

On Qualified Cardinality Restrictions in Fuzzy Description Logics under Łukasiewicz semantics

Fernando Bobillo

Dpt. of Computer Science and AI
University of Granada, Spain
fbobillo@decsai.ugr.es

Umberto Straccia

ISTI - CNR, Pisa, Italy
straccia@isti.cnr.it

Abstract

Fuzzy Description Logics have been proposed as a family of languages to describe vague or imprecise structured knowledge. This work deals with one of the less studied constructors, qualified cardinality restrictions, showing some counter-intuitive behaviours under Łukasiewicz semantics, and proposing a new semantics and the corresponding reasoning algorithm.

Keywords: Fuzzy Description Logics, Fuzzy Ontologies, Fuzzy Logic, Logic for the Semantic Web.

1 Introduction

Description Logics (DLs) [1] are a logical reconstruction of the so-called frame-based knowledge representation languages, with the aim of providing a simple well-established Tarski-style declarative semantics to capture the meaning of the most popular features of structured representation of knowledge. Nowadays, DLs have gained even more popularity due to their application in the context of the *Semantic Web* [2]. Indeed, the current standard language for specifying ontologies is the Web Ontology Language (OWL) [10], which comprises three sublanguages of increasing expressive power: OWL Lite, OWL DL and OWL Full. OWL 1.1 has been recently proposed as an extension of OWL [11]. The logical counterpart of OWL Lite, OWL DL and OWL 1.1 are the DLs $SHIF(\mathbf{D})$, $SHOIN(\mathbf{D})$ and $SRQIQ(\mathbf{D})$, respectively.

Fuzzy DLs have been proposed as an extension to classical DLs with the aim to deal with fuzzy, vague and imprecise concepts. Since the first work of J. Yen in 1991 [20], an important number of works on *fuzzy DLs* can be found in the literature (for a good survey we refer the reader to [9]). However, to date relative little work has been done in reasoning in fuzzy DLs with qualified cardinality restrictions (only [13, 3]). We argue that qualified cardinality restrictions are an important feature on DLs. For instance, such restrictions allow to define a father having two daughters as $Man \sqcap (\geq 2 hasRole.Woman)$. In fact, they are one of the main motivations for extending the current standard language OWL to OWL 1.1.

In this work we present a fuzzy DL with qualified cardinality restrictions under Łukasiewicz semantics. We will analyze the behaviour of the constructor, propose a new semantics and provide a reasoning algorithm.

In the remainder, we proceed as follows. Section 2 describes the fuzzy DL $ALCQ$ and discusses several semantics for qualified cardinality restrictions. Section 3 presents our reasoning algorithm. Finally, Section 4 presents some conclusions and ideas for future work.

2 Fuzzy Description Logics

The fuzzy DL that we will present in this section is an extension of [18]. The main ingredients of DLs are concepts, which denote unary predicates, and roles, which denote binary predicates. Then there are connectives which allow to construct complex concepts. For instance, if we use the concept *Human* to denote the set of

humans, and the concept *Male* to denote the set of male objects, the complex concept (conjunction) $Human \sqcap Male$ will denote the set of male humans. Moreover, if *hasChild* denotes a role then the concept $Human \sqcap \exists hasChild.Male$ will denote the set of humans having a male child, while $Human \sqcap \forall hasChild.Male$ will denote the set of humans such that if they have children then they have to be male.

2.1 Syntax

Let \mathbf{A} , \mathbf{R} and \mathbf{I} be non-empty finite and pairwise disjoint sets of *concept names* (denoted A), *roles names* (denoted R), and *individual names* (denoted a, b), respectively. The syntax of fuzzy \mathcal{ALCQ} concepts (denoted C, D) is as follows (n is 0 or a natural number):

$$C, D := \top \mid \perp \mid A \mid C \sqcap D \mid C \sqcup D \mid \neg C \mid \forall R.C \mid \exists R.C \mid \geq n R.C \mid \leq n R.C$$

where in $\geq n R.C$ we assume $n \geq 1$.

A *fuzzy knowledge base* (KB) $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$ consists of a fuzzy ABox \mathcal{A} and a fuzzy TBox \mathcal{T} .

A *fuzzy ABox* \mathcal{A} consists of a finite set of *fuzzy concept* and *fuzzy role assertion axioms* of the form $\langle a:C, \alpha \rangle$ and $\langle (a, b):R, \alpha \rangle$, where $\alpha \in (0, 1]$. Informally, $\langle \tau, \alpha \rangle$ constrains the membership degree of τ to be at least α . Hence, $\langle jim:YoungPerson, 0.2 \rangle$ says that *jim* is a *YoungPerson* with degree at least 0.2, while $\langle (jim, tom):hasFriend, 1 \rangle$, states that *jim* and *tom* are friends.

