

# Extending Datatype Restrictions in Fuzzy Description Logics

Fernando Bobillo  
 Dept. of Computer Science and Systems  
 Engineering, University of Zaragoza, Spain  
 fbobillo@unizar.es

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

## Abstract

*Fuzzy Description Logics (DLs) are a family of logics which allow the representation of (and the reasoning within) structured knowledge affected by vagueness. Although a relatively important amount of work has been carried out in the last years, little attention has been given to the role of datatypes in fuzzy DLs. This paper presents a fuzzy DL with three kinds of extended datatype restrictions, together with the necessary rules to reason with them.*

## 1. Introduction

*Description Logics (DLs) are a family of logics which allow the representation of (and the reasoning with) structured knowledge [1]. A natural extension for dealing with vague knowledge, very common in the real-world, are fuzzy DLs, which have proved to be useful in several applications.*

Although a relatively important amount of work has been carried out [5], little attention has been given to the role of datatypes in fuzzy DLs. Previous work allow simple datatype restrictions, for instance  $(\geq \text{hasAge } 18)$  (denoting people of age) [4]. This paper introduces extended datatype restrictions that allow, for example, to specify:

- $\langle \text{john} : (= \text{hasAge } x) \rangle$  and  $\langle \text{tom} : (= \text{hasAge } x) \rangle$ , stating that John and Tom have the same age.
- $\langle \text{john} : (\geq \text{hasAge } \text{tri}(20, 40, 60)) \rangle$ , stating that John's age is at least *about 40* years, with *about 40* being defined with a triangular function.
- $\langle \text{john} : (= \text{hasGrossSalary } (0.8 \cdot \text{hasNetSalary} + 0.2 \cdot \text{paysTaxes})) \rangle$ , meaning the 20% of John's gross salary is payed as taxes.

The remainder of the paper is organized as follows. Section 2 presents a fuzzy extension of the DL  $\mathcal{ALCF}(\mathbf{D})$ , and then Section 3 provides a reasoning algorithm for the new datatype restrictions. Finally, Section 4 sets out some conclusions and ideas for future work.

## 2. A fuzzy DL with datatype restrictions

In this section we define a fuzzy extension of the DL  $\mathcal{ALCF}(\mathbf{D})$  including what we have called *extended datatype restrictions*.  $\mathcal{ALCF}(\mathbf{D})$  extends the well known DL  $\mathcal{ALC}$  with *functional roles* (also called attributes or features) and *concrete domains* allowing to deal with *datatypes* such as reals, integer and strings.

Let  $\mathbf{A}$ ,  $\mathbf{R}_A$ ,  $\mathbf{F}_c$ ,  $\mathbf{I}$ , and  $\mathbf{I}_c$  be non-empty enumerable and pair-wise disjoint sets of *concept names* (denoted  $A$ ), *abstract role names* (denoted  $R$ ), *concrete feature names* (denoted  $t$ ), *abstract individual names* (denoted  $a$ ), and *concrete individual names* (denoted  $c$ ).

$\mathcal{ALC}$  concepts can be inductively built as follows:

$$C \rightarrow \top \mid \perp \mid A \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \neg C \mid \forall R.C \mid \exists R.C$$

Example:  $\text{Man} \sqcap \exists \text{hasChild} . (\text{Man} \sqcup \text{Woman})$  represents a father, a person which has a child (a son or a daughter).

In  $\mathcal{ALCF}(\mathbf{D})$  the above syntax is increased with extended datatype restrictions:

$$\begin{aligned} C &\rightarrow DR \\ DR &\rightarrow (\geq t x) \mid (\leq t x) \mid (= t x) \mid (\geq t FE) \mid (\leq t FE) \mid \\ & \quad (= t FE) \mid (\geq t NE) \mid (\leq t NE) \mid (= t NE) \\ NE &\rightarrow \text{tri}(a, b, c) \mid NE_1 + \dots + NE_n \mid NE_1 - NE_2 \mid \\ & \quad NE_1 \cdot \dots \cdot NE_n \mid NE_1 / NE_2 \mid r \\ FE &\rightarrow FE_1 + \dots + FE_n \mid FE_1 - FE_2 \mid FE \cdot t \mid r \mid t \end{aligned}$$

where  $x$  is a variable,  $\text{tri}(a, b, c)$  is triangular membership function with parameters  $a \leq b \leq c \in \mathbb{R}$ , and  $r$  is a number.  $DR$  is called a datatype restriction,  $NE$  is a fuzzy number expression, and  $FE$  (or, more formally,  $FE(t_1, \dots, t_n)$ ) is an arithmetic expression over concrete features.

Addition, subtraction and scalar product of features are defined in the usual away. Let  $\text{tri}(a_i, b_i, c_i)$  denote the fuzzy number expression  $NE_i$  and observe that  $r = \text{tri}(r, r, r)$ . Addition, subtraction, product and division of fuzzy number expressions are defined as follows:

$$\begin{aligned}
NE_1 + \dots + NE_n &= (\sum_{i=1}^n a_i, \sum_{i=1}^n b_i, \sum_{i=1}^n c_i) \\
NE_1 - NE_2 &= (a_1 - c_2, b_1 - b_2, c_1 - a_2) \\
NE_1 \dots \dots NE_n &= (\prod_{i=1}^n a_i, \prod_{i=1}^n b_i, \prod_{i=1}^n c_i) \\
NE_1/NE_2 &= (a_1/c_2, b_1/b_2, c_1/a_2)
\end{aligned}$$

An  $\mathcal{ALCF}(\mathbf{D})$  fuzzy knowledge base (fuzzy KB)  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  consists of a fuzzy TBox  $\mathcal{T}$ , and a fuzzy ABox  $\mathcal{A}$ , with axioms about concepts and individuals, respectively.

In general, a fuzzy TBox  $\mathcal{T}$  is a finite set of fuzzy concept inclusion axioms  $\langle C \sqsubseteq D, \alpha \rangle$ , with  $\alpha \in (0, 1]$ . Informally,  $\langle C \sqsubseteq D, \alpha \rangle$  states that all instances of concept  $C$  are instances of concept  $D$  to degree  $\alpha$ , i.e. the subsumption degree between  $C$  and  $D$  is at least  $\alpha$ . However, for computational reasons, we will restrict to *acyclic* TBoxes containing axioms of the form  $\langle A \sqsubseteq C, n \rangle$  and  $\langle C \sqsubseteq A, 1 \rangle$ . This allows to reason using the *lazy expansion* optimization technique [4].

