Fuzzy Descriptions Logics with Fuzzy Truth Values

Fernando Bobillo[1] Umberto Straccia[2]

1. Dpt. of Computer Science & Systems Engineering University of Zaragoza, Spain
2. ISTI - CNR, Pisa, Italy
Email: fbobillo@unizar.es, straccia@isti.cnr.it

***Abstract*— *Fuzzy Description Logics are a family of logics which allow to deal with structured knowledge affected by vagueness. Although a relatively important amount of work has been carried out in the last years, current fuzzy DLs are open to be extend with several features worked out in the fuzzy logic literature. In this work, we extend fuzzy DLs with fuzzy truth values, allowing to state sentences such as "Tina is young is* almost *true".**

***Keywords*—** Fuzzy Description Logics, Fuzzy Truth Values

1 Introduction

Description Logics (DLs) [1], have gained popularity due to their application in the context of the *Semantic Web* [2]. *Ontologies* play a key role in the Semantic Web. An ontology consists of a hierarchical description of important concepts in a particular domain, along with the description of the properties (of the instances) of each concept. DLs play a particular role in this context as they are essentially the theoretical counterpart of the *Web Ontology Language OWL DL*, a state of the art language to specify ontologies.

It is well-known that "classical" ontology languages are not appropriate to deal with *fuzzy/vague/imprecise knowledge*, which is inherent to several real world domains [3, 4]. Since fuzzy set theory and fuzzy logic are suitable formalisms to handle these types of knowledge, fuzzy ontologies emerge as useful in several applications, such as (multimedia) information retrieval, image interpretation, ontology mapping, matchmaking and the Semantic Web [5]. So far, several fuzzy extensions of DLs can be found in the literature (see the survey in [5]) and some fuzzy DL reasoners exist, such as FUZZYDL [6], DELOREAN [7], FIRE [8] or DLMEDIA [9].

In this paper we allow fuzzy DL sentences to be qualified with *fuzzy truth values* [10], and, thus, allow expressions such as "Tina is young is *very true*" and "Tina is young is *almost true*". We show the syntax, semantics and reasoning algorithms for the extension provided in this paper.

We proceed as follows. To make the paper self-contained, the next section recalls salient notions of mathematical fuzzy logics [11]. Section 3 introduces fuzzy $\mathcal{ALC}$ [3], which is the fuzzy variant of the Description Logic $\mathcal{ALC}$. $\mathcal{ALC}$ is usually considered as a reference language, whenever new features are introduced into DLs. Section 4 extends fuzzy DLs with fuzzy truth values and Section 5 provides reasoning algorithms. Section 6 concludes and describes future work.

2 Preliminaries: Mathematical Fuzzy Logic

In the setting of fuzzy logics, the convention prescribing that a statement is either true or false is changed and is a matter of degree measured on an ordered scale $\mathcal{S}$ that is no longer $\{0, 1\}$, but, e.g., the unit interval $[0, 1]$. This degree of fit is called *degree of truth* of the statement ϕ in the interpretation $\mathcal{I}$. Fuzzy logics logics provide compositional calculi of degrees of truth, including degrees between "true" and "false". A statement is now not true or false only, but may have a truth degree taken from a *truth space* $\mathcal{S}$, usually $[0, 1]$ (in that case we speak about *Mathematical Fuzzy Logic* [11]).

In the illustrative fuzzy logic that we consider in this section, *fuzzy statements* have the form $\phi \geqslant l$ or $\phi \leqslant u$, where $l, u \in [0, 1]$ [13, 11] and ϕ is a statement, which encode that the degree of truth of ϕ is *at least* l resp. *at most* u. For example, $ripe\_tomato \geqslant 0.9$ says that we have a rather ripe tomato (the degree of truth of $ripe\_tomato$ is at least 0.9).

A *fuzzy interpretation* $\mathcal{I}$ maps each basic statement p_i into $[0, 1]$ and is then extended inductively to all statements:

$$\begin{array}{lcl} \mathcal{I}(\phi \wedge \psi) & = & \mathcal{I}(\phi) \otimes \mathcal{I}(\psi)\,; \\ \mathcal{I}(\phi \vee \psi) & = & \mathcal{I}(\phi) \oplus \mathcal{I}(\psi)\,; \\ \mathcal{I}(\phi \rightarrow \psi) & = & \mathcal{I}(\phi) \Rightarrow \mathcal{I}(\psi)\,; \\ \mathcal{I}(\neg\phi) & = & \ominus \mathcal{I}(\phi)\,, \end{array}$$

where $\otimes$, $\oplus$, $\Rightarrow$, and $\ominus$ are so-called *truth combination functions*, namely, *triangular norms* (or *t-norms*), *triangular conorms* (*t-conorms*), *implication functions*, and *negation functions*, respectively, which extend the classical Boolean conjunction, disjunction, implication, and negation, respectively, to the fuzzy case (see [11] for a formal definition of these functions and their properties). Several t-norms, t-conorms, implication functions, and negation functions have been given in the literature, giving raise to different fuzzy logics with different logical properties. In fuzzy logic, one usually distinguishes three different logics (see Fig. 1), namely Łukasiewicz, Gödel, and Product logic [11]. Zadeh "logic" (fuzzy operators originally considered by Zadeh [12]) is entailed by Łukasiewicz logic.

A *fuzzy set* R over a countable crisp set X is a function $R\colon X \rightarrow [0, 1]$. A (binary) *fuzzy relation* R over two countable crisp sets X and Y is a function $R\colon X \times Y \rightarrow [0, 1]$. The *degree of subsumption* between two fuzzy sets A and B is defined as $\inf_{x \in X} A(x) \Rightarrow B(x)$.

The notions of satisfiability and logical consequence are defined in the standard way. A fuzzy interpretation $\mathcal{I}$ *satisfies* a fuzzy statement $\phi \geqslant l$ (resp., $\phi \leqslant u$) or $\mathcal{I}$ is a *model* of $\phi \geqslant l$ (resp., $\phi \leqslant u$), denoted $\mathcal{I} \models \phi \geqslant l$ (resp., $\mathcal{I} \models \phi \leqslant u$), iff $\mathcal{I}(\phi) \geqslant l$ (resp., $\mathcal{I}(\phi) \leqslant u$). $\phi \geqslant l$ is a *tight logical consequence* of a set of fuzzy statements $\mathcal{K}$ iff l is the infimum of $\mathcal{I}(\phi)$ subject to all models $\mathcal{I}$ of $\mathcal{K}$. The latter value is equivalent to $l = \sup\{r \mid \mathcal{K} \models \phi \geqslant r\}$, it is called *Best Entailment Degree* (BED), and is denoted $bed(\mathcal{K}, \phi)$, while the *Best Satisfiability Degree*, denoted as $bsd(\mathcal{K}, \phi)$ is defined as $\sup_{\mathcal{I} \models \mathcal{K}} \mathcal{I}(\phi)$. We refer the reader to [11, 13, 14] for reasoning algorithms for fuzzy propositional and First-Order Logics (FOLs).