A *fuzzy TBox* \mathcal{T} is a finite set of *fuzzy General Concept Inclusion axioms* (GCIs) $\langle C \sqsubseteq D, \alpha \rangle$, where $\alpha \in (0, 1]$ and C, D are concepts. Informally, $\langle C \sqsubseteq D, \alpha \rangle$ states that all instances of concept C are instances of concept D to degree α , that is, the subsumption degree between C and D is at least α . For instance, $\langle Elephant \sqsubseteq Animal, 1 \rangle$ states that the class of elephants is a subclass of the class of animals. We write $C \equiv D$ as a shorthand of the two axioms $\langle C \sqsubseteq D, 1 \rangle$ and $\langle D \sqsubseteq C, 1 \rangle$.

2.2 Semantics

The main idea is that concepts and roles are interpreted as fuzzy subsets of an interpretation's do-

main. Therefore, axioms, rather than being “classical” evaluated (being either true or false), they are “many-valued” evaluated in $[0, 1]$.

In this paper, we will consider the Łukasiewicz family of fuzzy operators, which is defined as follows (where $\alpha, \beta \in [0, 1]$, \otimes is the t-norm, \oplus the t-conorm, \ominus the negation and \Rightarrow the implication, see [8] for a definition of these functions):

$$\begin{aligned} \ominus \alpha &= 1 - \alpha \\ \alpha \otimes \beta &= \max\{\alpha + \beta - 1, 0\} \\ \alpha \oplus \beta &= \min\{\alpha + \beta, 1\} \\ \alpha \Rightarrow \beta &= \min\{1, 1 - \alpha + \beta\} \end{aligned}$$

A *fuzzy interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a nonempty set $\Delta^{\mathcal{I}}$ (the *domain*) and of a *fuzzy interpretation function* $\cdot^{\mathcal{I}}$ that assigns: (i) to each concept C a function $C^{\mathcal{I}}: \Delta^{\mathcal{I}} \rightarrow [0, 1]$; (ii) to each role R a function $R^{\mathcal{I}}: \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0, 1]$; (iii) to each individual a an element in $\Delta^{\mathcal{I}}$.

We also impose the *unique names assumption* (UNA) over the individuals, i.e. if $a \neq b$ then $a^{\mathcal{I}} \neq b^{\mathcal{I}}$, where a, b are individuals (different individuals denote different objects of the domain).

The mapping $\cdot^{\mathcal{I}}$ is extended to complex concepts as specified in Table 1. $\cdot^{\mathcal{I}}$ is extended to the fuzzy axioms τ as follows:

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

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 \mathcal{E} of axioms E , we say that \mathcal{I} *satisfies* \mathcal{E} iff \mathcal{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 KB $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$, denoted $\mathcal{I} \models \mathcal{K}$, iff \mathcal{I} is a model of each component \mathcal{A} and \mathcal{T} , respectively.

An 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 .

Given \mathcal{K} and a fuzzy axiom τ of the forms $\langle x:C, \alpha \rangle$, $\langle (x, y):R, \alpha \rangle$ or $\langle C \sqsubseteq D, \alpha \rangle$, it is of interest to compute τ 's best lower degree value bound. The *greatest lower bound* of τ w.r.t. \mathcal{K} (denoted $glb(\mathcal{K}, \tau)$) is $glb(\mathcal{K}, \tau) = \sup\{\alpha \mid \mathcal{K} \models \langle \tau \geq \alpha \in [0, 1] \rangle\}$, where $\sup \emptyset = 0$. Determining the *glb* is called the *Best Degree Bound* (BDB) problem.

$$\begin{aligned}
\perp^{\mathcal{I}}(a) &= 0 \\
\top^{\mathcal{I}}(a) &= 1 \\
(C \sqcap D)^{\mathcal{I}}(a) &= C^{\mathcal{I}}(a) \otimes D^{\mathcal{I}}(a) \\
(C \sqcup D)^{\mathcal{I}}(a) &= C^{\mathcal{I}}(a) \oplus D^{\mathcal{I}}(a) \\
(\neg C)^{\mathcal{I}}(a) &= \ominus C^{\mathcal{I}}(a) \\
(\forall R.C)^{\mathcal{I}}(a) &= \inf_{b \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(a, b) \Rightarrow C^{\mathcal{I}}(b) \\
(\exists R.C)^{\mathcal{I}}(a) &= \sup_{b \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(a, b) \otimes C^{\mathcal{I}}(b) \\
(\geq n R.C)^{\mathcal{I}}(a) &= \sup_{b_1, \dots, b_n \in \Delta^{\mathcal{I}}} [(\otimes_{i=1}^n \{R^{\mathcal{I}}(a, b_i) \otimes C^{\mathcal{I}}(b_i)\}) \otimes (\otimes_{j < k} \{b_j \neq b_k\})] \\
(\leq n R.C)^{\mathcal{I}}(a) &= \inf_{b_1, \dots, b_{n+1} \in \Delta^{\mathcal{I}}} [(\otimes_{i=1}^{n+1} \{R^{\mathcal{I}}(a, b_i) \otimes C^{\mathcal{I}}(b_i)\}) \Rightarrow (\oplus_{j < k} \{b_j = b_k\})]
\end{aligned}$$

Table 1: Semantics of the complex fuzzy concepts.