A fuzzy ABox  $\mathcal{A}$  consists of a finite set of fuzzy concept and fuzzy role assertions of the form  $\langle a:C, \alpha \rangle$  and  $\langle (a, b):R, \alpha \rangle$ , where  $a, b$  are abstract individuals,  $C$  is a concept and  $R$  is an abstract role. Informally,  $\langle \tau, \alpha \rangle$  constrains the membership degree of  $\tau$  to be at least  $\alpha$ .

## 2.1. $\mathcal{ALCF}(\mathbf{D})$ semantics

In  $\mathcal{ALCF}(\mathbf{D})$  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 assigns:

1. to each concept  $C$  a function  $C^{\mathcal{I}}: \Delta^{\mathcal{I}} \rightarrow [0, 1]$ ;
2. to each role  $R$  a function  $R^{\mathcal{I}}: \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0, 1]$ ;
3. to each concrete feature  $t$  a partial function  $t^{\mathcal{I}}: \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}} \rightarrow \{0, 1\}$  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;
4. to each abstract individual  $a$  an element in  $\Delta^{\mathcal{I}}$ ;
5. to each concrete individual  $c$  an element in  $\Delta_{\mathbf{D}}$ .

Notice that we force features (and, consequently, datatype restrictions) to be crisp. In our opinion, the notion of functionality induces a crisp interpretation over a concrete feature and, hence saying that e.g., the degree of truth of  $\text{hasAge}(x, 18)$  is 0.5 is rather unrealistic [4].

Due to space limitations, we will focus on *real datatypes*. Hence, we assume that  $x$  ranges over  $\mathbb{R}$ ,  $r \in \mathbb{R}$ , and every feature  $t: [k_1, k_2] \subseteq \mathbb{R} \rightarrow \{0, 1\}$ .

The fuzzy interpretation function is extended to the constructors of fuzzy  $\mathcal{ALC}$  and to datatype restrictions as shown in Tables 2.1 and 2.1, respectively.

Given a relation  $(z, c): t$ , we say that  $c$  is a *t-filler* for  $z$ .

$$\begin{aligned}
\perp^{\mathcal{I}}(z) &= 0 \\
\top^{\mathcal{I}}(z) &= 1 \\
(C_1 \sqcap C_2)^{\mathcal{I}}(z) &= C_1^{\mathcal{I}}(z) \otimes C_2^{\mathcal{I}}(z) \\
(C_1 \sqcup C_2)^{\mathcal{I}}(z) &= C_1^{\mathcal{I}}(z) \oplus C_2^{\mathcal{I}}(z) \\
(\neg C)^{\mathcal{I}}(z) &= \ominus C^{\mathcal{I}}(z) \\
(\forall R.C)^{\mathcal{I}}(z) &= \inf_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(z, y) \Rightarrow C^{\mathcal{I}}(y) \\
(\exists R.C)^{\mathcal{I}}(z) &= \sup_{y \in \Delta^{\mathcal{I}}} R^{\mathcal{I}}(z, y) \otimes C^{\mathcal{I}}(y)
\end{aligned}$$

Table 1. Semantics of fuzzy concepts in fuzzy  $\mathcal{ALC}$

- The intuition behind  $(\geq t x)^{\mathcal{I}}(z)$  is that there exists a  $t$ -filler  $c$  for  $z$  such that the value of  $c$  is greater or equal than the value of the variable  $x$ .
- In  $(\geq t FE)^{\mathcal{I}}(z)$ , there are a  $t$ -filler  $c$  and  $t_i$ -fillers  $c_i$  ( $i \in \{1, \dots, n\}$ ) for  $z$  such that the value of  $c$  is at least the evaluation of expression  $FE$ .
- In  $(\geq t NE)^{\mathcal{I}}(z)$ , there exist a  $t$ -filler  $c$  and a concrete individual  $c'$  such that  $c'$  is a  $NE$  and the value of  $c$  is greater or equal than the value of  $c'$ . We say that  $c'$  is a *representative* of the values less or equal than  $c$ . The semantics is the result of viewing “ $z$  is  $(\geq NE)$ ” as the open first-order formula  $\exists y.(z \geq y) \wedge NE(y)$ .

Note that the following equivalences hold:

$$\begin{aligned}
(= t x) &= (\leq t x) \sqcap (\geq t x) \\
(= t FE) &= (\leq t FE) \sqcap (\geq t FE) \\
(= t NE) &= (\leq t NE) \sqcap (\geq t NE) \\
(= t NE) &= \exists t.NE \\
\neg(= t NE) &= \forall t.(\neg NE)
\end{aligned}$$

$\cdot^{\mathcal{I}}$  is also extended to the axioms as specified below:

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

$\mathcal{I}$  satisfies an axiom of the form  $\langle \tau, \alpha \rangle$  iff  $\tau^{\mathcal{I}} \geq \alpha$ , and  $\mathcal{I}$  satisfies a fuzzy KB  $\mathcal{K}$  iff  $\mathcal{I}$  satisfies each axiom in  $\mathcal{K}$ .

In fuzzy DLs, there are several reasoning tasks such as KB satisfiability, subsumption, or greatest lower bound, but they are mutually reducible [4, 7].

## 3. Reasoning with extended datatypes

Reasoning with classical DLs is usually achieved by *tableau* algorithms [2]. Tableau algorithms check the satisfiability of a fuzzy KB  $\mathcal{K}$  by trying to build a completion forest, a mathematical structure from which a model may easily be built. From the algorithm it is immediate either to

