{1 - x, y\}$). Some few works consider a fully Łukasiewcz compliant semantics [4], [3], [16]. [16] 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) and proposes a reasoning solution, which is based on a mixture of tableau rules and Mixed Integer Linear Programming (MILP) and is implemented in the *fuzzyDL* reasoner, available from Straccia's home page.

In this paper, we consider a fuzzy DL with a novel semantics, based on the product as t-norm. To the best of our knowledge, this is the first attempt in this direction. In the remainder, we proceed as follows. In the next section we describe the basics of fuzzy DLs and our product-based semantics, then in Section III we address the inference algorithm and finally Section IV presents some conclusions and ideas for future work.

II. FUZZY DLS

In this section, we define fuzzy $\mathcal{ALCF}(D)$. We recall here the semantics given in [4], [3], [16], [17] and then we consider a family of operators with product as t-norm.

A. 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 $\mathcal{ALC}$ [11] extended with functional roles (also called attributes or features) and concrete domains [10] 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 $\mathbb{Q}^+ \cup \{0\}$ Using these functions, we may then define, for instance, $Young \colon Natural \rightarrow [0, 1]$ 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 also 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 $f_m \colon [0, 1] \rightarrow [0, 1]$. For instance, we may define $very(x) = x^2$ and $slightly(x) = \sqrt{x}$.

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

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

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

1-4244-1210-2/07/$25.00 ©2007 IEEE.

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:

$$\begin{array}{rcl} C & := & \top \mid \bot \mid A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid m(C) \mid \\ & & \forall R.C \mid \exists R.C \mid \forall T_1, \ldots, T_n.D \mid \exists T_1, \ldots, T_n.D \, . \end{array}$$

Concerning axioms and assertions, similarly to [17], we define fuzzy axioms as follows.

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. 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., a minor,

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 $\mathit{YoungPerson}$ with degree at least 0.2, while $\langle (\mathit{jim}, \mathit{180}){:}\mathit{hasHeight}, 1 \rangle$, where $\mathit{hasHeight}$ 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.

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 $\mathcal{A}$.

B. Semantics

The semantics extends [15]. The main idea is that concepts and roles are interpreted as fuzzy subsets of an interpretation's domain. Therefore, $\mathcal{ALCF}(\mathbf{D})$ 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 $\wedge, \vee, \neg$ and $\rightarrow$ in infix notation, in place of a t-norm, s-norm, negation function and implication function. 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 $\Delta^{\mathcal{I}}$ (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 $C^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \rightarrow [0, 1]$; (ii) to each abstract role R a function $R^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0, 1]$; (iii) to each abstract feature r a partial function $r^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0, 1]$ 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 concrete role T a function $R^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}} \rightarrow [0, 1]$; (v) to each concrete feature t a partial function $t^{\mathcal{I}} \colon \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}} \rightarrow [0, 1]$ 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 $f_m \colon [0, 1] \rightarrow [0, 1]$; (vii) to each abstract individual a an element in $\Delta^{\mathcal{I}}$; $(viii)$ 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 I (where $x, y \in \Delta^{\mathcal{I}}$ and $v \in \Delta_{\mathbf{D}}$).

We comment briefly some points. The semantics of $\exists R.C$ $(\exists R.C)^{\mathcal{I}}(d) = \sup_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x, y) \wedge C^{\mathcal{I}}(y)$ is the result of viewing $\exists R.C$ as the open first order formula $\exists y.F_R(x, y) \wedge F_C(y)$ (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 $\forall y.F_R(x, y) \rightarrow F_C(y)$.

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, iff $\alpha^{\mathcal{I}} \geqslant n$. Similarly, $\mathcal{I} \models \langle \alpha \geqslant n \rangle$, where α is a concept or a role assertion axiom, iff $\alpha^{\mathcal{I}} \geqslant n$. We say that a concept C is *satisfiable* iff there is an interpretation $\mathcal{I}$ and an individual $x \in \Delta^{\mathcal{I}}$ such that $C^{\mathcal{I}}(x) > 0$.

For a set of fuzzy axioms $\mathcal{E}$, we say that $\mathcal{I}$ *satisfies* $\mathcal{E}$ iff I satisfies each element in $\mathcal{E}$. We say that $\mathcal{I}$ is a *model* of E (resp. $\mathcal{E}$) iff $\mathcal{I} \models E$ (resp. $\mathcal{I} \models \mathcal{E}$). $\mathcal{I}$ *satisfies* (is a *model* of) a fuzzy 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 $\mathcal{K}$, denoted $\mathcal{K} \models E$ iff 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.

