Towards Spatial Reasoning in Fuzzy Description Logics

Umberto Straccia

Abstract—Fuzzy Description Logics are 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, fuzzy DLs are open to be extended with several features worked out in the fuzzy logic literature. In this work, we extend fuzzy DLs towards supporting fuzzy spatial reasoning and, thus, offer a framework for modeling spatial relations such as “region a is part of region b, which is connected to region c, a is close to c and b is right over c”. 

I. INTRODUCTION

Description Logics (DLs) [1] play a key role in the design of Ontologies and in the Semantic Web [8]. 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. However, 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 [20], [22]. Fuzzy ontologies emerge as useful in several applications, such as (multimedia) information retrieval, image interpretation, ontology matching, matchmaking and the Semantic Web [10]. So far, several fuzzy extensions of DLs can be found in the literature (see the survey in [10]) and some fuzzy DL reasoners have been implemented, such as FUZZYDL [4], DELOREAN [2] or FIRE [19].

In this work, we make a first step in extending fuzzy DLs towards fuzzy spatial reasoning, by supporting both fuzzy spatial relations of the region connection calculus (RCC) [5], [14], [18] as well as application domain dependent fuzzy spatial relations such as “close”, “far”, “over” (see, e.g. [9], [17]). The need for such an extension is particularly evident in the image interpretation domain, as shown, for instance in [9]. We refer the reader to [18] for an extensive overview on fuzzy spatial reasoning.

In the following, Section II (resp. Section III) will provide the basic concepts and algorithms related to mathematical fuzzy logic (resp. fuzzy spatial logic) we will rely on, Section IV specifies our spatial fuzzy DLs, while Section V describes the reasoning algorithms. Eventually, Section VI concludes and provides some issues that may be addressed in future work.

II. MATHEMATICAL FUZZY LOGIC

In Mathematical Fuzzy Logic [7], the convention prescribing that a statement is either true or false is changed and is a matter of degree measured on an ordered scale S that is no longer {0, 1}, but the unit interval [0, 1]. This degree is called degree of truth of the statement φ in the interpretation I. In this section, fuzzy statements have the form φ ⩾ l or φ ⩽ u, where l, u ∈ [0, 1] (see, e.g. [6], [7]) 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 ⩾ 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 I maps each basic statement p_i into [0, 1] and is then extended inductively to all statements: I(φ ∧ ψ) = I(φ) ⊗ I(ψ), I(φ ∨ ψ) = I(φ) ⊕ I(ψ), I(¬ φ) = ⊖ I(φ), I(∃ x.R(x)) = sup a∈A I(φ(a)), I(∀ x.R(x)) = inf a∈A I(φ(a)), where A is the domain of I, and ⊗, ⊕, ⊖, and ∆ are t-norms, t-conorms, implication functions, and negation functions, respectively, which extend the classical Boolean conjunction, disjunction, and negation, respectively, to the fuzzy case. One usually distinguishes three different logics, namely Łukasiewicz, Gödel, and Product logic [7]. Zadeh “logic”, namely a ⊗ b = min(a, b), a ⊕ b = max(a, b), ⊖ a = 1 − a and a → b = max(1 − a, b) is entailed by Łukasiewicz logic.

<table>
<thead>
<tr>
<th>Łukasiewicz Logic</th>
<th>Gödel Logic</th>
<th>Product Logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>a ⊗ b</td>
<td>max(a + b − 1, 0)</td>
<td>min(a, b)</td>
</tr>
<tr>
<td>a ⊕ b</td>
<td>min(a + b, 1)</td>
<td>max(a, b)</td>
</tr>
<tr>
<td>a ⇒ b</td>
<td>min(1 − a + b, 1)</td>
<td>1 if a ≤ b</td>
</tr>
<tr>
<td>⊖ a</td>
<td>1 − a</td>
<td>1 if a = 0</td>
</tr>
</tbody>
</table>

A fuzzy set R over a countable crisp set X is a function R : X → [0, 1]. A (binary) fuzzy relation R over two countable crisp sets X and Y is a function R : X × Y → [0, 1]. R is reflexive iff for all x ∈ X, R(x, x) = 1 holds, while R is symmetric iff for all x, y ∈ X, R(x, y) = R(y, x) holds. We say that R is functional iff R is a partial function: R : X × Y → {0, 1} such that for each x ∈ X there is unique y ∈ X where R(x, y) is defined. The degree of subsumption between two fuzzy sets A and B is defined as inf_{x ∈ X} A(x) ⇒ B(x) and may be seen as the degree of the FOL formula ∀ x.A(x) ⇒ B(x), while the degree of overlap between two fuzzy sets A and B is defined as sup_{x ∈ X} A(x) ⊗ B(x) and may be seen as the degree of the FOL formula ∃ x.A(x) ∧ B(x).