$$\begin{aligned}
(\geq t x)^{\mathcal{I}}(z) &= \sup_{c \in \Delta_D} [t^{\mathcal{I}}(z, c) \wedge (c \geq x)] \\
\neg(\geq t x)^{\mathcal{I}}(z) &= \inf_{c \in \Delta_D} [\neg t^{\mathcal{I}}(z, c) \vee (c < x)] \\
(\leq t x)^{\mathcal{I}}(z) &= \sup_{c \in \Delta_D} [t^{\mathcal{I}}(z, c) \wedge (c \leq x)] \\
\neg(\leq t x)^{\mathcal{I}}(z) &= \inf_{c \in \Delta_D} [\neg t^{\mathcal{I}}(z, c) \vee (c > x)] \\
(= t x)^{\mathcal{I}}(z) &= \sup_{c \in \Delta_D} [t^{\mathcal{I}}(z, c) \wedge (c = x)] \\
\neg(= t x)^{\mathcal{I}}(z) &= \inf_{c \in \Delta_D} [\neg t^{\mathcal{I}}(z, c) \vee (c \neq x)] \\
(\geq t FE)^{\mathcal{I}}(z) &= \sup_{c, c_1, \dots, c_n \in \Delta_D} [t^{\mathcal{I}}(z, c) \wedge t_1^{\mathcal{I}}(z, c_1) \wedge \dots \wedge t_n^{\mathcal{I}}(z, c_n) \wedge (c \geq FE(t_1, \dots, t_n))] \\
\neg(\geq t FE)^{\mathcal{I}}(z) &= \inf_{c, c_1, \dots, c_n \in \Delta_D} [\neg t^{\mathcal{I}}(z, c) \vee \neg t_1^{\mathcal{I}}(z, c_1) \vee \dots \vee \neg t_n^{\mathcal{I}}(z, c_n) \vee (c < FE(t_1, \dots, t_n))] \\
(\leq t FE)^{\mathcal{I}}(z) &= \sup_{c, c_1, \dots, c_n \in \Delta_D} [t^{\mathcal{I}}(z, c) \wedge t_1^{\mathcal{I}}(z, c_1) \wedge \dots \wedge t_n^{\mathcal{I}}(z, c_n) \wedge (c \leq FE(t_1, \dots, t_n))] \\
\neg(\leq t FE)^{\mathcal{I}}(z) &= \inf_{c, c_1, \dots, c_n \in \Delta_D} [\neg t^{\mathcal{I}}(z, c) \vee \neg t_1^{\mathcal{I}}(z, c_1) \vee \dots \vee \neg t_n^{\mathcal{I}}(z, c_n) \vee (c > FE(t_1, \dots, t_n))] \\
(= t FE)^{\mathcal{I}}(z) &= \sup_{c, c_1, \dots, c_n \in \Delta_D} [t^{\mathcal{I}}(z, c) \wedge t_1^{\mathcal{I}}(z, c_1) \wedge \dots \wedge t_n^{\mathcal{I}}(z, c_n) \wedge (c = FE(t_1, \dots, t_n))] \\
\neg(= t FE)^{\mathcal{I}}(z) &= \inf_{c, c_1, \dots, c_n \in \Delta_D} [\neg t^{\mathcal{I}}(z, c) \vee \neg t_1^{\mathcal{I}}(z, c_1) \vee \dots \vee \neg t_n^{\mathcal{I}}(z, c_n) \vee (c \neq FE(t_1, \dots, t_n))] \\
(\geq t NE)^{\mathcal{I}}(z) &= \sup_{c, c' \in \Delta_D} [t^{\mathcal{I}}(z, c) \wedge (c \geq c') \wedge NE^{\mathcal{I}}(c')] \\
\neg(\geq t NE)^{\mathcal{I}}(z) &= \inf_{c, c' \in \Delta_D} [\neg t^{\mathcal{I}}(z, c) \vee (c < c') \vee \neg NE^{\mathcal{I}}(c')] \\
(\leq t NE)^{\mathcal{I}}(z) &= \sup_{c, c' \in \Delta_D} [t^{\mathcal{I}}(z, c) \wedge (c \leq c') \wedge NE^{\mathcal{I}}(c')] \\
\neg(\leq t NE)^{\mathcal{I}}(z) &= \inf_{c, c' \in \Delta_D} [\neg t^{\mathcal{I}}(z, c) \vee (c > c') \vee \neg NE^{\mathcal{I}}(c')] \\
(= t NE)^{\mathcal{I}}(z) &= \sup_{c \in \Delta_D} [t^{\mathcal{I}}(z, c) \wedge NE^{\mathcal{I}}(c)] \\
\neg(= t NE)^{\mathcal{I}}(z) &= \inf_{c \in \Delta_D} [\neg t^{\mathcal{I}}(z, c) \vee \neg NE^{\mathcal{I}}(c)]
\end{aligned}$$

Table 2. Semantics of datatype restrictions and their negations.

build a model in case  $\mathcal{K}$  is satisfiable, or to detect that  $\mathcal{K}$  is unsatisfiable.

Our algorithm is similar, but combines a tableau algorithm with an optimization problem, since the application of the tableau rules may generate some constraints which have to be minimized for finding a solution [4].

A completion-forest  $\mathcal{F}$  is a collection of trees whose distinguished roots are arbitrarily connected by edges.

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

Each edge  $\langle v, w \rangle$  is labelled with a set  $\mathcal{L}(\langle v, w \rangle)$  of expressions of the form  $\langle R, l \rangle$ , where  $R$  is a role. The intuition here is that  $\langle v, w \rangle$  is an instance of  $R$  to degree greater or equal than of the evaluation of  $l$ .

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  $x_i, y_i$  are variables, and  $l, l'$  are arithmetic expressions on the variables occurring in the labels.

The algorithm initializes a forest  $\mathcal{F}$  to contain:

- A root node  $v^i$ , for each individual  $a_i$  occurring in  $\mathcal{A}$ , labelled with  $\mathcal{L}(v^i)$  such that  $\mathcal{L}(v^i)$  contains  $\langle C_i, n \rangle$  for each fuzzy assertion  $\langle a_i : C_i, n \rangle \in \mathcal{A}$ .
- An edge  $\langle v^i, v^j \rangle$ , for each fuzzy assertion  $\langle (a_i, a_j) : R_i, n \rangle \in \mathcal{A}$ , labelled with  $\mathcal{L}(\langle v^i, v^j \rangle)$  such that  $\mathcal{L}(\langle v^i, v^j \rangle)$  contains  $\langle R_i, n \rangle$ .

$x_{\tau}$  denotes the variable associated to the assertion  $\tau$  of the form  $a : C$  or  $(a, b) : R$ .  $x_{\tau}$  will take the truth value associated to  $\tau$ . Similarly, given a concrete individual  $c$ ,  $x_c$  will take the value of  $c$ .

$\mathcal{F}$  is then expanded by repeatedly applying some reasoning rules, which may add some constraints to  $\mathcal{C}_{\mathcal{F}}$ . For example, consider the rules for conjunction and disjunction in Gödel fuzzy logic (recall that the t-norm is the minimum and the t-conorm is the maximum) [8]<sup>1</sup>:

$\sqcap_G$ . If  $\langle C \sqcap D, \alpha \rangle \in \mathcal{L}(v)$  then append  $\langle C, \alpha \rangle$ , and  $\langle D, \alpha \rangle$  to  $\mathcal{L}(v)$ .