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

$$
\begin{array}{rcl}
\bot^{\mathcal{I}}(x) & = & 0 \\
\top^{\mathcal{I}}(x) & = & 1 \\
(C_1 \sqcap C_2)^{\mathcal{I}}(x) & = & {C_1}^{\mathcal{I}}(x) \wedge {C_2}^{\mathcal{I}}(x) \\
(C_1 \sqcup C_2)^{\mathcal{I}}(x) & = & {C_1}^{\mathcal{I}}(x) \vee {C_2}^{\mathcal{I}}(x) \\
(\neg C)^{\mathcal{I}}(x) & = & \neg C^{\mathcal{I}}(x)) \\
(m(C))^{\mathcal{I}}(x) & = & f_m(C^{\mathcal{I}}(x)) \\
(\forall R.C)^{\mathcal{I}}(x) & = & \inf_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,y) \to C^{\mathcal{I}}(y) \\
(\exists R.C)^{\mathcal{I}}(x) & = & \sup_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(x,y) \wedge C^{\mathcal{I}}(y) \\
(\forall T_1, \ldots, T_n.D)^{\mathcal{I}}(x) & = & \inf_{y_1,\ldots,y_n \in \Delta_{\mathbf{D}}} (\bigwedge_{i=1}^{n} {T_i}^{\mathcal{I}}(x,y_i)) \to D^{\mathcal{I}}(y_1,\ldots,y_n) \\
(\exists T_1, \ldots, T_n.D)^{\mathcal{I}}(x) & = & \sup_{y_1,\ldots,y_n \in \Delta_{\mathbf{D}}} (\bigwedge_{i=1}^{n} {T_i}^{\mathcal{I}}(x,y_i)) \wedge D^{\mathcal{I}}(y_1,\ldots,y_n) \,.
\end{array}
$$

TABLE I

Fuzzy DL semantics.

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 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 using two axioms:

$$
\begin{array}{lcl}
\mathit{AudiTT} & = & \mathit{SportsCar} \sqcap \exists \mathit{hasPrice}.R(x; 30500, 31500) \\
\mathit{Query} & = & \mathit{SportsCar} \sqcap \exists \mathit{hasPrice}.L(x; 30000, 32000)
\end{array}
$$

where $\mathit{hasPrice}$ 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, $\mathit{glb}(\mathcal{K}, C) = 0.5$ and corresponds to the point where both requests intersects (i.e., the car may be sold at \$31000).

C. Product as t-norm

In the rest of the paper, we will concentrate on the novel semantics given by the following operators:

$$
\begin{array}{rcl}
\neg\alpha & = & 1 - \alpha \\
\alpha \wedge \beta & = & \alpha \cdot \beta \\
\alpha \vee \beta & = & \alpha + \beta - \alpha \cdot \beta \\
\alpha \to \beta & = & \left\{ \begin{array}{ll} 1 & \text{if } \alpha \leqslant \beta \\ \beta/\alpha & \text{if } \alpha > \beta \end{array} \right.
\end{array}
$$

These operators correspond to the product logic family but replacing Gödel negation ($\neg\alpha = 1$ if $\alpha = 0$ or 0 otherwise) with Łukasiewicz negation. We follow the inspiration of the probabilistic theory, which combines this negation (in the probability of the negated event) and the product (in the probability of the conjunction of independent events) [8]. We are doubtful about the practical interest of Gödel negation and we prefer to use a continuous and involutive negation.

A property of our logic is that $\neg\forall R.C \neq \exists R.(\neg C)$ and $\neg\exists R.C \neq \forall R.(\neg C)$. Note that the equality holds for Łukasiewicz logic and "Zadeh semantics". This is interesting since assuming the inter-definability of quantifiers is generally an unnecessary restriction [4]. However, De Morgan laws are still verified, i.e. $\neg(\alpha \vee \beta) = \neg\alpha \wedge \neg\beta$ and $\neg(\alpha \wedge \beta) = \neg\alpha \vee \neg\beta$. Moreover, product t-norm is subidempotent and product t-conorm is superidempotent. Hence, in general $C \neq C \sqcap C$ and $C \neq C \sqcup C$.

Finally, we note that [3], [4] shows for Łukasiewicz logic that if $\mathcal{K}$ has a model then it has a *witnessed model*, and that this does not hold for any logic with Gödel negation (such as Gödel or product logics). However, we note that this result holds for product logic with Łukasiewicz negation as well.