ISBN: 978-989-95079-6-8

	Łukasiewicz Logic	Gödel Logic	Product Logic	Zadeh Logic
$a \otimes b$	$\max(a+b-1,0)$	$\min(a,b)$	$a \cdot b$	$\min(a,b)$
$a \oplus b$	$\min(a+b,1)$	$\max(a,b)$	$a+b-a \cdot b$	$\max(a,b)$
$a \Rightarrow b$	$\min(1-a+b,1)$	$\begin{cases} 1 & \text{if } a \leq b \\ b & \text{otherwise} \end{cases}$	$\min(1,b/a)$	$\max(1-a,b)$
$\ominus a$	$1-a$	$\begin{cases} 1 & \text{if } a=0 \\ 0 & \text{otherwise} \end{cases}$	$\begin{cases} 1 & \text{if } a=0 \\ 0 & \text{otherwise} \end{cases}$	$1-a$

Figure 1: Combination functions of various fuzzy logics.

For illustrative purposes, we provide a simple and effective way to solve the entailment problem, in the case of Zadeh logic and Lukasiewicz logic, in terms of Mixed Integer Linear Programming (MILP) –see also [14]. The calculus depends on the t-norm, t-conorm and negation functions considered. Suppose we are looking for $bed(\mathcal{K},\phi)$, then,

$$bed(\mathcal{K},\phi) = \min\{x \mid \mathcal{K} \cup \{\phi \leq x\} \text{ is satisfiable.}\} \quad (1)$$

Indeed, suppose the minimal value is $\bar{n}$. We will know then that for any interpretation $\mathcal{I}$ satisfying $\mathcal{K}$, it cannot be $\mathcal{I}(\phi) < \bar{n}$, and, thus, $\mathcal{I}(\phi) \geq \bar{n}$ has to hold.

The above problem can be solved by means of MILP (we consider Zadeh logic only, for Lukasiewicz logic the procedure is similar) [1]. For a formula ϕ consider a variable x_ϕ. The intuition is that the degree of truth of ϕ is greater or equal to x_ϕ. The MILP problem determining $bed(\mathcal{K},\phi)$ is as follows:

$$\begin{array}{l} \min x. \text{ such that } x \in [0,1] \\ \quad x_{\neg\phi} \geq 1-x, \sigma(\neg\phi), \\ \quad \text{for all } (\phi' \geq n) \in \mathcal{K}, x_{\phi'} \geq n, \sigma(\phi'), \\ \quad \text{for all } (\phi' \leq n) \in \mathcal{K}, x_{\neg\phi'} \geq 1-n, \sigma(\neg\phi') \end{array} \quad (2)$$

where the function σ, transforming a many-valued proposition into a set of inequations, is inductively defined as follows:

$$\sigma(\phi) = \begin{cases} x_p \in [0,1] & \text{if } \phi = p \\ x_{\phi'} = 1 - x_\phi, x_\phi \in [0,1] & \text{if } \phi = \neg\phi' \\ x_{\phi_1} \geq x_\phi, x_{\phi_2} \geq x_\phi, \\ \sigma(\phi_1), \sigma(\phi_2), x_\phi \in [0,1] & \text{if } \phi = \phi_1 \wedge \phi_2 \\ x_{\phi_1} + x_{\phi_2} = x_\phi, x_{\phi_1} \leq y, \\ x_{\phi_2} \leq 1-y, \sigma(\phi_1), \sigma(\phi_2), \\ y \in \{0,1\}, x_\phi \in [0,1], \\ \text{where } y \text{ is a new binary variable} & \text{if } \phi = \phi_1 \vee \phi_2 \end{cases}.$$

In a similar way, we may determine $bsd(\mathcal{K},\phi)$ as

$$\begin{array}{l} \min -x. \text{ such that } x \in [0,1] \\ \quad x_\phi \geq x, \sigma(\phi), \\ \quad \text{for all } (\phi' \geq n) \in \mathcal{K}, x_{\phi'} \geq n, \sigma(\phi'), \\ \quad \text{for all } (\phi' \leq n) \in \mathcal{K}, x_{\neg\phi'} \geq 1-n, \sigma(\neg\phi') \end{array} \quad (3)$$

Note that we may verify whether $\mathcal{K}$ is satisfiable by checking if $bed(\mathcal{K},p) = 1$, where p is a new propositional letter not occurring in $\mathcal{K}$, and that under Łukasiewicz logic and Zadeh semantics we end up with a *bounded Mixed Integer Linear Program* (bMILP) optimization problem [15].

3 Fuzzy $\mathcal{ALC}$

Syntax. Consider an alphabet of *concepts names* (denoted A), *abstract roles names* (denoted R), *abstract individual names* (denoted a). From a First-Order Logic point of view, concepts may be seen as a formulae with one free variable (and, thus, may be seen as class descriptors), while roles as binary predicates (and, thus, may be used to describe properties of a class). *Concepts* (denoted C or D) of the language can be built inductively from atomic concepts (A), top concept $\top$, bottom concept $\bot$, abstract roles (R) as:

$C, D \rightarrow$	A	$\mid$	(atomic concept)
	$\top$	$\mid$	(top concept)
	$\bot$	$\mid$	(bottom concept)
	$C \sqcap D$	$\mid$	(concept conjunction)
	$C \sqcup D$	$\mid$	(concept disjunction)
	$\neg C$	$\mid$	(concept negation)
	$\forall R.C$	$\mid$	(universal quantification)
	$\exists R.C$		(existential quantification)

As illustrative purpose, Fig. 2 provides a First-Order Logic translation of $\mathcal{ALC}$ concepts and examples.

Syntax			*FOL*	*Example*
C, D	$\rightarrow$	$\top$	$\top(x)$	
		$\bot$	$\bot(x)$	
		A	$A(x)$	$Human$
		$C \sqcap D$	$C(x) \wedge D(x)$	$Human \sqcap Male$
		$C \sqcup D$	$C(x) \vee D(x)$	$Nice \sqcup Rich$
		$\neg C$	$\neg C(x)$	$\neg Meat$
		$\exists R.C$	$\exists y.R(x,y) \wedge C(y)$	$\exists has\_child.Blond$
		$\forall R.C$	$\forall y.R(x,y) \rightarrow C(y)$	$\forall has\_child.Human$

Figure 2: $\mathcal{ALC}$ concepts and First-Order Logic reading.