Finally, the *best satisfiability bound* of a concept C , denoted $glb(\mathcal{K}, C)$, is defined as $\sup_{\mathcal{I}} \sup_{x \in \Delta^{\mathcal{I}}} \{C^{\mathcal{I}}(x) \mid \mathcal{I} \models \mathcal{K}\}$. Essentially, among all models \mathcal{I} of the KB, we are determining the maximal degree of truth that the concept C may have over all individuals $x \in \Delta^{\mathcal{I}}$.

2.3 Cardinality restrictions

D. Sánchez et al. have considered the fuzzy DL \mathcal{ALCQ} [13]. While their work is interesting since they allow the use of fuzzy quantifiers, for instance making possible to express that a customer *mostly* buys cheap items, reasoning becomes particularly harder. Moreover, their approach strongly relies on finite models, which is a problem if we want to extend the expressivity of the logic (for example, \mathcal{SHIF} does not have the finite model property, i.e., there are concepts which have only infinite models [7]).

Our semantics for cardinality restrictions was introduced in [14] as an extension of the semantics presented in [16], and derives from the classical case, by deriving the concept $(\geq n R.C)$ as the first-order formula

$$\exists x_1, \dots, x_n. (\bigwedge_{i < j} x_i \neq x_j) \wedge (\bigwedge_i R(x, x_i) \wedge C(x_i))$$

and assuming that $(\leq n R.C)$ is the same as $\neg(\geq n + 1 R.C)$. However, the semantics may be counter-intuitive, as the following example shows:

Example 2.1 Assume the following model: $((fernando, apple): likes)^{\mathcal{I}} = ((fernando, banana): likes)^{\mathcal{I}} = ((fernando, orange): likes)^{\mathcal{I}} = ((fernando, peach): likes)^{\mathcal{I}} = 0.5$;

$(apple:Fruit)^{\mathcal{I}} = (banana:Fruit)^{\mathcal{I}} = (orange:Fruit)^{\mathcal{I}} = (peach:Fruit)^{\mathcal{I}} = 1$; $apple^{\mathcal{I}}, banana^{\mathcal{I}}, orange^{\mathcal{I}}, peach^{\mathcal{I}}$ are different.

Then, $(\leq 1 likes.Fruit)^{\mathcal{I}}(fernando) = 1$. In this example, while one may expect fernando not to have more than 1 filler, he has 4 and, clearly, he could have many more fillers x_i as long as they satisfy $((fernando, x_i): likes)^{\mathcal{I}} + (x_i:Fruit)^{\mathcal{I}} \leq 1$.

In our opinion, the semantics of cardinality restrictions should satisfy the following properties:

Property 1 If $(\leq n R.C)^{\mathcal{I}}(a) = 1$ then $|\{b \mid (R(a, b)^{\mathcal{I}} \otimes C(b))^{\mathcal{I}} > 0\}| \leq n$.

Property 2 $\exists R.C \equiv \geq 1R.C$.

Property 3 $\leq n R.C \equiv \neg(\geq n + 1 R.C)$.

Property 1 requires that $\otimes_{i=1}^{n+1} \{R^{\mathcal{I}}(a, b_i) \otimes C^{\mathcal{I}}(b_i)\} > 0$ if (and only if) $R^{\mathcal{I}}(a, b_i) \otimes C^{\mathcal{I}}(b_i) > 0$ for every $i \in \{1, \dots, n + 1\}$. However, Łukasiewicz t-norm does not verify this property. As a solution, we propose to use for cardinality restrictions the semantics in Table 2, which combines Łukasiewicz and Gödel t-norms in order to verify Properties 1 2 and 3; solving the counter-intuitive effect of Example 2.1 (in fact, under this semantics $(\leq 1 likes.Fruit)^{\mathcal{I}}(fernando) = 0.5$).

As a final comment, S. Calegari et al. have introduced another semantics for unqualified cardinality restrictions (a special case where $C = \top$ and it is syntactically omitted). In their proposal, $\geq nR$ and $\leq nR$ are crisp concepts [4]. $\geq nR$ is actually interpreted as “the individual has at least $n R$ -successors with degree greater than 0”, and

$$\begin{aligned}
(\geq n R.C)^{\mathcal{I}}(a) &= \sup_{b_1, \dots, b_n \in \Delta^{\mathcal{I}}} [\min_{i=1}^n \{R^{\mathcal{I}}(a, b_i) \otimes C^{\mathcal{I}}(b_i)\} \otimes (\otimes_{j < k} \{b_j \neq b_k\})] \\
(\leq n R.C)^{\mathcal{I}}(a) &= \inf_{b_1, \dots, b_{n+1} \in \Delta^{\mathcal{I}}} [\min_{i=1}^{n+1} \{R^{\mathcal{I}}(a, b_i) \otimes C^{\mathcal{I}}(b_i)\} \Rightarrow (\oplus_{j < k} \{b_j = b_k\})]
\end{aligned}$$

Table 2: New semantics for cardinality restrictions.