III. Reasoning

Our procedure is inspired on [16]. We leave out for reasons of space, how to deal with fuzzy concrete domains and modifiers. The basic idea behind 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) [2]. In order to determine e.g. $\mathit{glb}(\mathcal{K}, a{:}C)$, 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 $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \cup \{\langle a{:}\neg C, \neg x \rangle\} \rangle$ 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, in order 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 $\mathit{glb}(\mathcal{K}, C \sqsubseteq D)$ as the minimal value of x such that $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \cup \{\langle a : C, x_1 \rangle\} \cup \{\langle a : D \geqslant x_2 \rangle\} \rangle$ is satisfiable under the constraints $\{y \cdot x_1 + (1-y) \cdot x_2 \geqslant y \cdot x_2 + (1-y) \cdot x_1, x \geqslant (1-y), y \cdot x_2 \leqslant x \cdot x_1, x_i \in [0,1], y \in \{0,1\}\}$, where a is new abstract

[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 $\mathit{glb}(\mathcal{K}, (a{:}C)) = \bar{n}$

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.

Using this logic, we end up with a bounded Mixed Integer Quadratically Constrained Programming optimization problem (bMICQP, see below). We recall that under "Zadeh semantics" and Łukasiewicz logic we end up with a bounded Mixed Integer Linear Program (bMILP) problem [16]. Interestingly, the tableaux contains only one branch and only and, thus, just one bMICQP problem has to be solved.

A. bMIQCP problems

Let $x = \langle x_1, \ldots, x_k\rangle$ and $y = \langle y_1, \ldots, y_m\rangle$ be variables over $\mathbb{Q}$, over the integers (the variables in y are called control variables). For all $i \in 0, 1, \cdots, n$, let a_i be an integer vector of length k, b_i be an integer vector of length m, h_i be an integer number and $Q_i(x, y) = 1/2 \cdot (x + y)^T \cdot C_i \cdot (x + y)$, where C_i is a symmetric integer matrix of dimension $(k + m) \times (k + m)$. Let $f(x, y)$ be an $k + m$-ary linear function. The MICQP problem is to find $\bar{x} \in \mathbb{Q}^k, \bar{y} \in \mathbb{Z}^m$ such that $f(\bar{x}, \bar{y}) = \min\{f(x, y) : a_0 \cdot x + b_0 \cdot y \geqslant h_0\}$ subject to a set of n constraints of the form: $a_i \cdot x + b_i \cdot y + Q_i(x, y) \geqslant h_i$ or $a_i \cdot x + b_i \cdot y + Q_i(x, y) \leqslant h_i$, for all $i = 1, \ldots, n$. Notice that the objective function is linear, while the restrictions can contain quadratic sections. The general case can be restricted to what concerns the paper as we can deal with bounded MIQCP (bMIQCP). That is, the rational variables range over $[0, 1]$, while the integer variables ranges over $\{0, 1\}$.

Furthermore, we say that $M \subseteq [0, 1]^k$ is bMICQP-representable iff there is a bMICQP (a_i, b_i, C_i, h_i) with k real and m $0-1$ variables such that $M = \{x : \exists y \in \{0, 1\}^m\}$ such that $a_i \cdot x + b_i \cdot y + Q_i(x, y) \geqslant h_i$ or $a_i \cdot x + b_i \cdot y + Q_i(x, y) \leqslant h_i$. Notice that every constructor of the logic is bMICQP representable i.e. it generates a set of bMICQP representable constraints.

B. A fuzzy tableau

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 can be seen as an extension of the tableau presented in [6], and is inspired by the one presented in [13], [14].