$\sqcup_G$ . If  $\langle C \sqcup D, \alpha \rangle \in \mathcal{L}(v)$  then (i) append  $\langle C, x_1 \rangle$  and  $\langle D, x_2 \rangle$  to  $\mathcal{L}(v)$ , and (ii)  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_1 + x_2 = \alpha\} \cup \{x_1 \leq y\} \cup \{x_2 \leq 1 - y\} \cup \{x_1 \in [0, 1]\} \cup \{x_2 \in [0, 1]\} \cup \{y \in \{0, 1\}\}$ .

The completion-forest is complete when none of the completion rules are applicable. Then, the optimization problem on the set of constraints  $\mathcal{C}_{\mathcal{F}}$  is solved.

In general, in fuzzy DLs we may end up with a bMINLP (bounded Mixed Integer Non Linear Programming) problem [4]. However, in Zadeh, Łukasiewicz and Gödel fuzzy logics we end up with a bMILP (bounded Mixed Integer Linear Programming) problem [6], whereas in Product fuzzy logic we end up with a MIQCP (Mixed Integer Quadratically Constrained Programming) problem.

<sup>1</sup>For the sake of clarity, we include here the most comprehensive formulation of the rules. We note that they can be optimized as shown in [4], although it is out of the scope of this paper.

A reasoning algorithm for  $\mathcal{ALCF}(\mathbf{D})$  has already been presented in [4], restricting to datatype restrictions involving real numbers. In this section, we will show the additional rules for managing our extended datatype restrictions.

### 3.1. Datatype restrictions with variables

We start by presenting the rules to reason with datatype restrictions with a variable as argument, and then we will provide an example. There are six rules, corresponding to  $(\geq t x)$ ,  $\neg(\geq t x)$ ,  $(\leq t x)$ ,  $\neg(\leq t x)$ ,  $(= t x)$ , and  $\neg(= t x)$ , respectively.

$(\geq t x)$ . If  $\langle (\geq t x), \alpha \rangle \in \mathcal{L}(v)$  then: (i) if  $\langle t(v, c), \beta \rangle \notin \mathcal{L}(v, c)$  then add  $\langle t(v, c), 1 - y \rangle$  to  $\mathcal{L}(v, c)$  and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_c \geq k_1, x_c \leq k_2\}$ , (ii) for some  $c$  such that  $\langle t(v, c), \beta \rangle \in \mathcal{L}(v, c)$ ,  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{\alpha + y \geq \epsilon\} \cup \{\alpha + y \leq 1\} \cup \{x_{t(v, c)} \geq 1 - y\} \cup \{x_c \geq x - (k_2 - k_1)y\} \cup \{y \in \{0, 1\}\}$ , where  $y$  is a new variable.

$\neg(\geq t x)$ . If (i)  $\langle \neg(\geq t x), \alpha_1 \rangle \in \mathcal{L}(v)$  and (ii)  $\langle t(v, c), \alpha_2 \rangle \in \mathcal{L}(v, c)$ , then  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{y_1 \leq y_2\} \cup \{x_{t(v, c)} = 1 - y_1\} \cup \{\alpha_1 + y_1 + y_2 \geq \epsilon\} \cup \{\alpha_1 - y_1 + y_2 \leq 1\} \cup \{\alpha_2 \leq 1 - y_1\} \cup \{x_c \leq (x - \epsilon) - (k_1 - k_2 - \epsilon)y_2\} \cup \{y_i \in \{0, 1\}\}$ , where  $y_i$  are new variables.

$(\leq t x)$ . If  $\langle (\leq t x), \alpha \rangle \in \mathcal{L}(v)$  then: (i) if  $\langle t(v, c), \beta \rangle \notin \mathcal{L}(v, c)$  then add  $\langle t(v, c), 1 - y \rangle$  to  $\mathcal{L}(v, c)$  and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_c \geq k_1, x_c \leq k_2\}$ , (ii) for some  $c$  such that  $\langle t(v, c), \beta \rangle \in \mathcal{L}(v, c)$ ,  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{\alpha + y \geq \epsilon\} \cup \{\alpha + y \leq 1\} \cup \{x_{t(v, c)} \geq 1 - y\} \cup \{x_c \leq x - (k_2 - k_1)y\} \cup \{y \in \{0, 1\}\}$ , where  $y$  is a new variable.

$\neg(\leq t x)$ . If (i)  $\langle \neg(\leq t x), \alpha_1 \rangle \in \mathcal{L}(v)$  and (ii)  $\langle t(v, c), \alpha_2 \rangle \in \mathcal{L}(v, c)$ , then  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{y_1 \leq y_2\} \cup \{x_{t(v, c)} = 1 - y_1\} \cup \{\alpha_1 + y_1 + y_2 \geq \epsilon\} \cup \{\alpha_1 - y_1 + y_2 \leq 1\} \cup \{\alpha_2 \leq 1 - y_1\} \cup \{x_c \geq (x + \epsilon) - (k_2 + \epsilon - k_1)y_2\} \cup \{y_i \in \{0, 1\}\}$ , where  $y_i$  are new variables.

$(= t x)$ . If  $\langle (= t x), \alpha \rangle \in \mathcal{L}(v)$  then: (i) if  $\langle t(v, c), \beta \rangle \notin \mathcal{L}(v, c)$  then add  $\langle t(v, c), 1 - y \rangle$  to  $\mathcal{L}(v, c)$  and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_c \geq k_1, x_c \leq k_2\}$ , (ii) for some  $c$  such that  $\langle t(v, c), \beta \rangle \in \mathcal{L}(v, c)$ ,  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{\alpha + y \geq \epsilon\} \cup \{\alpha + y \leq 1\} \cup \{x_{t(v, c)} \geq 1 - y\} \cup \{x_c \geq x - (k_2 - k_1)y\} \cup \{x_c \leq x - (k_2 - k_1)y\} \cup \{y \in \{0, 1\}\}$ , where  $y$  is a new variable.

$\neg(= t x)$ . If (i)  $\langle \neg(= t x), \alpha_1 \rangle \in \mathcal{L}(v)$  and (ii)  $\langle t(v, c), \alpha_2 \rangle \in \mathcal{L}(v, c)$ , then:  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{y_2 \geq y_1\} \cup \{y_3 \geq y_2\} \cup \{x_{t(v, c)} = 1 - y_1\} \cup \{\alpha_1 + y_1 + y_2 \geq \epsilon\} \cup \{\alpha_1 - y_1 + y_2 \leq 1\} \cup \{\alpha_2 \leq 1 - y_1\} \cup \{x_c \leq (x - \epsilon) - (k_1 - \epsilon - k_2)y_2 - (k_1 - \epsilon - k_2)(1 - y_3)\} \cup \{x_c \geq (x + \epsilon) - (k_2 + \epsilon - k_1)y_3\} \cup \{y_i \in \{0, 1\}\}$ , where  $y_i$  are new variables.