$\leq nR$ is interpreted dually. However, this semantics may also lead to counter-intuitive examples. For instance, Property 2 does not hold.

3 Reasoning Algorithm

Our reasoning algorithm extends [18]. The basic idea of our reasoning algorithm is as follows. Consider $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$. In order to solve the BDB problem, we combine appropriate DL tableaux rules with methods developed in the context of *Many-Valued Logics* (MVLs) [5].

In order to determine e.g. $glb(\mathcal{K}, a:C)$, we consider an expression of the form $\langle a:\neg C, 1-x \rangle$ (informally, $\langle a:C \leq x \rangle$), where x is a $[0, 1]$ -valued variable. Then we construct a tableaux for $\mathcal{K} = \langle \mathcal{A} \cup \{\langle a:\neg C, 1-x \rangle\}, \mathcal{T} \rangle$ in which the application of satisfiability preserving rules generates new fuzzy assertion axioms together with *inequations* over $[0, 1]$ -valued variables. These inequations have to hold in order to respect the semantics of the DL constructors. Finally, to determine the greatest lower bound, we *minimize* the original variable x such that all constraints are satisfied¹. Similarly, for $C \sqsubseteq D$, we can compute $glb(\mathcal{K}, C \sqsubseteq D)$ as the minimal value of x such that $\mathcal{K} = \langle \mathcal{A} \cup \{\langle a:C \sqcap \neg D, 1-x \rangle\}, \mathcal{T} \rangle$ is satisfiable, where a is new individual. Hence, 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 $\langle \mathcal{A} \cup \{\langle a:C, x \rangle\}, \mathcal{T} \rangle$ is satisfiable. In Łukasiewicz logic we end up with a *bounded Mixed Integer Linear Program* (bMILP) optimization problem [12].

Note that $(\neg \exists R.C)^{\mathcal{I}} = (\forall R.\neg C)^{\mathcal{I}}$, $(\neg \forall R.C)^{\mathcal{I}} = (\exists R.\neg C)^{\mathcal{I}}$, $\neg(\leq n R.C)^{\mathcal{I}} = (\geq n+1 R.C)^{\mathcal{I}}$ and $\neg(\geq n R.C)^{\mathcal{I}} = (\leq n-1 R.C)^{\mathcal{I}}$. This al-

¹Informally, 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, $(a:C)^{\mathcal{I}} \geq \bar{n}$ has to hold. Which means that $glb(\mathcal{K}, (a:C)) = \bar{n}$

lows us to transform concept expressions into a semantically equivalent *Negation Normal Form* (NNF), which is obtained by pushing in the usual manner negation on front of concept names only. With $nnf(C)$ we denote the NNF of concept C .

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 $\langle \tau, l \rangle$, where l is a linear expression over variables in \mathbf{V} , \mathbf{W} and rational 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.

Given $\mathcal{K} = \langle \mathcal{A}, \mathcal{T} \rangle$, 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, $\mathcal{L} : \mathbf{S} \times sub(\mathcal{K}) \rightarrow [0, 1]$ maps each element and concept, to a membership degree (the degree of the element being an instance of the concept), and $\mathcal{E} : \mathbf{R}_{\mathcal{K}} \times (\mathbf{S} \times \mathbf{S}) \rightarrow [0, 1]$ 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 $\mathcal{V} : \mathbf{I}_{\mathcal{A}} \rightarrow \mathbf{S}$ 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 the following conditions:

1. $\mathcal{L}(s, \perp) = 0$ and $\mathcal{L}(s, \top) = 1$ for all $s \in \mathbf{S}$,
2. If $\mathcal{L}(s, \neg A) \geq \alpha$, then $\mathcal{L}(s, A) \leq \ominus \alpha$.
3. If $\mathcal{L}(s, C \sqcap D) \geq \alpha$, then $\mathcal{L}(s, C) \otimes \mathcal{L}(s, D) \geq \alpha$.
4. If $\mathcal{L}(s, C \sqcup D) \geq \alpha$, then $\mathcal{L}(s, C) \oplus \mathcal{L}(s, D) \geq \alpha$.
5. If $\mathcal{L}(s, \forall R.C) \geq \alpha$, then $\mathcal{E}(R, \langle s, t \rangle) \leq \mathcal{L}(t, C) + 1 - \alpha$ for all $t \in \mathbf{S}$.
6. If $\mathcal{L}(s, \exists R.C) \geq \alpha$, then there exists $t \in \mathbf{S}$ such that $\mathcal{E}(R, \langle s, t \rangle) \otimes \mathcal{L}(t, C) \geq \alpha$.
7. If $\langle C \sqsubseteq D, \alpha \rangle \in \mathcal{T}$, then $\mathcal{L}(s, C) \leq \mathcal{L}(s, D) + 1 - \alpha$, for all $s \in \mathbf{S}$.
8. If $\mathcal{L}(s, \geq n R.C) \geq \alpha$, then there are $b_1, \dots, b_n \in \mathbf{S}$ such that for all $1 \leq i \leq n$, $\mathcal{E}(R, \langle s, b_i \rangle) \otimes \mathcal{L}(b_i, C) \geq \alpha$.

9. If $\mathcal{L}(s, \leq n R.C) \geq \alpha$, then $\mathcal{L}(s, \geq n + 1R.C) \leq 1 - \alpha$.
10. If $\langle a:C, \alpha \rangle \in \mathcal{A}$, then $\mathcal{L}(\mathcal{V}(a), C) \geq \alpha$.
11. If $\langle (a, b):R, \alpha \rangle \in \mathcal{A}$, then $\mathcal{E}(R, \langle \mathcal{V}(a), \mathcal{V}(b) \rangle) \geq \alpha$.

It can be shown that:

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

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

A completion-forest \mathcal{F} for a fuzzy KB \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 $C \in \text{sub}(\mathcal{K})$, and l is either a rational, a variable x , or a negated variable, i.e. of the form $1 - x$, where x is a variable (the intuition here is that v is an instance of C to degree equal or greater than of the evaluation of l).

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

The forest has associated a set $\mathcal{C}_{\mathcal{F}}$ of constraints of the form $c \leq c', c = c', x_i \in [0, 1], y_i \in \{0, 1\}$, on the variables occurring the node labels and edge labels. c, c' are linear expressions. If nodes v and w are connected by an edge $\langle v, w \rangle$ with $\langle R, l \rangle$ occurring in $\mathcal{L}(\langle v, w \rangle)$, then w is called an R_l -successor of v and v is called an R_l -predecessor of w . A node v is an R -successor (resp. R -predecessor) of w if it is an R_l -successor (resp. R_l -predecessor) of w for some role R . As usual, *ancestor* is the transitive closure of *predecessor*.

We say that two non-root nodes v and w are *equivalent*, denoted $\mathcal{L}(v) \approx \mathcal{L}(w)$, if

$\mathcal{L}(v) = [\langle C_1, l_1 \rangle, \dots, \langle C_n, l_k \rangle]$, $\mathcal{L}(w) = [\langle C_1, l'_1 \rangle, \dots, \langle C_n, l'_k \rangle]$, and for all $1 \leq i \leq k$, either both l_i and l'_i are variables, or both l_i and l'_i are negated variables or both l_i and l'_i are the same rational in $[0, 1]$ (the intuition here is that v and w share the same properties).

A node v is *directly blocked* if and only if none of its ancestors are blocked, it is not a root node, and it has an ancestor w such that $\mathcal{L}(v) \approx \mathcal{L}(w)$. In this case, w directly blocks v . A node v is *blocked* if and only if it is directly blocked, or if one of its predecessors 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), or v is a successor of a node w and $\mathcal{L}(\langle w, v \rangle) = \emptyset$.

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 $\langle a_i:C_i, n \rangle \in \mathcal{A}$, (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 $\langle R_i, n \rangle$ (iii) $a \neq b$ for every pair of root nodes a, b . \mathcal{F} is then expanded by repeatedly applying the completion rules described below. The completion-forest is complete when none of the completion rules are applicable. Then, the bMILP problem on the set of constraints $\mathcal{C}_{\mathcal{F}}$ is solved.

We assume a procedure $Merge(v, w_1, w_2)$ composed by the following steps, where \cup is interpreted as “append to the end the list”:

1. $\mathcal{L}(w_1) = \mathcal{L}(w_1) \cup \mathcal{L}(w_2)$,
2. $\mathcal{L}(\langle v, w_1 \rangle) = \mathcal{L}(\langle v, w_1 \rangle) \cup \mathcal{L}(\langle v, w_2 \rangle)$,
3. $\mathcal{L}(\langle v, w_2 \rangle) = \emptyset$,
4. set $u \neq w_1$ for all u with $u \neq w_2$.

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

With x_{τ} we denote the variable associated to the *atomic assertion* τ of the form $a:A$ or $(a, b):R$. x_{τ} will take the truth value associated to τ , 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 $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{v:A} \geq l\} \cup \{x_{v:A} \in [0, 1]\}$.