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 $\mathcal{L}(s, A) \leqslant \neg n$.
3) If $\mathcal{L}(s, \neg\neg C) \geqslant n$, then $\mathcal{L}(s, C) \geqslant n$.
4) 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 $n = m_1 \wedge m_2$, for some m_1 and m_2.
5) If $\mathcal{L}(s, \neg(C \sqcap D)) \geqslant n$, then $\mathcal{L}(s, \neg C \sqcup \neg D) \geqslant n$.
6) 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 $n = m_1 \vee m_2$, for some m_1 and m_2.
7) If $\mathcal{L}(s, \neg(C \sqcup D)) \geqslant n$, then $\mathcal{L}(s, \neg C \sqcap \neg D) \geqslant n$.
8) If $\mathcal{L}(s, \forall R.C) \geqslant n$, then $\mathcal{L}(t, C) \geqslant \mathcal{E}(R, \langle s, t\rangle) \wedge n$, for all $t \in \mathbf{S}$.
9) If $\mathcal{L}(s, \neg\forall R.C) \geqslant n$, then there exists $t \in \mathbf{S}$ such that $\mathcal{E}(R, \langle s, t\rangle) \to \mathcal{L}(t, C) \leqslant 1 - n$.
10) 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 $n = m_1 \wedge m_2$, for some m_1 and m_2.
11) If $\mathcal{L}(s, \neg\exists R.C) \geqslant n$, then $\mathcal{E}(R, \langle s, t\rangle) \wedge \mathcal{L}(t, C) \leqslant 1 - n$, for all $t \in \mathbf{S}$.
12) If $\langle C \sqsubseteq D, n\rangle \in \mathcal{T}$, then $\mathcal{L}(s, D) \geqslant \mathcal{L}(s, C) \wedge n$, for all $s \in \mathbf{S}$.
13) If $\langle a{:}C, n\rangle \in \mathcal{A}$, then $\mathcal{L}(\mathcal{V}(a), C) \geqslant n$.
14) If $\langle (a, b){:}R, n\rangle \in \mathcal{A}$, then $\mathcal{E}(R, \langle \mathcal{V}(a), \mathcal{V}(b)\rangle) \geqslant n$.
15) 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 iff 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 13–15, this implies that $\mathcal{I}$ is a model of $\mathcal{T}$, and that it satisfies each fuzzy assertion in $\mathcal{A}$.

For the converse, in our logic it holds 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 $(\exists R.C)^\mathcal{I}(x) = R^\mathcal{I}(x, y) \wedge C^\mathcal{I}(y)$ and there is $x \in \Delta^\mathcal{I}$ such that $(C \sqsubseteq D)^\mathcal{I} = C^\mathcal{I}(x) \to D^\mathcal{I}(x)$.
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 $\mathcal{K}$. □

C. An algorithm for building a fuzzy tableau

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 [14], our algorithm works on *completion-forests* since an ABox might contain several individuals with arbitrary roles connecting them. It is worth to note that, while reasoning algorithms within DLs usually transform concept expressions into a semantically equivalent *Negation Normal Form* or NNF (which is obtained by pushing in the usual manner negation on front of concept names, modifiers and concrete predicate names only), we cannot make this assumption now since $\neg\forall R.C \neq \exists R.(\neg C)$ and $\neg\exists R.C \neq \forall R.(\neg C)$.

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 $l \leqslant l', l = l', x_i \in [0,1], y_i \in \{0,1\}$, on the variables occurring the node labels and edge labels. l, l' 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*.

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. 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_j are variables, or both l_i and l_j 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 $\mathcal{L}(v) \approx \mathcal{L}(w)$. 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^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 bMIQCP problem on the set of constraints $\mathcal{C}_{\mathcal{F}}$ is solved.

We also need a technical definition involving feature roles (see [10]). 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 rules for $(\exists)$ and $(\neg\forall)$ are applied as last. Also, all expressions in node labels are processed according to the order they are introduced into $\mathcal{F}$.

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

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

$(\sqcap)$. 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 $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_1 \cdot x_2 = l, x_i \in [0,1]\}$, where x_i are new variables.

$(\bar{\sqcap})$. If $\langle \neg(C \sqcap D), l \rangle \in \mathcal{L}(v)$ then append $\langle \neg C \sqcup \neg D, l \rangle$ to $\mathcal{L}(v)$.

$(\sqcup)$. 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 $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_1 + x_2 - x_1 \cdot x_2 = l, x_i \in [0,1]\}$, where x_i are new variables.

$(\bar{\sqcup})$. If $\langle \neg(C \sqcup D), l \rangle \in \mathcal{L}(v)$ then append $\langle \neg C \sqcap \neg D, l \rangle$ to $\mathcal{L}(v)$.

$(\forall)$. 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 $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x \geqslant l_1 \cdot l_2, x \in [0,1]\}$, where x is a new variable. The case for concrete roles is similar.

$(\bar{\exists})$. If $\langle \neg\exists R.C, l_1 \rangle \in \mathcal{L}(v)$, $\langle R, l_2 \rangle \in \mathcal{L}(\langle v, w \rangle$ and the rule has not been already applied to this pair then append $\langle \neg C, 1 - x_1 \rangle$ to $\mathcal{L}(w)$ and $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{(v,w):R} \leqslant x_2, x_1 \cdot x_2 = 1 - l_1, x_i \in [0,1]\}$, where x_i 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 $\langle \neg C, 1 - x_1 \rangle$ and $\langle D, x_2 \rangle$ to $\mathcal{L}(v)$, and $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_2 \geqslant x_1 \cdot n, x_i \in [0,1]\}$, where x_i are new variables.