A *Fuzzy Knowledge Base* $\mathcal{K}$ comprises a fuzzy ABox $\mathcal{A}$ and a fuzzy TBox $\mathcal{T}$. A *fuzzy ABox* consists of a finite set of *fuzzy assertions* of one of the following types: a fuzzy *concept assertion* of the form $\langle a{:}C, n\rangle$ (with informal meaning, individual a is an instance of concept C with degree at least n) or a fuzzy *role assertion* of the form $\langle (a,b){:}R, n\rangle$ (the pair of individuals (a,b) is an instance of role R with degree at least n). In FOL, $\langle a{:}C, n\rangle$ may be seen as a fuzzy statement of the form $C(a) \geqslant n$, while $\langle (a,b){:}R, n\rangle$ may be seen as a fuzzy statement of the form $R(a,b) \geqslant n$.

In general, 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 and $n \in (0,1]$. Informally, $\langle C \sqsubseteq D, n\rangle$ states that all instances of concept C are instances of concept D to degree n.$C = D$ is a shorthand for the two axioms $\langle C \sqsubseteq D, 1\rangle$ and $\langle D \sqsubseteq C, 1\rangle$.

However, for computational reasons, we will restrict TBoxes to be *acyclic*. That is, $\mathcal{T}$ is a finite set of *fuzzy concept inclusion axioms* $\langle A \sqsubseteq C, n\rangle$, and *concept definitions* $A = C$, where A is an atomic concept. Furthermore, we assume that $\mathcal{T}$ verifies two additional constraints: *(i)* there is no concept A such that it appears more than once on the left hand side of some axiom in $\mathcal{T}$. *(ii)* no cyclic definitions are present in $\mathcal{T}$. We will say that A *directly uses* primitive concept B in $\mathcal{T}$, if there is some axiom $\tau \in \mathcal{T}$ such that A is on the left hand side of τ and B occurs in the right hand side of τ. Let *uses* be the transitive closure of the relation *directly uses* in $\mathcal{T}$. $\mathcal{T}$ is cyclic iff there is A such that A uses A in $\mathcal{T}$.

[1] Note that this is an optimized version w.r.t. [13, 14].

ISBN: 978-989-95079-6-8

It is well known that such TBoxes can be eliminated through a finite (although it can create an exponential growth of the KB), *expansion process*, both in the crisp and in the fuzzy case [3]. Instead, we will use a extension to the fuzzy case of the *lazy expansion* technique [16], which has proved to be more useful in practice.

Semantics. From a semantics point of view, a fuzzy interpretation $\mathcal{I}$ is a pair $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a non empty set $\Delta^{\mathcal{I}}$ (the interpretation domain) and a fuzzy interpretation function $\cdot^{\mathcal{I}}$ mapping: (i) an *abstract individual* a onto an element $a^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ such that if $a \neq b$ then $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ (Unique Name Assumption); (ii) a *concept* C onto a function $C^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0,1]$; (iii) an *abstract role* R onto a function $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0,1]$.

Given arbitraries t-norm $\otimes$, t-conorm $\oplus$, negation function $\ominus$ and implication function $\Rightarrow$, the fuzzy interpretation function is extended to *complex concepts and roles* as shown in Fig. 3, where $C^{\mathcal{I}}$ denotes the membership function of the fuzzy concept C with respect to the fuzzy interpretation $\mathcal{I}$. $C^{\mathcal{I}}(x)$ gives us the degree of being the individual x an element of the fuzzy concept C under $\mathcal{I}$. Similarly, $R^{\mathcal{I}}$ denotes the membership function of the fuzzy role R with respect to $\mathcal{I}$. $R^{\mathcal{I}}(x,y)$ gives us the degree of being (x,y) an element of the fuzzy role R under $\mathcal{I}$. The fuzzy interpretation function is extended to *fuzzy axioms* in Fig. 3.

Concept		Semantics
$(\top)^{\mathcal{I}}(x)$	=	1
$(\bot)^{\mathcal{I}}(x)$	=	0
$(A)^{\mathcal{I}}(x)$	=	$A^{\mathcal{I}}(x)$
$(C \sqcap D)^{\mathcal{I}}(x)$	=	$C^{\mathcal{I}}(x) \otimes D^{\mathcal{I}}(x)$
$(C \sqcup D)^{\mathcal{I}}(x)$	=	$C^{\mathcal{I}}(x) \oplus D^{\mathcal{I}}(x)$
$(\neg C)^{\mathcal{I}}(x)$	=	$\ominus C^{\mathcal{I}}(x)$
$(\forall R.C)^{\mathcal{I}}(x)$	=	$\inf_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \Rightarrow C^{\mathcal{I}}(y)\}$
$(\exists R.C)^{\mathcal{I}}(x)$	=	$\sup_{y \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(x,y) \otimes C^{\mathcal{I}}(y)\}$
Axiom		**Semantics**
$(a:C)^{\mathcal{I}}$	=	$C^{\mathcal{I}}(a^{\mathcal{I}})$
$((a,b):R)^{\mathcal{I}}$	=	$R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}})$
$(C \sqsubseteq D)^{\mathcal{I}}$	=	$\inf_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) \Rightarrow D^{\mathcal{I}}(x)\}$

Figure 3: Semantics of fuzzy concepts and axioms.

A fuzzy interpretation $\mathcal{I}$ *satisfies* (is a *model* of) a fuzzy statement $\langle \alpha, n \rangle$ iff $\alpha^{\mathcal{I}} \geq n$. The notions of logical consequence, best entailment degree and best satisfiability degree of α are as for Section 2. We additionally define the *Best Satisfiability Degree* [6] of a concept C w.r.t. a fuzzy KB $\mathcal{K}$ as $bsd(\mathcal{K}, C) = \sup_{\mathcal{I} \models \mathcal{K}} \sup_{x \in \Delta^{\mathcal{I}}} C^{\mathcal{I}}(x)$.

4 Fuzzy DLs with fuzzy truth values

So far, we have seen that fuzzy statements are of the form $\langle \alpha, n \rangle$, where $n \in [0,1]$. As next, we extend such statements to allow fuzzy truth values, such as "very true, almost true, almost false", in place of a "crisp" value $n \in [0,1]$. Essentially, we have fuzzy truth-qualified statements in which the truth is now a fuzzy set and, thus, we allow statements such as "Tina is young is very true".

As pointed out in [17] (see also [4, 6, 18]), there are many functions to specify fuzzy set membership degrees in fuzzy set theory and practice. The most frequently used are the trapezoidal (Fig. 4 (a)), the triangular (Fig. 4 (b)), the L-function (left-shoulder function, Fig. 4 (c)), the R-function (right-shoulder function, Fig. 4 (d)) and linear hedges (Fig. 4 (e), where $a = c/(c+1), b = 1/(c+1)$). We will call these functions (defined over $[0,1]$) *truth qualifiers* and allow them to be used to modify the degree of truth of a sentence. We will use

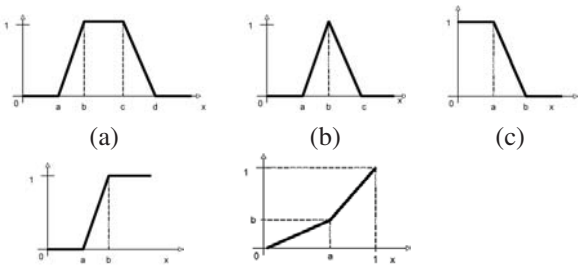

Figure 4: (a) Trapezoidal function; (b) Triangular function; (c) L-function; (d) R-function; (e) Linear function.

the abbreviations $\mathsf{trpz(a,b,c,d)}, \mathsf{tri(a,b,c)}, \mathsf{ls(a,b)}, \mathsf{rs(a,b)}$ and $\mathsf{ln(c)}$ to denote them, respectively.