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

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

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

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

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

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

(\sqsubseteq). If $\langle C \sqsubseteq D, \alpha \rangle \in \mathcal{T}$ and v is a node to which this rule has not yet been applied then append $\langle nnf(\neg C), 1 - x_1 \rangle$ and $\langle D, x_2 \rangle$ to $\mathcal{L}(v)$, and $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_1 \leq x_2 + 1 - \alpha, x_i \in [0, 1]\}$, where x_i are new variables.

(ch). If $\langle \bowtie n R.C, l \rangle \in \mathcal{L}(v)$, $\bowtie \in \{\leq, \geq\}$ and there is an R -successor w of v such that the rule has not yet been applied to v and w , then append $\mathcal{L}(w) = \mathcal{L}(w) \cup \{\langle C, x \rangle, \langle \neg C, 1 - x \rangle\}$ and $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x \in [0, 1]\}$, where x is a new variable.

(\leq). If $\langle \leq n R.C, l \rangle \in \mathcal{L}(v)$ and there are $n + 1$ R -successors w_1, \dots, w_{n+1} of v with $\langle C, l_i \rangle \in \mathcal{L}(w_i)$, $\langle R, r_i \rangle \in \mathcal{L}(\langle v, w_i \rangle)$, then non-deterministically apply one of the following sub-rules:

1. $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{l = 0\}$.
2. Append $\langle \neg C_i, 1 - x_i \rangle$ to $\mathcal{L}(w_i)$ and $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{(v,w_i):R} + x_i + y_i \leq 2 - l, y_1 + \dots + y_{n+1} = n, x_i \in [0, 1], y_i \in \{0, 1\}\}$, where x_i, y_i are new variables.
3. For every pair of individuals w_i and w_j , $1 \leq i < j \leq n$, such that w_j is not an ancestor of w_i and not $w_i \neq w_j$, apply $Merge(v, w_i, w_j)$.
4. If for all pairs w_i, w_j , $1 \leq i < j \leq n$, $w_i \neq w_j$, then there is an inconsistency, so add $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{0 = 1\}$.

(\geq). If $\langle \geq n R.C, l \rangle \in \mathcal{L}(v)$, v is not blocked, the rule has not yet been applied to v , then: create n new nodes $w_1 \dots w_n$ with $\langle R, r_i \rangle$ to $\mathcal{L}(\langle v, w_i \rangle)$,

$\langle C, c_i \rangle$ to $\mathcal{L}(w_i)$, $w_i \neq w_j$ and $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{y_i \leq 1 - l, r_i \leq 1 - y_i, c_i \leq 1 - y_i, r_i + c_i = l + 1 - y_i, c_i \in [0, 1], r_i \in [0, 1], y_i \in \{0, 1\}\}$, where c_i, r_i, y_i are new variables.

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

Let us explain the intuition behind the new rules.

(\geq) creates new n R -successors in case they do not exist, in such a case that the semantics of the concept constructor is satisfied ($R^{\mathcal{I}}(v, w_i) \otimes C^{\mathcal{I}}(w_i) \geq l$). (ch) states that w belongs to C and $\neg C$ to some degree, and we know that $C^{\mathcal{I}}(x) + (\neg C)^{\mathcal{I}}(x) = C^{\mathcal{I}}(x) + 1 - C^{\mathcal{I}}(x) = 1$. Note that, as opposed to the crisp case, the rule is deterministic. (\leq) is more tricky. It guarantees that there do not exist $n + 1$ R -successors, which leads to several cases:

- The first one is quite simple, if $l = 0$ then we have $\langle \leq n R.C, 0 \rangle$, which is a tautology.
- The second possibility guarantees that the minimum over all $R^{\mathcal{I}}(v, w_i) \otimes C^{\mathcal{I}}(w_i) \geq l$ is less or equal than $1 - l$. That is, there is an R -successor w_i satisfying this. The constraints on the control variables y_i require that exactly one of them takes the value 0. If a control variable takes the value 1, it does not impose any restriction. Otherwise, if $y_i = 0$ then $x_{(v,w_i):R} \otimes x_i \leq 1 - l$ and, together with the assertion $\langle \neg C_i, 1 - x_i \rangle$ to $\mathcal{L}(v)$, this guarantees that $R^{\mathcal{I}}(v, w_i) \otimes C^{\mathcal{I}}(w_i) \leq 1 - l$.
- The third case covers the case where two successors may be interpreted as the same individual, so we merge them into one equivalent individual.
- The latter case simulates the case where no individual can be merged (for all possible pairs of individuals, they are required to be different) but $l \neq 0$, so consequently the KB is inconsistent.