The notions of satisfaction and logical consequence are defined in the standard way. A fuzzy interpretation I satisfies a fuzzy statement φ ⩾ l (resp., φ ⩽ u) or I is a model of φ ⩾ l (resp., φ ⩽ u), denoted I |= φ ⩾ l (resp., I |= φ ⩽ u), iff I(φ) ⩾ l (resp., I(φ) ⩽ u). Furthermore, φ ⩾ l is a right logical consequence of a set of fuzzy statements K iff l is the infimum of I(φ) subject to all models I of K. The latter value is equivalent to l = sup { r | K |= r }, it is called Best Entailment Degree (BED), and is denoted bed(K, φ),
while the Best Satisfiability Degree, denoted as $bsd(K,\phi)$, is defined as $\sup_{I\vdash K} I(\phi)$. We refer the reader to [7], [6] for reasoning algorithms for fuzzy propositional and First-Order Logics (FOLs).

For sake of this paper, we provide a simple and effective way to solve the entailment problem for propositional fuzzy logic, in the case of Zadeh logic and Lukasiewicz logic. We will use the concepts and algorithms we describe next in the following sections. Our solution is based on Mixed Integer Linear Programming (MILP) – see also [6]. The calculus depends on the t-norm, t-conorm and negation functions considered. Suppose we are looking for $bsd(K,\phi)$, then,

$$bsd(K,\phi) = \min x . \text{such that } K \cup \{ \phi \leq x \} \text{ satisfiable}.$$  

The above problem can be solved by means of MILP. The solution proposed here is an optimized version w.r.t. [6]. For a formula $\phi$ consider a variable $x_\phi$. The intuition is that the degree of truth of $\phi$ is greater or equal to $x_\phi$. The MILP problem determining $bsd(K,\phi)$ is as follows:

$$\min x . \text{such that } x \in [0,1], x_\phi \geq 1 - x, \sigma(\neg \phi),$$

for all $(\phi' \geq n) \in K, x_{\phi'} \geq n, \sigma(\phi'),$

for all $(\phi' \leq n) \in K, x_{\phi'} \geq 1 - n, \sigma(\neg \phi')$

where the function $\sigma$, transforming a many-valued proposition into a set of inequalities, is inductively defined as follows:

$$\sigma(\phi) = \begin{cases} x_\phi \in [0,1] & \text{if } \phi = p \\ x_{\phi'} = \sigma(x_\phi), x_\phi \in [0,1] & \text{if } \phi = \neg \phi' \\ x_{\phi_1} \otimes x_{\phi_2} = x_\phi, & \phi = \phi_1 \land \phi_2 \\ x_{\phi_1} \lor x_{\phi_2} = x_\phi, & \phi = \phi_1 \lor \phi_2 \\ \sigma(\neg \phi) = \sigma(\neg \phi). \end{cases}$$

For instance, under Łukasiewicz, $x_{\phi_1} \otimes x_{\phi_2} = x_\phi$ is encoded as the set of equations $y \in \{0,1\}, x_\phi \leq y, x_\phi = (1 - y) \leq x_{\phi_1} + x_{\phi_2} - 1 \leq y, x_{\phi_1} + x_{\phi_2} - y \leq x_\phi + (1 - y)$. If $y = 0$, then $x_\phi = 0$ (it simulates the case where $x_{\phi_1} + x_{\phi_2} \leq 1$, and hence $x_{\phi_1} \otimes x_{\phi_2} = 0$), while if $y = 1$, then $x_\phi = x_{\phi_1} + x_{\phi_2} - 1$ and, thus, $x_{\phi_1} \otimes x_{\phi_2} = x_\phi$. In a similar way, we may determine

$$bsd(K,\phi) = \min -x. \text{such that } x \in [0,1], x_\phi \geq x, \sigma(\phi),$$

for all $(\phi' \geq n) \in K, x_{\phi'} \geq n, \sigma(\phi'),$

for all $(\phi' \leq n) \in K, x_{\phi'} \geq 1 - n, \sigma(\neg \phi').$

Note that we may verify whether $K$ is satisfiable by checking if $bsd(K,p) = 1$, where $p$ is a new propositional letter not occurring in $K$, and that under Łukasiewicz logic and Zadeh semantics we end up with a bounded Mixed Integer Linear Program (bMILP) optimization problem [16].

III. A FUZZY SPATIAL LOGIC

The fuzzy spatial logic we are considering here, is based on the Fuzzy Region Connection Calculus (f-RCC), which starts from the crisp variant RCC [14]. We fuzzyfy it similarly to [5], [18]. So, consider a fixed finite non-empty set of regions $\mathcal{R}$. Topological relations of RCC are defined in terms of an arbitrary reflexive and symmetric relation $C$ on $\mathcal{R}$, called connection, $C(x,y)$ understood to mean that $x$ is connected with $y$. The axioms defining the topological relations are shown in Fig.1. The intuitive meaning of some of these relations is shown in Fig. 2. In our fuzzy RCC calculus, $C$ is a fuzzy relation, i.e., for each pair $(a,b)$ of regions, $C(a,b)$ is a degree in $[0,1]$ reflecting to what extent $a$ and $b$ are connected.