Syntax. Let α be a concept assertion $a{:}C$, a role assertion $(a,b){:}R$ or a GCI $C \sqsubseteq D$, and $n \in [0,1]$. Let q be a truth qualifier, as in Fig. 4. Then we extend $\mathcal{ALC}$ axioms $\langle \alpha, n \rangle$ also to be of the form $\langle \alpha, q \rangle$, where a truth qualifier q may occur in place of a value $n \in [0,1]$. For instance, "Tina is young is very true" may be represented with $\langle \mathsf{tina{:}Young}, \mathsf{ln(4)} \rangle$. Similarly, we may represent an imprecision about the actual degree of truth, such as, "Tina is young is true to degree around 0.7", as an axioms of the form $\langle \mathsf{tina{:}Young}, \mathsf{tri(0.6, 0.7, 0.8)} \rangle$.

In the following, a knowledge base will be split into two parts, $\mathcal{K} = \mathcal{K}_U \cup \mathcal{K}_Q$ in which $\mathcal{K}_U$ contains only unqualified fuzzy axioms of type $\langle \alpha, n \rangle$, where $n \in [0,1]$, while $\mathcal{K}_Q$ contains only qualified fuzzy axioms of type $\langle \alpha, q \rangle$, where q is a truth qualifier. In FOL, we may see $\mathcal{K}$ as the formula

$$\Gamma_{\mathcal{K}} = \bigwedge_{\tau_i \in \mathcal{K}_U} \tau_i \wedge \bigwedge_{\tau_j \in \mathcal{K}_Q} \tau_j \,. \tag{4}$$

We will also use Γ_U for the left conjunct, while use Γ_Q for the right conjunct and, thus, $\Gamma_{\mathcal{K}} = \Gamma_U \wedge \Gamma_Q$.

Semantics. So far, given an interpretation $\mathcal{I}$, an axiom of the form $\langle \alpha, n \rangle$ is crisp under $\mathcal{I}$, in the sense that either $\mathcal{I}$ satisfies $\langle \alpha, n \rangle$ (i.e., $\alpha^{\mathcal{I}} \geq n$) or $\mathcal{I}$ does not satisfy $\langle \alpha, n \rangle$ (i.e., $\alpha^{\mathcal{I}} < n$). Once we move to truth qualified axioms, axioms are no longer true or false, but have a degree of truth depending on the qualifier. More precisely, the *degree of truth* of a fuzzy axiom τ of the form $\langle \alpha, q \rangle$ under $\mathcal{I}$, denoted $\mathcal{I}(\tau)$ is defined as the value $q(\alpha^{\mathcal{I}})$ (the application of the qualifier q to the truth value $\alpha^{\mathcal{I}}$). So, if "Tina is Young" is true to degree 0.9 under $\mathcal{I}$, then "Tina is young is very true" to degree 0.6 under $\mathcal{I}$. We may also extend this notion the axioms of the form $\langle \alpha, n \rangle$ by defining the *degree of truth* of a fuzzy axiom τ of the form $\langle \alpha, n \rangle$ to be $\mathcal{I}(\tau) = 1$ if $\alpha^{\mathcal{I}} \geq n$, otherwise $\mathcal{I}(\tau) = 0$, which is compatible with what we have defined so far.

Now, consider a knowledge base $\mathcal{K} = \mathcal{K}_U \cup \mathcal{K}_Q$, a fuzzy axiom τ and an interpretation $\mathcal{I}$. The *degree of truth* of $\mathcal{K}_x$ ($x \in \{U, Q\}$) under $\mathcal{I}$, denoted $\mathcal{I}(\mathcal{K}_x)$ is defined as

$$\mathcal{I}(\mathcal{K}_x) = \bigotimes_{\tau_i \in \mathcal{K}_x} \mathcal{I}(\tau_i) \,.$$

In FOL, this is the same as $\mathcal{I}(\Gamma_x)$. Note that if $\mathcal{I}$ is a model of $\mathcal{K}_U$ then $\mathcal{I}(\mathcal{K}_U) = 1$, else $\mathcal{I}(\mathcal{K}_U) = 0$. Furthermore, the *degree of entailment* of a τ w.r.t. $\mathcal{K}$ under $\mathcal{I}$, denoted $\mathcal{I}(\mathcal{K}, \tau)$, is defined as

$$\mathcal{I}(\mathcal{K}, \tau) = (\mathcal{I}(\mathcal{K}_U) \otimes \mathcal{I}(\mathcal{K}_Q)) \Rightarrow \mathcal{I}(\tau) \,. \tag{5}$$

ISBN: 978-989-95079-6-8

Essentially, the degree of entailment under $\mathcal{I}$ is the evaluation under $\mathcal{I}$ of the FOL implication $(\Gamma_U \wedge \Gamma_Q) \rightarrow \tau$, which is quite natural. Please note that if $\mathcal{I} \not\models \mathcal{K}_U$ then $\mathcal{I}(\mathcal{K}_U) = 0$ and, thus, $\mathcal{I}(\Gamma_\mathcal{K} \rightarrow \tau) = 1$, while if $\mathcal{I} \models \mathcal{K}_U$ then $\mathcal{I}(\Gamma_\mathcal{K} \rightarrow \tau) = \mathcal{I}(\Gamma_Q \rightarrow \tau)$. The *Best Entailment Degree* of a τ w.r.t. $\mathcal{K}$, denoted $bed(\mathcal{K}, \tau)$ is defined as $bed(\mathcal{K}, \tau) = \inf_\mathcal{I} \mathcal{I}(\mathcal{K}, \tau)$. By the previous observations, it easily follows that

$$bed(\mathcal{K}, \tau) = \inf_{\mathcal{I} \models \mathcal{K}_U} \mathcal{I}(\Gamma_Q \rightarrow \tau) \,, \tag{6}$$

where $\inf \emptyset = 1$. Finally, for an axiom α of the form $a{:}C$, $(a,b){:}R$ or a $C \sqsubseteq D$, the *Best Entailment Degree* of α w.r.t. $\mathcal{K}$, denoted $bed(\mathcal{K}, \alpha)$ is defined as

$$bed(\mathcal{K}, \alpha) = \sup\{n \mid 1 = bed(\mathcal{K}, \langle \alpha, n \rangle)\} \,, \tag{7}$$

and the *Best Satisfiability Degree* of a C w.r.t. $\mathcal{K}$, is

$$bsd(\mathcal{K}, C) = \sup_{\mathcal{I} \models \mathcal{K}_U} \sup_{x \in \Delta^\mathcal{I}} \mathcal{I}(\Gamma_Q) \Rightarrow C^\mathcal{I}(x) \,.$$