$(\exists)$. 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 append $\langle C, x_2 \rangle$ to $\mathcal{L}(w)$, and $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_1 \cdot x_2 = l, x_i \in [0,1]\}$, where x_i are new variables. The case for concrete roles is similar.

$(\bar{\forall})$. If $\langle \neg\forall R.C, l \rangle \in \mathcal{L}(v)$ 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 append $\langle C, x_2 \rangle$ to $\mathcal{L}(w)$ and $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{y \cdot x_1 + (1-y) \cdot x_2 \geqslant y \cdot x_2 + (1-y) \cdot x_1, l \leqslant y, y \cdot x_2 \leqslant x_1 - l \cdot x_1, x_i \in [0,1], y \in \{0,1\}\}$, where x_i, y are new variables. The case for concrete roles is similar.

Let us comment the $(\bar{\forall})$-rule. If $\langle \neg\forall R.C, l \rangle \in \mathcal{L}(v)$ then

$\inf_b R(a,b) \rightarrow C(b) \leqslant 1 - l$. For a new abstract individual b, there are two possibilities to satisfy this restriction: (i) $R^{\mathcal{I}}(a,b) \leqslant C^{\mathcal{I}}(b)$ and $l = 0$; (ii) $R^{\mathcal{I}}(a,b) \geqslant C^{\mathcal{I}}(b)$ and $C^{\mathcal{I}}(b) \leqslant R^{\mathcal{I}}(a,b) - R^{\mathcal{I}}(a,b) \cdot l$. The binary variable y simulates the two branches: *(i)* if $y = 0$ then $l = 0$ and $x_2 \geqslant x_1$; *(ii)* if $y = 1$ then $x_2 \leqslant x_1$ and $x_2 \leqslant x_1 - x_1 \cdot l$.

Rules for modifiers and concrete roles are not shown since they are equal to "Zadeh semantics" and Łukasiewicz logics [16]. However, as explained above, we cannot suppose in them that concepts are in NNF.

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 [6]. 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 rules $(\exists)$ and $(\neg\forall)$. 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:

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

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 $\mathcal{C}_{\mathcal{F}}$. □

IV. Conclusions

In this work we presented a reasoning algorithm for fuzzy $\mathcal{ALCF}(D)$ with product as t-norm. Our logic has interesting properties such as not allowing inter-definability of quantifiers, but still verifying De Morgan laws. The reasoning algorithm combines tableaux rules with a reduction to a bMIQCP problem (while under Łukasiewicz logic and "Zadeh semantics" we end up with a bMILP problem).

The result could 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 [5]. We also plan to implement the reasoning algorithm as an extension of the *fuzzyDL* reasoner.

Acknowledgment

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

References

[1] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

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

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

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

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

[6] Ian Horrocks, Ulrike Sattler and Stephan Tobies. Reasoning with individuals for the description logic SHIQ. In David MacAllester, editor, *Proceedings of the 17th International Conference on Automated Deduction (CADE-17)*, number 1831 in Lecture Notes in Artificial Intelligence, pages 482–496, Germany, 2000. Springer Verlag.

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

[8] Thomas Lukasiewicz. *Probabilistic Description Logics for the Semantic Web*. Technical Report INFSYS 1843-06-05, Institut für Informationssysteme, Technische Universität, Vienna, Austria, 2006.

[9] Thomas Lukasiewicz and Umberto 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, Vienna, Austria, 2006.

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

[11] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991.

[12] Giorgos Stoilos, Giorgos Stamou, Vassilis Tzouvaras, Jeff Z. Pan and Ian Horrocks. The fuzzy description logic f-SHIN. *Proc. of the ISWC Workshop on Uncertainty Reasoning for the Semantic Web*, 2005.

[13] Giorgos Stoilos, Giorgos Stamou, Vassilis Tzouvaras, Jeff 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.

[14] Giorgos Stoilos, Umberto Straccia, Giorgos Stamou and Jeff 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.

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

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

[17] Umberto Straccia. A fuzzy description logic for the semantic web. In Elie Sanchez, editor, *Fuzzy Logic and the Semantic Web*, Capturing Intelligence, chapter 4, pages 73–90. Elsevier, 2006.

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