For the sake of concrete illustration, let us explain a couple of rules. Initially, let us consider the  $(\geq t x)$  rule. If  $\langle (\geq t x), \alpha \rangle \in \mathcal{L}(v)$ , there must exist a  $t$ -filler  $c$  satisfying  $t^{\mathcal{I}}(x, c) \wedge (c \geq x) \geq \alpha$ . There are two possibilities for satisfying the semantics:

- $\alpha = 0$ , because  $\tau^{\mathcal{I}} \geq 0$  is always true.

- $\alpha > 0$ , but  $x_{t(v, c)} = 1$  and  $x_c > x$ , since this implies  $t^{\mathcal{I}}(z, c) \wedge (c \geq n) = 1$  and hence  $\langle \geq t x, \alpha \rangle$  is satisfied.

So, the first step is to create a  $t$ -filler  $c$  if it does not exist, adding  $\langle t(v, c), 1 - y \rangle$  to  $\mathcal{L}(v, c)$ . Then, we use a binary variable  $y$  simulating the two branches:

- If  $y = 1$ , then  $\alpha = 0$  and the other variables are not constrained ( $x_{t(v, c)} \geq 0, x_c \in [k_1, k_2]$ ).
- If  $y = 0$ , then  $\alpha \geq \epsilon \Leftrightarrow \alpha > 0, x_{t(v, c)} = 1$  and  $x_c > x$ .

Notice that strict inequalities are not allowed, and hence the inequality  $x > 0$  is expressed as  $x \geq \epsilon$ , for a sufficiently small  $\epsilon > 0$ .

Let us consider now the  $\neg(\geq t x)$  rule. If  $\langle \neg(\geq t x), \alpha_1 \rangle \in \mathcal{L}(v)$  and, for some  $c$ ,  $\langle t(v, c), \alpha_2 \rangle \in \mathcal{L}(v, c)$ , there are three possibilities to satisfy these formulae:

- $\alpha_1 = 0$ , because  $\tau^{\mathcal{I}} \geq 0$  is always true.
- $x_{t(v, c)} = 0$ , since it implies  $[\neg t^{\mathcal{I}}(z, c) \vee (c < n)] = 1$  and hence  $\langle \neg(\geq t x), \alpha_1 \rangle$  is satisfied. Now, in order to satisfy  $t(v, c) \geq \alpha_2$ ,  $\alpha_2$  must be 0.
- $x_{t(v, c)} = 1$  and  $x_c < n$ , because this implies  $(c < n) \Leftrightarrow [\neg t^{\mathcal{I}}(x, c) \vee (c < n)] = 1$  and hence  $\langle \neg(\geq t x), \alpha_1 \rangle$  is satisfied.

In order to cover all these possibilities, the binary variables  $y_1$  and  $y_2$  simulate three branches:

- $y_1 = 0, y_2 = 1$ . In this case,  $\alpha_1 = 0$  and hence the other variables are not constrained ( $x_{t(v, c)} = 1, \alpha_2 \leq 1, x_c \in [k_1, k_2]$ ).
- $y_1 = 1, y_2 = 1$ . Now,  $x_{t(v, c)} = 0$  and  $\alpha_2 = 0$ .  $x_c$  is not constrained ( $x_c \in [k_1, k_2]$ ).
- $y_1 = 0, y_2 = 0$ . Then,  $x_{t(v, c)} = 1$  and  $x_c < x$ .
- Since  $y_1 \leq y_2$ , the case  $y_1 = 1$  and  $y_2 = 0$  is not possible.

$(\geq t x)$  and  $(\leq t x)$  rules are similar, and  $(= t x)$  uses a combination of them. This is also true for the other rules. The main difficulty is that, while in  $\neg(\geq t x)$  we add  $x_c < x$  to simulate that  $c \geq x$  is not true, in  $\neg(= t x)$  a binary variable simulates the two possibilities satisfying that  $c = x$  is not true:  $c \geq x$  or  $c \leq x$ .

Let us also remark that these rules are an extension of the rules for real numbers in [4]. These rules also work properly if we replace the variable  $x$  with a real number.

### 3.2. Datatype restrictions with arithmetic of features

We will present the rules to reason with datatype restrictions with a arithmetic expression on features (of the form  $FE(t_1, \dots, t_n)$ ) and then show an example. There are six rules, corresponding to  $(\geq t FE)$ ,  $(\neg(\geq t FE))$ ,  $(\leq t FE)$ ,  $(\neg(\leq t FE))$ ,  $(= t FE)$ , and  $(\neg(= t FE))$ , respectively.

$(\geq t FE)$ . If  $\langle(\geq t FE(t_1, \dots, t_n)), \alpha\rangle \in \mathcal{L}(v)$  then: (i) if  $\langle t(v, c), \beta\rangle \notin \mathcal{L}(v, c)$  then add  $\langle t(v, c), 1 - y\rangle$  to  $\mathcal{L}(v, c)$  and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_c \geq k_1, x_c \leq k_2\}$ , (ii) if  $\langle t_i(v, c_i), \beta_i\rangle \notin \mathcal{L}(v, c_i)$  then add  $\langle t_i(v, c_i), 1 - y\rangle$  to  $\mathcal{L}(v, c_i)$  and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{c_i} \geq k_1, x_{c_i} \leq k_2\}$ , (iii) for some  $c$  such that  $\langle t(v, c), \beta\rangle \in \mathcal{L}(v, c)$ , and some  $c_i$  such that  $\langle t_i(v, c_i), \beta_i\rangle \in \mathcal{L}(v, c_i)$ ,  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{\alpha + y \geq \epsilon\} \cup \{\alpha + y \leq 1\} \cup \{x_{t(v, c)} \geq 1 - y\} \cup \{x_{t(v, c_i)} \geq 1 - y\} \cup \{x_c \geq FE(c_1, \dots, c_n) - (k_2 - k_1)y\} \cup \{y \in \{0, 1\}\}$ , where  $y$  is a new variable.