5 Reasoning

We next provide a reasoning algorithm for fuzzy $\mathcal{ALC}$ with truth qualifiers. To start with, we require a calculus for fuzzy $\mathcal{ALC}$ without truth qualifiers. Our algorithm is inspired by the one implemented within the FUZZYDL system [6] (which follows from the one presented in [17]). However, the presence of truth qualifiers requires some modifications. Furthermore, the version we present here for $\mathcal{ALC}$, introduces some optimizations that will require less "variables" and, thus, is expected to be more efficient. For the sake of this paper, *we provide a calculus under Łukasiewicz logic*, as in [17, 19]. A similar algorithm under product logic is expected to be more involved (see, e.g., [20, 21]).

W.l.o.g., we may assume that concepts are in *Negation Normal Form* (NNF), which is obtained by pushing in the usual manner negation on front of concept names only, by applying recursively the equivalences below.

$\neg\top \equiv \bot$	$\neg(C_1 \sqcup C_2) \equiv \neg C_1 \sqcap \neg C_2$	$\neg\exists R.C \equiv \forall R.\neg C$
$C \sqcap \bot \equiv \bot$	$C \sqcup \bot \equiv C$	$\neg\neg C \equiv C$
$\neg\bot \equiv \top$	$\neg(C_1 \sqcap C_2) \equiv \neg C_1 \sqcup \neg C_2$	$\neg\forall R.C \equiv \exists R.\neg C$
$C \sqcap \top \equiv C$	$C \sqcup \top \equiv \top$	

Now, our goal is to provide a terminating procedure determining $bed(\mathcal{K}, \alpha)$, where α is of the form $a{:}C$, $(a,b){:}R$ or a $C \sqsubseteq D$, and $bed(\mathcal{K}, \tau,)$, for truth qualified fuzzy axioms τ of the form $\langle \alpha, q \rangle$ (q is a truth qualifier).

5.1 Reasoning in $\mathcal{ALC}$ without truth qualifiers

Let's focus first on $bed(\mathcal{K}, \alpha)$, where $\mathcal{K}$ *does not* contain any truth qualifier, i.e. $\mathcal{K}_Q = \emptyset$. If α is a fuzzy role assertion of the form $(a,b){:}R$ then, in order to determine $bed(\mathcal{K}, (a,b){:}R)$, we may reduce it to the BED for concept assertions, as

$$bed(\mathcal{K}, (a,b){:}R) = bed(\mathcal{K} \cup \{\langle b{:}B, 1\rangle\}, a{:}\exists R.B) \,,$$

where B is a new concept (i.e., it does not occur in $\mathcal{K}$).

Consider $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$, where $\mathcal{T}$ is acyclic. The basic idea behind our reasoning algorithm is based on the observations and algorithm of Section 2. So, not surprisingly, in order to determine $bed(\mathcal{K}, \alpha)$, we will compute it as

$$bed(\mathcal{K}, \alpha) = \min x. \text{ such that } \mathcal{K} \cup \{(\alpha \leq x)\} \text{ satisfiable.} \tag{8}$$

Then by applying satisfiability preserving rules, we generate new *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 BED, we *minimize* the original variable x such that all constraints are satisfied. More specifically,

$$\begin{array}{lcl} bed(\mathcal{K}, a{:}C) & = & \min x \text{ such that } \mathcal{K} \cup \{\langle a{:}\neg C, 1-x\rangle\} \text{ satisfiable} \\ bed(\mathcal{K}, C \sqsubseteq D) & = & \min x \text{ such that } \mathcal{K} \cup \{\langle b{:}C \sqcap \neg D, 1-x\rangle\} \text{ satisfiable} \\ bsd(\mathcal{K}, C) & = & \min -x \text{ such that } \mathcal{K} \cup \{\langle b{:}C, x\rangle\} \text{ satisfiable} \,, \end{array}$$

where b is a new individual. This means that determining the minimum degree of satisfiability [2] of a KB is the main reasoning issue to be addressed. Note also that we may determine if $\mathcal{K}$ has a model by e.g. checking whether $bed(\mathcal{K}, b{:}A) = 1$, where b and A do not occur in $\mathcal{K}$.

Like most of the tableaux algorithms, our algorithm works on *completion-forests* since an ABox might contain several individuals with arbitrary roles connecting them. A completion-forest $\mathcal{F}$ is a collection of trees whose distinguished roots are arbitrarily connected by edges. The forest has associated a set $\mathcal{C}_\mathcal{F}$ of constraints of the form $l \leq l'$, $l = l'$, $x_i \in [0,1], y_i \in \{0,1\}$, where l, l' are arithmetic expressions, on the variables occurring the node labels and edge labels.

Each node v is labelled with a set $\mathcal{L}(v)$ of concepts C. If $C \in \mathcal{L}(v)$ then we consider a variable $x_{v:C}$. The intuition here is that v is an instance of C to degree equal or greater than of the value of the variable $x_{v:C}$ in a minimal solution. Essentially, $x_{v:C}$ will hold the degree of truth of $v{:}C$. Each edge $\langle v, w\rangle$ is labelled with a set $\mathcal{L}(\langle v, w\rangle)$ of roles R. If $R \in \mathcal{L}(\langle v, w\rangle)$ then we consider a variable $x_{(v,w):R}$ (the intuition here is that $\langle v, w\rangle$ is an instance of R to degree equal or greater than the value of the variable $x_{(v,w):R}$ in a minimal solution (as for concept assertions, $x_{(v,w):R}$ will hold the degree of truth of $(v,w){:}R$). We will assume that there is a bijection between assertions α and variables x_α.

We are ready now to present our calculus. We first start with an initialization step, which builds the starting completion-forest. Then we apply to it a set of completion-forest transforming rules until no more rule can be applied. Finally, we solve the MILP problem associated to the set of constraints.