An important aspect is that without the $\leq n R.C$ construct, the generated tableaux is deterministic and, thus, just *one* bMILP problem has to be

solved [17]. This is no longer true once we introduce cardinality restrictions: due to the (\leq) rule, several bMILP problems may need to be solved in order to determine whether the KB is satisfiable or not. Furthermore, in order to find the minimum solution, in fact, it is necessary to solve *all* of them.

Example 3.1 Let us show that $\mathcal{K} = \langle \mathcal{A}, \emptyset \rangle$, with $\mathcal{A} = \{ \langle a: \geq 3 R.A, 1 \rangle, \langle a: \leq 1 R.B, 1 \rangle, \langle a: \leq 1 R.\neg B, 1 \rangle \}$ is unsatisfiable (adapted from an example proposed in [19]).

To start with, we construct a forest \mathcal{F} with a root node a with $\mathcal{L}(a) = [\langle \geq 3 R.A, 1 \rangle, \langle \leq 1 R.B, 1 \rangle, \langle \leq 1 R.\neg B, 1 \rangle]$.

Now we apply (\geq) rule and create 3 new R -successors b_1, b_2, b_3 such that $\mathcal{L}(b_i) = [\langle A, c_i \rangle]$, $\mathcal{L}(\langle a, b_i \rangle) = \langle R, r_i \rangle$ for $1 \leq i \leq 3$. We also add $\{ b_1 \neq b_2, b_1 \neq b_3, b_2 \neq b_3 \}$ to $\mathcal{C}_{\mathcal{F}}$, and also some constraints ensuring that $c_i \otimes r_i = 1$, that is, $c_i = r_i = 1$.

Next, we apply (ch) rule to b_1, b_2, b_3 , so we add $\{ \langle B, x_i \rangle, \langle \neg B, 1 - x_i \rangle \}$ to $\mathcal{L}(b_i)$ and $\{ x_i \in [0, 1] \}$ to $\mathcal{C}_{\mathcal{F}}$, for $i = \{1, 2, 3\}$.

Now we apply (\leq) rule to some pair of R -successors of a . Without loss of generality, we consider the assertion $\langle \leq 1 R.C, 1 \rangle$, individuals b_i and $b_j, C \in \{B, \neg B\}$, $i, j \in \{1, 2, 3\}$, $i \neq j$.

Clearly, the first possibility of the rule generates an inconsistency ($1 \neq 0$), and also the fourth by definition. The third possibility is not applicable since $\mathcal{C}_{\mathcal{F}}$ contains $b_i \neq b_j$ due to the application of (\geq) rule. Thus, it only remains to check the second possibility. We append $\langle \neg C, 1 - x_i \rangle$ to $\mathcal{L}(b_i)$, $\langle \neg C, 1 - x_j \rangle$ to $\mathcal{L}(b_j)$ and $\{ x_{(a, b_i)}:R + x_i + y_i \leq 1, x_{(a, b_j)}:R + x_j + y_j \leq 1, y_i + y_j = 1, x_i \in [0, 1], x_j \in [0, 1], y_i \in \{0, 1\}, y_j \in \{0, 1\} \}$ to $\mathcal{C}_{\mathcal{F}}$. Hence, one of the control variables should be equal to 0, and the other equal to 1.

Without loss of generality, assume that $y_j = 0$. We also have that $\mathcal{L}(\langle a, b_j \rangle) = \langle R, r_j \rangle$. Later, (R) rule will create $x_{(a, b_j)}:R \geq r_j$. But we have seen that r_j should be equal to 1 (the constraints in the application of (\geq) rule force that $c_j \otimes r_j = 1$). So, in order to satisfy $x_{(a, b_j)}:R + x_j \leq 1$ without contradictions, $x_j = 0$. Since we have $\langle \neg C, 1 - x_j \rangle$ in $\mathcal{L}(b_j)$, the conclusion is that (in-

formally) $\langle \neg C, 1 \rangle$ belongs to $\mathcal{L}(b_j)$.

Then, (\leq) rule is applied to $\langle \leq 1 R.C, 1 \rangle$ and b_i, b_k , $k \in \{1, 2, 3\} \setminus \{j\}$. Using the same reasoning as before, $\langle \neg C, 1 \rangle$ belongs to $\mathcal{L}(b_p)$ for some $p \in \{i, k\}$. Next, we apply (\leq) rule to $\langle \leq 1 R.\neg C, 1 \rangle$ and some pair of R -successors of a . Using the same reasoning as before, it will append $\langle C, 1 \rangle$ to the set \mathcal{L} of two of the R -successors.

Summing up, the algorithm appends $\langle \neg C, 1 \rangle$ to the set \mathcal{L} of two successors and $\langle C, 1 \rangle$ to the set \mathcal{L} of two successors, so, since there are 3 successors, there will be at least one successor b such that both $\langle \neg C, 1 \rangle$ and $\langle C, 1 \rangle$ belong to its set \mathcal{L} . But the application of the (A), (\bar{A}) rules generate $x_{b:C} \leq 0$ and $x_{b:C} \geq 1$ respectively, which is a contradiction. Hence, \mathcal{K} is unsatisfiable.