$(\neg \geq t FE)$ . If (i)  $\langle \neg(\geq t FE(t_1, \dots, t_n)), \alpha_1\rangle \in \mathcal{L}(v)$  and (ii)  $\langle t(v, c), \alpha_2\rangle \in \mathcal{L}(v, c)$  and (iii)  $\langle t_i(v, c_i), \beta_i\rangle \in \mathcal{L}(v, c_i)$ , then  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{y_1 \leq y_2\} \cup \{x_{t(v, c)} = 1 - y_1\} \cup \{x_{t(v, c_i)} = 1 - y_1\} \cup \{\alpha_1 + y_1 + y_2 \geq \epsilon\} \cup \{\alpha_1 - y_1 + y_2 \leq 1\} \cup \{\alpha_2 \leq 1 - y_1\} \cup \{\beta_i \leq 1 - y_1\} \cup \{x_c \leq FE(c_1, \dots, c_n) - \epsilon - (k_2 - k_1 - \epsilon)y_2\} \cup \{y_i \in \{0, 1\}\}$ , where  $y_i$  are new variables.

$(\leq t FE)$ . If  $\langle(\leq t FE(t_1, \dots, t_n)), \alpha\rangle \in \mathcal{L}(v)$  then: (i) if  $\langle t(v, c), \beta\rangle \notin \mathcal{L}(v, c)$  then add  $\langle t(v, c), 1 - y\rangle$  to  $\mathcal{L}(v, c)$  and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_c \geq k_1, x_c \leq k_2\}$ , (ii) if  $\langle t_i(v, c_i), \beta_i\rangle \notin \mathcal{L}(v, c_i)$  then add  $\langle t_i(v, c_i), 1 - y\rangle$  to  $\mathcal{L}(v, c_i)$  and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{c_i} \geq k_1, x_{c_i} \leq k_2\}$ , (iii) for some  $c$  such that  $\langle t(v, c), \beta\rangle \in \mathcal{L}(v, c)$ , and some  $c_i$  such that  $\langle t_i(v, c_i), \beta_i\rangle \in \mathcal{L}(v, c_i)$ ,  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{\alpha + y \geq \epsilon\} \cup \{\alpha + y \leq 1\} \cup \{x_{t(v, c)} \geq 1 - y\} \cup \{x_{t(v, c_i)} \geq 1 - y\} \cup \{x_c \leq FE(c_1, \dots, c_n) - (k_2 - k_1)y\} \cup \{y \in \{0, 1\}\}$ , where  $y$  is a new variable.

$(\neg \leq t FE)$ . If (i)  $\langle \neg(\leq t FE(t_1, \dots, t_n)), \alpha_1\rangle \in \mathcal{L}(v)$  and (ii)  $\langle t(v, c), \alpha_2\rangle \in \mathcal{L}(v, c)$  and (iii)  $\langle t_i(v, c_i), \beta_i\rangle \in \mathcal{L}(v, c_i)$ , then  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{y_1 \leq y_2\} \cup \{x_{t(v, c)} = 1 - y_1\} \cup \{x_{t(v, c_i)} = 1 - y_1\} \cup \{\alpha_1 + y_1 + y_2 \geq \epsilon\} \cup \{\alpha_1 - y_1 + y_2 \leq 1\} \cup \{\alpha_2 \leq 1 - y_1\} \cup \{\beta_i \leq 1 - y_1\} \cup \{x_c \geq FE(c_1, \dots, c_n) + \epsilon - (k_2 - k_1 + \epsilon)y_2\} \cup \{y_i \in \{0, 1\}\}$ , where  $y_i$  are new variables.

$(= t FE)$ . If  $\langle(= t FE(t_1, \dots, t_n)), \alpha\rangle \in \mathcal{L}(v)$  then: (i) if  $\langle t(v, c), \beta\rangle \notin \mathcal{L}(v, c)$  then add  $\langle t(v, c), 1 - y\rangle$  to  $\mathcal{L}(v, c)$  and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_c \geq k_1, x_c \leq k_2\}$ , (ii) if  $\langle t_i(v, c_i), \beta_i\rangle \notin \mathcal{L}(v, c_i)$  then add  $\langle t_i(v, c_i), 1 - y\rangle$  to  $\mathcal{L}(v, c_i)$  and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{c_i} \geq k_1, x_{c_i} \leq k_2\}$ , (iii) for some  $c$  such that  $\langle t(v, c), \beta\rangle \in \mathcal{L}(v, c)$ , and some  $c_i$  such that  $\langle t_i(v, c_i), \beta_i\rangle \in \mathcal{L}(v, c_i)$ ,  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{\alpha + y \geq \epsilon\} \cup \{\alpha + y \leq 1\} \cup \{x_{t(v, c)} \geq 1 - y\} \cup \{x_{t(v, c_i)} \geq 1 - y\} \cup \{x_c \geq FE(c_1, \dots, c_n) - (k_2 - k_1)y\} \cup \{x_c \leq FE(c_1, \dots, c_n) - (k_2 - k_1)y\} \cup \{y \in \{0, 1\}\}$ , where  $y$  is a new variable.

$(\neg = t FE)$ . If (i)  $\langle \neg(= t FE(t_1, \dots, t_n)), \alpha_1\rangle \in \mathcal{L}(v)$  and (ii)  $\langle t(v, c), \alpha_2\rangle \in \mathcal{L}(v, c)$  and (iii)  $\langle t_i(v, c_i), \beta_i\rangle \in \mathcal{L}(v, c_i)$ ,

then  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{y_2 \geq y_1\} \cup \{y_3 \geq y_2\} \cup \{x_{t(v, c)} = 1 - y_1\} \cup \{x_{t(v, c_i)} = 1 - y_1\} \cup \{\alpha_1 + y_1 + y_2 \geq \epsilon\} \cup \{\alpha_1 - y_1 + y_2 \leq 1\} \cup \{\alpha_2 \leq 1 - y_1\} \cup \{\beta_i \leq 1 - y_1\} \cup \{x_c \leq FE(c_1, \dots, c_n) - \epsilon - (k_1 - \epsilon - k_2)y_2 - (k_1 - \epsilon - k_2)(1 - y_3)\} \cup \{x_c \geq FE(c_1, \dots, c_n) + \epsilon - (k_2 + \epsilon - k_1)y_3\} \cup \{y_i \in \{0, 1\}\}$ , where  $y_i$  are new variables.