The algorithm initializes a forest $\mathcal{F}$ as follows. Consider $\mathcal{K}$: (i) $\mathcal{F}$ contains a root node a_i for each individual a_i occurring in $\mathcal{A}$; (ii) $\mathcal{F}$ contains an edge $\langle a, b\rangle$ for each fuzzy assertion $\langle (a,b){:}R, n\rangle \in \mathcal{A}$; (iii) for each fuzzy assertion $\langle a{:}C, n\rangle \in \mathcal{A}$, add both C to $\mathcal{L}(a)$ and $x_{a:C} \geq n$ to the set of constraints $\mathcal{C}_\mathcal{F}$; (iv) for each fuzzy assertion $\langle (a,b){:}R, n\rangle \in \mathcal{A}$, add both R to $\mathcal{L}(\langle a, b\rangle)$ and $x_{(a,b):R} \geq n$ to the set of constraints $\mathcal{C}_\mathcal{F}$; (v) for all introduced variables x_α, add $x_\alpha \in [0,1]$ to $\mathcal{C}_\mathcal{F}$.

$\mathcal{F}$ 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.

As anticipated, we will use a extension to the fuzzy case of the *lazy expansion* technique in order to remove the axioms in $\mathcal{T}$. The basic idea is as follows (recall that there are only two types of fuzzy concept inclusions): given $\langle A \sqsubseteq C, n\rangle$, add C only to nodes with a label containing A, and given $\langle C \sqsubseteq A, 1\rangle$, add $\neg C$ only to nodes with a label containing $\neg A$.

[2] In the last equation, we use the fact that in our setting $\max x \; s.t. \; x \in S = \min -x \; s.t. \; x \in S$.

ISBN: 978-989-95079-6-8

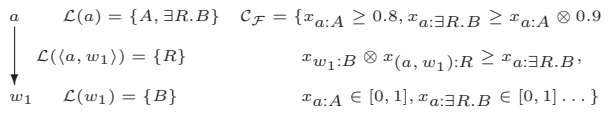

Figure 5: Complete forest for Example 1.

We 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 $\mathcal{F}$. Note that we do not need a notion of *blocking* as $\mathcal{T}$ is acyclic.

Now we are ready to present the inference rules:

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

$(\bot)$. If $\bot \in \mathcal{L}(v)$ then $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{v:\bot} = 0\}$.

$(\top)$. If $\top \in \mathcal{L}(v)$ then $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{v:\top} = 1\}$.

$(\sqcap)$. If *(i)* $C \sqcap D \in \mathcal{L}(v)$, and *(ii)* not both $C \in \mathcal{L}(v)$ and $D \in \mathcal{L}(v)$ then add C and D to $\mathcal{L}(v)$, and $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{v:C} \otimes x_{v:D} = x_{a:C \sqcap D}\} \cup \{x_{v:C}, x_{v:D} \in [0,1]\}$.

$(\sqcup)$. If *(i)* $C \sqcup D \in \mathcal{L}(v)$, and *(ii)* not both $C \in \mathcal{L}(v)$ and $D \in \mathcal{L}(v)$ then add C and D to $\mathcal{L}(v)$, and $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{v:C} \oplus x_{v:D} = x_{a:C \sqcup D}\} \cup \{x_{v:C}, x_{v:D} \in [0,1]\}$.

$(\forall)$. If *(i)* $\forall R.C \in \mathcal{L}(v)$, $R \in \mathcal{L}(\langle v, w\rangle)$, and *(ii)* the rule has not been already applied to this pair then add C to $\mathcal{L}(w)$, and $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{w:C} \geq x_{a:\forall R.C} \otimes x_{(v,w):R}\} \cup \{x_{w:C} \in [0,1]\}$.

$(\sqsubseteq)$. If *(i)* $\langle A \sqsubseteq C, n\rangle \in \mathcal{T}$, *(ii)* $A \in \mathcal{L}(v)$, and *(iii)* v is a node to which this rule has not yet been applied then *(i)* append C to $\mathcal{L}(v)$, and *(ii)* $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{v:C} \geq x_{v:A} \otimes n\} \cup \{x_{v:C} \in [0,1]\}$.

$(\bar{\sqsubseteq})$. If *(i)* $\langle C \sqsubseteq A, 1\rangle \in \mathcal{T}$, *(ii)* $\neg A$, and *(iii)* v is a node to which this rule has not yet been applied then *(i)* append $\neg C$ to $\mathcal{L}(v)$, and *(ii)* $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{v:\neg A} \geq x_{v:\neg C}\} \cup \{x_{v:\neg C} \in [0,1]\}$.

$(\exists)$. If $\exists R.C \in \mathcal{L}(v)$ then create a new node w, add R to $\mathcal{L}(\langle v, w\rangle)$, add C to $\mathcal{L}(w)$, and $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{w:C} \otimes x_{(v,w):R} = x_{v:\exists R.C}\} \cup \{x_{(v,w):R}, x_{w:C} \in [0,1]\}$.

Note that in order to write the fuzzy operators, we may need to create some new control variables. For example, under Łukasiewicz t-norm, $x_1 \otimes x_2 \geq l$ can be written as $\{l \leq y, x_1 + x_2 \leq 1 + l, x_1 + x_2 - l \geq y, y_i \in \{0,1\}\}$. If $y = 0$, then $l = 0$ (it simulates the case where $x_1 + x_2 \leq 1$, and hence $x_1 \otimes x_2 = 0$), and if $y = 1$, then $l = x_1 + x_2 - 1$ [17, 19].

Example 1 illustrates the behaviour of the algorithm.

Example 1 *Consider* $\mathcal{K} = \{\langle A \sqsubseteq \exists R.B, 0.9\rangle, \langle a{:}A, 0.8\rangle\}$. *Let's show that* $\mathcal{K}$ *is satisfiable. The forest* $\mathcal{F}$ *is initialized with a root node* a*, labelled with* $\mathcal{L}(a) = \{A\}$*, and set of constraints* $\mathcal{C}_\mathcal{F} = \{x_{a:A} \geq 0.8\}$*. The we apply the* $(\sqsubseteq)$ *rule and add* A *and* $\exists R.B$ *to* $\mathcal{L}(a)$*, and add to the constraint set* $\{x_{a:\exists R.B} \geq x_{a:A} \otimes 0.9\} \cup \{x_{a:A}, x_{a:\exists R.B} \in [0,1]\}$*. As next, we apply the* $(\exists)$ *rule to node* a *and, thus, we create a new node* w_1*, labelled with* $\mathcal{L}(w_1) = \{B\}$*, and an edge* $\langle a, w_1\rangle$ *labelled with* $\mathcal{L}(\langle a, w_1\rangle) = \{R\}$*, and we update the constraint set with* $\mathcal{C}_\mathcal{F} = \mathcal{C}_\mathcal{F} \cup \{x_{w_1:B} \otimes x_{(a,w_1):R} = x_{a:\exists R.B}\} \cup \{x_{(a,w_1):R}, x_{w_1:B} \in [0,1]\}$*. Rule* $(\sqsubseteq)$ *is not applicable to node* w_1 *because* $A \notin \mathcal{L}(b)$*.*

The complete forest $\mathcal{F}$ *in Figure 5 shows the computation so far. It only remains to find a solution to the inequalities.* □