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

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} where $\mathcal{C}_{\mathcal{F}}$ has a solution, then \mathcal{K} has 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.

In order to proof the completeness, we rely on the existence of witnessed models, which has been proved to hold for Łukasiewicz logic [6].

Finally, we note that the algorithm could easily be adapted to work with the semantics of cardinality restrictions proposed in Table 1, by modifying (\geq) and (\leq) rules so they consider Łukasiewicz t-norm instead of a minimum.

4 Conclusions

In this paper we have presented a reasoning algorithm for fuzzy DLs under Łukasiewicz semantics with a novel semantics for qualified cardinality restrictions. We have shown that adding cardinality restrictions makes the reasoning more difficult because the algorithm needs to solve several optimization problems.

Clearly, our approach can easily be extended to work with concept modifiers and fuzzy concrete domains as in [15]. The fuzzy operators of the Zadeh family can be represented using Łukasiewicz logic, so our use of cardinality restrictions is more general than [3, 14]. Finally, we argue that it could be combined with the algorithm for fuzzy $SHIF(\mathbf{D})$ [17] in order to obtain the very expressive DL $SHIQ(\mathbf{D})$.

Acknowledgements

The research of Fernando Bobillo has been partially supported by a FPU scholarship and project TIN2006-15041-C04-01, both from the Spanish Ministerio de Educación y Ciencia.

References

- [1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [2] T. Berners-Lee, J. Hendler and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.
- [3] F. Bobillo, M. Delgado and J. Gómez-Romero. Optimizing the crisp representation of the fuzzy description logic $SR\mathcal{O}\mathcal{I}\mathcal{Q}$. In *Proceedings of the 3rd ISWC Workshop on Uncertainty Reasoning for the Semantic Web (URSW 2007)*, CEUR Workshop Proceedings, 2007.
- [4] S. Calegari and D. Ciucci. Fuzzy Ontology, Fuzzy Description Logics and Fuzzy-OWL. In *Proceedings of WILF 2007*, LNCS 4578, pp. 118–126. Springer Verlag, 2007.
- [5] R. Hähnle. Advanced many-valued logics. In D. M. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic, 2nd Edition*, volume 2. Kluwer, 2001.
- [6] P. Hájek. Making fuzzy description logics more general. *Fuzzy Sets and Systems*, 154 (1), 1–15, 2005.
- [7] I. Horrocks, U. Sattler and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.
- [8] G. Klir and B. Yuan. *Fuzzy sets and fuzzy logic: Theory and applications*. Prentice-Hall, 1995.
- [9] T. Lukasiewicz and U. Straccia. An overview of uncertainty and vagueness in description logics for the semantic web. Technical report INF-SYS RR-1843-06-07, Institut für Informationssysteme, Technische Universität Wien, 2006.
- [10] D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. <http://www.w3.org/TR/owl-features>, 2004.
- [11] P. F. Patel-Schneider and I. Horrocks. OWL 1.1 Web Ontology Language Overview. <http://www.w3.org/Submission/owl111-overview>, 2006.
- [12] H. Salkin and M. Kamlesh. *Foundations of Integer Programming*. North-Holland, 1988.
- [13] D. Sánchez and A. G. B. Tettamanzi. Fuzzy quantification in fuzzy description logics. In E. Sanchez, editor, *Fuzzy Logic and the Semantic Web, Capturing Intelligence*, chapter 8, pp. 135–159. Elsevier, 2006.
- [14] G. Stoilos, G. Stamou and J. Z. Pan. Handling imprecise knowledge with fuzzy description logic. In *Proceedings of the International Workshop on Description Logics (DL 06)*, 2006.
- [15] U. Straccia. Description logics with fuzzy concrete domains. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI-05)*, pp. 559–567. AUAI Press, 2005.
- [16] U. Straccia. A fuzzy description logic for the semantic web. In E. Sanchez, editor, *Fuzzy Logic and the Semantic Web, Capturing Intelligence*, chapter 4, pp. 73–90. Elsevier, 2006.
- [17] U. Straccia. Reasoning in $\mathcal{L}\text{-}SHIF$: an expressive fuzzy description logic under Łukasiewicz semantics. Technical report ISTI-018/2007, ISTI - CNR, 2007.
- [18] U. Straccia and F. Bobillo. Mixed integer programming, general concept inclusions and fuzzy description logics. In *Mathware & Soft Computing* 14(3):247–259.
- [19] S. Tobies. Complexity results and practical algorithms for logics in knowledge representation. PhD thesis, RWTH Aachen, Germany, 2001.
- [20] J. Yen. Generalizing term subsumption languages to fuzzy logic. In *Proc. of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, pp. 472–477, 1991.
- [21] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.