The rules are similar to those on Section 3.1. For instance, let us compare rules  $(\geq t x)$  and  $(\geq t FE)$ . The differences are the following:

- $(\geq t x)$  creates a  $t$ -filler, whereas  $(\geq t FE)$  creates  $n + 1$  fillers  $c, c_1, c_2, \dots, c_n$  for the features  $t, t_1, t_2, \dots, t_n$ .
- $(\geq t x)$  adds a constraint on the value of the relation via  $t(x_{t(v, c)} \geq 1 - y)$ , whereas  $(\geq t FE)$  adds such a constraint for each of the  $n + 1$  features involved.
- $(\geq t x)$  adds a constraint stating that if  $y = 0$  then  $x_c \geq x$ , whereas in  $(\geq t FE)$  the constraint is obviously extended: if  $y = 0$  then  $x_c \geq FE(x_{c_1}, x_{c_2}, \dots, x_{c_n})$ . For example, given the concept  $(\geq \text{hasGrossSalary } 0.8 \cdot \text{hasNetSalary})$ , the constraint imposes  $x_c = 0.8 \cdot x_{c_1}$  when  $y = 0$ .

### 3.3. Datatype restrictions with arithmetic of fuzzy numbers

It only remains to present the rules dealing with datatype restrictions with a fuzzy number expression  $NE$  as argument. There are six rules, corresponding to  $(\geq t NE)$ ,  $(\neg(\geq t NE))$ ,  $(\leq t NE)$ ,  $(\neg(\leq t NE))$ ,  $(= t x)$ , and  $(\neg(= t NE))$ , respectively.

$(\geq t NE)$ . If  $\langle(\geq t NE), \alpha\rangle \in \mathcal{L}(v)$  then: (i) if  $\langle t(v, c), \beta\rangle \notin \mathcal{L}(v, c)$  then add  $\langle t(v, c), 1 - y\rangle$  to  $\mathcal{L}(v, c)$  and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_c \geq k_1, x_c \leq k_2\}$ , (ii) for some  $c$  such that  $\langle t(v, c), \beta\rangle \in \mathcal{L}(v, c)$ , then: (a) if there does not exist a representative  $c'$  of the values  $\langle x, (\geq t NE)\rangle$ , then mark  $c'$  as a representative of the values  $\langle x, (\geq t NE)\rangle$  and add  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{c'} \geq k_1, x_{c'} \leq k_2\}$ , (b) append  $\langle NE, \alpha\rangle$  to  $\mathcal{L}(c')$ , (c)  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{\alpha \geq y \cdot \epsilon\} \cup \{\alpha \leq y\} \cup \{x_{t(v, c)} = y\} \cup \{x_c - x_{c'} \geq (1 - y) \cdot (k_1 - k_2)\} \cup \{y \in \{0, 1\}\}$ .

$(\neg \geq t NE)$ . If (i)  $\langle \neg(\geq t NE), \alpha_1\rangle \in \mathcal{L}(v)$ , (ii)  $\langle t(v, c), \alpha_2\rangle \in \mathcal{L}(v, c)$ , and (iii) there exists a representative  $c'$  of the values  $\langle x, (\leq t NE)\rangle$ , then (i) add  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{c'} \geq k_1, x_{c'} \leq k_2\}$ , (ii) append  $\langle NE, \alpha_1 - y_2\rangle$  to  $\mathcal{L}(c')$ , and (iii)  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{y_1 + y_2 + y_3 + y_4 = 3\} \cup \{\alpha_1 \leq y_3\} \cup \{\alpha_2 \leq y_4\} \cup \{x_{t(v, c)} = y_4\} \cup \{x_c \leq (x_{c'} - \epsilon) - (k_1 - k_2 - \epsilon)y_1\} \cup \{y_i \in \{0, 1\}\}$ .

$(\leq t NE)$ . If  $\langle(\leq t NE), \alpha\rangle \in \mathcal{L}(v)$  then: (i) if  $\langle t(v, c), \beta\rangle \notin \mathcal{L}(v, c)$  then add  $\langle t(v, c), 1 - y\rangle$  to  $\mathcal{L}(v, c)$  and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_c \geq k_1, x_c \leq k_2\}$ , (ii) for some  $c$  such that  $\langle t(v, c), \beta\rangle \in \mathcal{L}(v, c)$ , then: (a) if there does not exist a representative  $c'$  of the values  $\langle x, (\leq t NE)\rangle$ , then mark  $c'$  as a representative of the values  $\langle x, (\leq t NE)\rangle$  and add

$\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{c'} \geq k_1, x_{c'} \leq k_2\}$ , (b) append  $\langle NE, \alpha \rangle$  to  $\mathcal{L}(c')$ , (c)  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{\alpha \geq y \cdot \epsilon\} \cup \{\alpha \leq y\} \cup \{x_{t(v,c)} = y\} \cup \{x_{c'} - x_c \geq (1 - y) \cdot (k_1 - k_2)\} \cup \{y \in \{0, 1\}\}$ .  
 $(\neg \leq t NE)$ . If (i)  $\langle \neg(\leq t NE), \alpha_1 \rangle \in \mathcal{L}(v)$ , (ii)  $\langle t(v, c), \alpha_2 \rangle \in \mathcal{L}(v, c)$ , and (iii) there exists a representative  $c'$  of the values  $\langle x, (\leq t NE) \rangle$ , then (i) add  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{c'} \geq k_1, x_{c'} \leq k_2\}$ , (ii) append  $\langle NE, \alpha_1 - y_2 \rangle$  to  $\mathcal{L}(c')$ , and (iii)  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{y_1 + y_2 + y_3 + y_4 = 3\} \cup \{\alpha_1 \leq y_3\} \cup \{\alpha_2 \leq y_4\} \cup \{x_{t(v,c)} = y_4\} \cup \{x_{c'} \leq (x_c - \epsilon) - (k_1 - k_2 - \epsilon)y_1\} \cup \{y_i \in \{0, 1\}\}$ .  
 $(= t NE)$ . If  $\langle (\geq t NE), \alpha \rangle \in \mathcal{L}(v)$  then: (i) if  $\langle t(v, c), \beta \rangle \notin \mathcal{L}(v, c)$  then add  $\langle t(v, c), 1 - y \rangle$  to  $\mathcal{L}(v, c)$  and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_c \geq k_1, x_c \leq k_2\}$ , (ii) for some  $c$  such that  $\langle t(v, c), \beta \rangle \in \mathcal{L}(v, c)$ , then append  $\langle NE, \alpha \rangle$  to  $\mathcal{L}(c)$ , and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{x_{t(v,c)} \geq \alpha\}$ .  
 $(\neg = t NE)$ . If (i)  $\langle \neg(= t NE), \alpha_1 \rangle \in \mathcal{L}(v)$  and (ii)  $\langle t(v, c), \alpha_2 \rangle \in \mathcal{L}(v, c)$ , then append  $\langle NE, \alpha_1 - y_2 \rangle$  to  $\mathcal{L}(c)$  and  $\mathcal{C}_{\mathcal{F}} = \mathcal{C}_{\mathcal{F}} \cup \{y_2 + y_3 + y_4 = 2\} \cup \{\alpha_1 \leq y_3\} \cup \{\alpha_2 \leq y_4\} \cup \{x_{t(v,c)} = y_4\} \cup \{y_i \in \{0, 1\}\}$ .