Note that there is a significative difference with other similar algorithms for fuzzy DLs combining tableau algorithms with optimization problems [17, 20, 9]. In those algorithms, every time a concept C appears in the list of expressions of a node v, a new variable x is created. Instead, we introduce a variable $x_{v:C}$ once, and reuse it the following times. This reduction in the number of generated variables is important because it makes the bMILP problem easier to solve. We have:

Proposition 1 *For each KB* $\mathcal{K}$*, the tableau algorithm terminates and computes* $bed(\mathcal{K}, \alpha)$ *and* $bsd(\mathcal{K}, C)$*.*

5.2 Reasoning in $\mathcal{ALC}$ with truth qualifiers

Let us now consider the case $\mathcal{K}_Q \neq \emptyset$. We next address the problem of determining $bed(\mathcal{K}, \tau)$ and $bed(\mathcal{K}, \alpha)$. From Eq. (6), we now that if $\mathcal{K}_U$ has no model then immediately $bed(\mathcal{K}, \tau) = 1$ and, thus, this case is not of particular interest. So, let us assume that $\mathcal{K}_U$ has a model. The case $\tau = \langle \alpha, n\rangle$, with $n \in [0,1]$, is also not of particular interest as it suffices to compute $m = bed(\mathcal{K}, \alpha)$ and check whether $m \geq n$. So, it remains to determine both $bed(\mathcal{K}, \tau)$, for a fuzzy truth qualified statement τ, and $bed(\mathcal{K}, \alpha)$, where α is of the form $a{:}C$, $(a,b){:}R$ or a $C \sqsubseteq D$. From what we have seen so far, it is immediate that (where a is new individual [3])

$$bed(\mathcal{K}, \tau) = \min x. \text{ such that } \mathcal{K}_U \cup \{(\Gamma_Q \rightarrow \tau) \leq x\} \text{ satisfiable.} \quad (9)$$

$$bsd(\mathcal{K}, C) = \min -x. \text{ such that } \mathcal{K}_U \cup \{(\Gamma_Q \rightarrow a{:}C) \geq x\} \text{ satisfiable.} \quad (10)$$

$$bed(\mathcal{K}, \alpha) = \min x. \text{ such that } \mathcal{K}_U \cup \{(\Gamma_Q \rightarrow (\alpha \leq x)) = 1\} \text{ satisfiable.} \quad (11)$$

Now, let us address first the problem (9). The calculus is of the same style as in the previous section. We have that

$$bed(\mathcal{K}, \tau) = \min x. \text{ such that } \mathcal{K}_U \cup \{(\Gamma_Q \wedge \neg\tau) \geq 1 - x\} \text{ satisfiable .}$$

Therefore, it suffices to provide rules for encoding $(\Gamma_Q \wedge \neg\tau) \geq 1 - x$ as a set of MILP constraints. This is obtained as follows. Consider $\Gamma_Q = \bigwedge_{\tau_j \in \mathcal{K}_Q} \tau_j$ and assume $\tau = \langle \alpha, q\rangle$. Consider variables $x_{QC}, x_{\Gamma_Q}, x_{\neg\tau}$, and x_{τ_j}, where x_{Γ_Q} will hold the degree of truth of Γ_Q, x_{τ_j} ($x_{\neg\tau}$) will hold the degree of truth of τ_j ($\neg\tau$) and x_{QC} will hold the degree of truth of $(\Gamma_Q \wedge \neg\tau)$. Then, to encode $(\Gamma_Q \wedge \neg\tau)) \geq 1 - x$ we need

$$\begin{array}{lcl} x_{QC} & \geq & 1 - x \\ x_{\Gamma_Q} \otimes x_{\neg\tau} & = & x_{QC} \\ x_{\Gamma_Q} & = & \bigotimes_{\tau_j \in \mathcal{K}_Q} x_{\tau_j} \\ x_{QC}, x_{\Gamma_Q}, x_{\tau_j}, x_{\neg\tau} & \in & [0,1]\,. \end{array} \quad (12)$$

and we add $\langle \alpha, x_\alpha\rangle$ to $\mathcal{K}_U$. We now have to connect correctly the value of x_{τ_j} ($x_{\neg\tau}$) to the degree of truth of $\tau_j = \langle \alpha_j, q_j\rangle$ ($\neg\tau$). To this end, we need the constraints

$$x_{\tau_j} = q_j(x_{\alpha_j}) \quad (13)$$

$$x_{\neg\tau} = 1 - q(x_\alpha) \quad (14)$$

and we add $\langle \alpha, x_\alpha\rangle$ to $\mathcal{K}_U$. Note that, e.g. in Eq. 13, the equation $x_{\tau_j} = q(x_\alpha)$ is MILP expressible as q is MILP expressible function (see, e.g. [17]). For instance, if q is $\mathsf{ls}(\mathsf{a}, \mathsf{b})$ then $x_{\tau_j} = q(x_\alpha)$ may be expressed as

$$\begin{array}{l} x_\alpha + (1-a)y_1 \leq 1, x_\alpha - ay_2 \geq 0, x_\alpha + (1-b)y_2 \leq 1, \\ (b-a)x_{\tau_j} \geq x_\alpha - a - (1-a)y_2, \\ (b-a)x_{\tau_j} \leq x_\alpha - a - (1+a)y_1 + (b-a)y_2, \\ x_\alpha - by_3 \geq 0, x_{\tau_j} + y_3 \leq 1, y_1 + y_2 + y_3 = 1, \\ x_\alpha \in [0,1], y_i \in \{0,1\}\}, \end{array}$$

[3] Concerning Equation (10), suppose the minimal value is $\bar{n}$. We will know then that for any interpretation $\mathcal{I}$ satisfying $\mathcal{K}_U$, it cannot be $\mathcal{I}(\Gamma_Q \rightarrow (\alpha < \bar{n})) = 1$, and, thus, $\mathcal{I}(\Gamma_Q \rightarrow (\alpha \geq \bar{n})) = 1)$ has to hold and, thus, $bed(\mathcal{K}, \langle \alpha, \bar{n}\rangle) = 1$ and $\bar{n}$ is tight.

ISBN: 978-989-95079-6-8

where y_i are new variables (Eq. 14 is managed similarly). Now, we may proceed as for Section 5.1, where we consider these additional constraints and in which expressions of the form $\langle \alpha, x_\alpha \rangle$ are handled as for the $\langle \alpha, n \rangle$ case, except that the value n is replaced with the variable x_α instead [4]. The following can be shown:

Proposition 2 *For each KB $\mathcal{K}$, the tableau algorithm terminates and computes $bed(\mathcal{K}, \tau)$.*

As next, let us address the problem of determining $bsd(\mathcal{K}, C)$. The way we proceed is pretty similar as for $bed(\mathcal{K}, \tau)$. Now, x_{QC} will hold the degree of truth of $(\Gamma_Q \rightarrow a{:}C)$. To encode $(\Gamma_Q \rightarrow a{:}C) \geq x$, we replace the constraints (12) with

$$\begin{array}{lcl} x_{QC} & \geq & x \\ x_{\Gamma_Q} \Rightarrow x_{a:C} & = & x_{QC} \\ x_{\Gamma_Q} & = & \bigotimes_{\tau_j \in \mathcal{K}_Q} x_{\tau_j} \\ x_{QC}, x_{\Gamma_Q}, x_{\tau_j} & \in & [0,1]\,, \end{array} \tag{15}$$

we add $\langle a{:}C, x_{a:C} \rangle$ to $\mathcal{K}_U$, and then we proceed as for determining $bed(\mathcal{K}, \tau)$. The following can be shown.

Proposition 3 *For each KB $\mathcal{K}$, the tableau algorithm terminates and computes $bsd(\mathcal{K}, C)$.*

Finally, let us address the problem of determining $bed(\mathcal{K}, \alpha)$. Since $bed(\mathcal{K}, a{:}C) = \min x.$ such that $\mathcal{K}_U \cup \{(\Gamma_Q \rightarrow (a{:}\neg C \geq 1 - x)) = 1\}$ satisfiable, $bed(\mathcal{K}, (a,b){:}R) = bed(\mathcal{K} \cup \{\langle b{:}B, 1\rangle\}, a{:}\exists R.B)$, and $bed(\mathcal{K}, C \sqsubseteq D) = \min x.$ such that $\mathcal{K}_U \cup \{(\Gamma_Q \rightarrow (a{:}C \sqcap \neg D \geq 1 - x)) = 1\}$ satisfiable, we may restrict our attention to show how to encode $(\Gamma_Q \rightarrow (a{:}E \geq 1 - x)) = 1$ in MILP. To this end, let $\tau = \langle a{:}E, 1 - x \rangle$. Then we replace the constraints (12) with (recall that τ is either true or false)

$$\begin{array}{lcl} x_{\Gamma_Q} & \leq & 1 - y \\ x_{a:E} & \geq & 1 - x - y \\ x_{\Gamma_Q} & = & \bigotimes_{\tau_j \in \mathcal{K}_Q} x_{\tau_j} \\ x_{\Gamma_Q}, x_{\tau_j}, x_{a:E} & \in & [0,1] \\ y & \in & \{0,1\}\,, \end{array} \tag{16}$$

we add $\langle a{:}E, x_{a:E} \rangle$ to $\mathcal{K}_U$, and then we proceed as for determining $bed(\mathcal{K}, \tau)$. The following can be shown.

Proposition 4 *For each KB $\mathcal{K}$, the tableau algorithm terminates and computes $bed(\mathcal{K}, \alpha)$.*

Note that we may safe some obvious variables and equations in the constraints derived so far, such as in Eq 12–16.

6 Conclusions & Outlook

We have addressed the problem of allowing to deal with fuzzy truth qualified statements with fuzzy DLs and, thus, allowing statements such as "Tina is young is *almost true*". We have provided syntax, semantics of a fuzzy DL with truth qualified axioms, and a calculus addressing the various reasoning problems presented under Łukasiewicz logic.

We have also provided a novel simplified calculus for fuzzy DLs without truth qualified axioms, which is closer to the usual tableau algorithms for DLs, and which reduces the number of generated variables, and hence the size of the optimization problem to be solved.

It would be possibly to unify both approaches by using truth qualifiers of the form $\mathtt{trpz}(n, n, 1, 1)$ for fuzzy DLs without truth qualified axioms. However, it is more efficient to deal with these two cases using different strategies.

Along the line of qualified statements, we think it is useful to further extend the language by allowing so-called *probability qualified* statements (cf. [10], page 222) and, thus, allowing statements such as "The probability of the temperature t at given place and time is around 35°C is likely", where both "around" and "likely" are fuzzy sets (with, e.g. a triangular membership function to model "around", and a right shoulder function to model "likely").

[4] Note that in Step 3 and 4 of the forest initialization, in case we consider a fuzzy assertion of the form $\langle \alpha, x_\alpha \rangle$, we may omit to add $x_\alpha \geq x_\alpha$ to the constraint set $\mathcal{C}_\mathcal{F}$.

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] 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.

[3] U. Straccia. Reasoning within fuzzy Description Logics. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.

[4] U. 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.

[5] U. Straccia. Managing uncertainty and vagueness in Description Logics, logic programs and Description Logic programs. In *Reasoning Web, 4th International Summer School, Tutorial Lectures*, Lecture Notes in Computer Science 5224, pages 54–103. Springer Verlag, 2008.

[6] F. Bobillo and U. Straccia. fuzzyDL: An expressive fuzzy Description Logic reasoner. In *Proceedings of the 2008 International Conference on Fuzzy Systems (FUZZ-IEEE 2008)*, pages 923–930. IEEE Computer Society, 2008.

[7] F. Bobillo, M. Delgado and J. Gómez-Romero. DeLorean: A reasoner for fuzzy OWL 1.1. In *Proc. of the 4th International Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2008)*, CEUR Workshop Proceedings 423, 2008.

[8] G. Stoilos, N. Simou, G. Stamou and S. Kollias. Uncertainty and the Semantic Web. *IEEE Intelligent Systems*, 21(5):84–87, 2006.

[9] U. Straccia and G. Visco. DL-Media: an ontology mediated multimedia information retrieval system. In *Proc. of the 4th Int. Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2008)*, CEUR Workshop Proceedings 423, 2008.

[10] G. J. Klir and B. Yuan. *Fuzzy sets and fuzzy logic: theory and applications*. Prentice-Hall, Inc., 1995.

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

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

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

[14] R. Hähnle. Many-valued logics and mixed integer programming. *Annals of Mathematics and Artificial Intelligence*, 3,4(12):231–264, 1994.

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

[16] F. Baader, E. Franconi, B. Hollunder, B. Nebel, and H. J. Profitlich. An empirical analysis of optimization techniques for terminological representation systems, or: Making KRIS get a move on. *Applied Artificial Intelligence* 4:109–132, 1994.

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

[18] T. Lukasiewicz and U. Straccia. Managing uncertainty and vagueness in Description Logics for the Semantic Web. *Journal of Web Semantics*, 6:291–308, 2008.

[19] U. Straccia and F. Bobillo. Mixed integer programming, general concept inclusions and fuzzy Description Logics. *Mathware & Soft Computing*, 14(3):247–259, 2007.

[20] F. Bobillo and U. Straccia. A fuzzy Description Logic with product t-norm. In *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE-07)*, pages 652–657. IEEE Computer Society, 2007.

[21] F. Bobillo and U. Straccia. Fuzzy Description Logics with general t-norms and datatypes. In *Fuzzy Sets and Systems*. To appear.

ISBN: 978-989-95079-6-8