where  $y, y_i$  are new variables. These rules are similar to those on Section 3.1. For example, consider  $(\geq t NE)$  rule. There are two possibilities to satisfy the semantics:

- $\alpha = 0$ , because  $\tau^{\mathcal{I}} \geq 0$  is always true.
- $x_{t(v,c)} = 1$ ,  $NE(c')^{\mathcal{I}} \geq \alpha$ , and  $x_c > x$ , since this implies  $t^{\mathcal{I}}(z, c) \wedge (c \geq c') \wedge NE^{\mathcal{I}}(c') \geq \alpha$ .

The first step is to create a  $t$ -filler  $c$ , but now we also create (if necessary) an individual  $c'$  representative of the values greater or equal than  $NE$ . We also add  $NE(c')^{\mathcal{I}} \geq \alpha^2$ . Finally, a binary variable  $y$  simulates two branches:

- If  $y = 0$ , then  $\alpha = 0$ ,  $x_{t(v,c)} \geq 0$ , and  $x_c$  is not constrained ( $x_c \in [k_1, k_2]$ ).
- If  $y = 1$ , then  $\alpha > 0$ ,  $x_{t(v,c)} = 1$ , and  $x_c > x_{c'}$ .

**Example.** We have to choose among different products  $a_i$  that have been evaluated according to some criteria  $t_j$ . For instance,  $t_1 = \text{Cost}$ ,  $t_2 = \text{DeliveryTime}$ ,  $t_3 = \text{Quality}$ . Each criteria is represented as a concrete feature and has a relative importance  $w_j \in [0, 1]$  assigned by an expert ( $\sum_j w_j = 1$ ). E.g.,  $w_1 = 0.258$ ,  $w_2 = 0.105$ ,  $w_3 = 0.637$ .

Let  $a_{ij}$  denote the value (a real or a fuzzy number) of the attribute  $t_j$  for the alternative  $a_i$ . For instance,  $a_{11} = \text{VeryPoor}$ ,  $a_{12} = \text{Fair}$ ,  $a_{13} = \text{Good}$ ,  $a_{21} = \text{Good}$ ,  $a_{22} = \text{VeryGood}$ ,  $a_{23} = \text{Poor}$ , and  $a_{31} = \text{Fair}$ ,  $a_{32} = \text{Fair}$ ,  $a_{33} = \text{Poor}$ . The fuzzy numbers are defined as  $\text{VeryPoor} = \text{tri}(0, 0, 2)$ ,  $\text{Poor} = \text{tri}(1, 2.5, 4)$ ,  $\text{Fair} = \text{tri}(3, 5, 7)$ ,  $\text{Good} = \text{tri}(6, 7.5, 9)$ , and  $\text{VeryGood} = \text{tri}(8, 10, 10)$ .

Let  $\text{Alternative}$  be a shorthand for the concept  $(= t_{sum}(w_1 \cdot t_1 + w_2 \cdot t_2 + w_3 \cdot t_3))$ . For every alternative  $a_i$  with  $1 \leq i \leq 3$ , a decision matrix can be encoded as:

$$a_i : \text{Alternative} \sqcap (= t_1 a_{i1}) \sqcap (= t_2 a_{i2}) \sqcap (= t_3 a_{i3})$$

The rank value of  $a_i$  w.r.t. a fuzzy KB  $\mathcal{K}$  is computed using the middle of maxima (MOM) defuzzification [3]:

$$rv(\mathcal{K}, a_i) = \text{mom}(\mathcal{K}, \text{Alternative}, a_i, t_{sum})$$

Finally, the optimal alternative is:

$$\arg \max_{a_i} rv(\mathcal{K}, a_i).$$

In particular,  $rv(\mathcal{K}, a_1) = 5.301$ ,  $rv(\mathcal{K}, a_2) = 4.577$ ,  $rv(\mathcal{K}, a_3) = 3.408$ , so the ranking of the alternatives is  $a_1 \succ a_2 \succ a_3$ , and the optimal alternative is  $a_1$ .

## 4. Conclusions

In this paper we have presented a fuzzy DL which extends the expressivity of the supported datatype restrictions with respect to the related work. We have also provided the rules for extending current reasoning algorithms in order to deal with the new constructors of the logic. The rules have been implemented in the FUZZYDL reasoner [3], publicly available from <http://www.straccia.info>.

Our approach is independent of the family of fuzzy operators used, and thus the results apply to Zadeh fuzzy logic [7], and to the logic of any continuous t-norm (such as Łukasiewicz, Gödel, or Product) [4].

Due to space limitations we have restricted ourselves to features ranging over reals, which obviously includes the case of integers. Furthermore, it is possible to reduce strings to the case of reals using the algorithm in [4].

As future work we plan to use the additional expressivity to model some real-world application domains.

## References

- [1] F. Baader, I. Horrocks, and U. Sattler. Description Logics. In F. van Harmelen, V. Lifschitz, and B. Porter, editors, *Handbook of Knowledge Representation*. Elsevier, 2007.
- [2] F. Baader and U. Sattler. An overview of tableau algorithms for Description Logics. *Studia Logica*, 69:5–40, 2001.
- [3] F. Bobillo and U. Straccia. fuzzyDL: An expressive fuzzy Description Logic reasoner. In *Proceedings of the 17th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2008)*, pages 923–930. IEEE Computer Society, 2008.
- [4] F. Bobillo and U. Straccia. Fuzzy Description Logics with general t-norms and datatypes. *Fuzzy Sets and Systems*, 2009.
- [5] T. Lukasiewicz and U. Straccia. Managing uncertainty and vagueness in Description Logics for the semantic web. *Journal of Web Semantics*, 6(4):291–308, 2008.
- [6] H. M. Salkin and K. Mathur. *Foundations of Integer Programming*. North-Holland, 1989.
- [7] U. Straccia. Reasoning within fuzzy Description Logics. *Journal of Artificial Intelligence Research*, 14:137–166, 2001.
- [8] U. Straccia. Description Logics with fuzzy concrete domains. In *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI 2005)*. AUAI Press, 2005.

<sup>2</sup>The rule to reason with triangular functions has been presented in [8].