The other topological relations are defined in terms of $C$ as illustrated in in Fig.1. Now, a fuzzy spatial statement is of the form $\phi \geq l$ or $\phi \leq u$, where $l, u \in [0,1]$ and $\phi$ is a boolean combination of RCC topological relations of the form $S(a,b)$ with $a, b$ regions and $S$ relation as from Fig. 1. For instance, $PO(a,b) \geq 0.3$ and $P(a,b) \rightarrow C(b,d) \leq 0.9$ are fuzzy spatial statements. A fuzzy spatial knowledge base is a set of fuzzy spatial statements.

A fuzzy spatial interpretation $I$ maps each basic RCC statement $S(a,b)$ into $[0,1]$ according to the axioms described in Fig. 1, and is then extended inductively to all statements as in the previous section. For instance, the degree of truth of $P(a,b) \rightarrow C(b,d)$ is

$$I(P(a,b) \rightarrow C(b,d)) = I(P(a,b)) = I(C(b,d))$$

$$= I(\forall c, (C(c,a) \land \neg C(c,b))) = I(C(b,c))$$

Note that we may use $\min_{c \in \mathcal{R}}$ as place of $\inf_{c \in \mathcal{R}}$ as $\mathcal{R}$ finite. The notions of model, satisfiability, logical consequence, BED and BSD are defined as in the previous Section II. We refer the reader to [18] for reasoning algorithms.

For the same reasons as in Section II, we show how we may easily extend the reasoning techniques used in the previous section, to our fuzzy RCC calculus in the case of Zadeh logic and Łukasiewicz logic. The method is like for Section II, except that we have to take into account that some RCC relations involve quantifiers. Fortunately, this does not harm us much as $\mathcal{R}$ is finite and, thus, any RCC relation $S$ can be re-written as a RCC statement without quantifiers involving relation $C$ only by recursively applying $\sigma$ defined as follows.

For any RCC relation $S(a,b)$, let $\phi_S(a,b)$ be the definition of relation $S(a,b)$ according to Fig. 1. Then
represented in MILP).

\[ \sigma \) and \( \Phi \) provide a minimal variant of \( \text{ALCF} \) on knowledge base, where it is easy as it suffices to add the set of fuzzy spatial statements

\[ \text{binary spatial relations over } \mathbb{R} \]\n
are topological relations. In many cases, we would also like to apply.

So far, the binary spatial relations are those of RCC, which apply.

A fuzzy concrete domain (or simply fuzzy domain) [21] is a pair \( (\Delta_D, \Phi_D) \), where \( \Delta_D \) is an interpretation domain and \( \Phi_D \) is the set of fuzzy domain predicates \( d \) with a predefined arity \( n \) and an interpretation \( d^D : \Delta^n_D \rightarrow [0,1] \), which is a \( n \)-ary fuzzy relation over \( \Delta_D \). In our specific spatial fuzzy DL, we assume that \( \Delta_D = \mathbb{R} \) and the fuzzy predicates have arity two and consist of the set of RCC relations defined over \( \mathbb{R} \) and of a set of application dependent binary spatial relations \( S \) with fixed fuzzy membership function over \( \mathbb{R} \times \mathbb{R} \), as seen in the previous section.

Now, consider pairwise disjoint alphabets (disjoint from \( \mathbb{R} \) and \( S \)) of concepts names (denoted \( A \)), abstract roles names (denoted \( R \)), concrete roles names (denoted \( T \)), abstract individual names (denoted \( a \)) and concrete individual names (denoted \( r \)). Within the alphabet of abstract and concrete roles, we have distinguished subsets of abstract functional roles names (denoted \( f \)) and concrete functional roles names (denoted \( t \)), respectively. We call functional roles also features. 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 (\( R \)), concrete roles (\( T \)) as:

\[ C, C' \rightarrow \begin{array}{c|c}
A & (\text{atomic concept}) \\
\top & (\text{top concept}) \\
\bot & (\text{bottom concept}) \\
C \cap C' & (\text{concept conjunction}) \\
C \cup C' & (\text{concept disjunction}) \\
\neg C & (\text{concept negation}) \\
\forall R.C & (\text{universal quantification}) \\
\exists R.C & (\text{existential quantification}) \\
\forall(T, T') \mathbf{d} & (\text{concrete universal quantification}) \\
\exists(T, T') \mathbf{d} & (\text{concrete existential quantification}) \\
\end{array} \]

where \( \mathbf{d} \) is a boolean combination of fuzzy domain predicates, called spatial expression. For instance, \( \text{House} \cap \exists (\text{has_Loc}, \text{has_CarParkLoc}) \). Close is a concept that informally determines the degree of being an object a house having a car park close to it, where we assume \( \text{Close} \in S \), and \( \text{has_Loc} \) and \( \text{has_CarParkLoc} \) are concrete features. As illustrative purpose, Fig. 4 provides both a FOL translation of \( \text{ALC} \) concepts and example expressions.

A Fuzzy Knowledge Base \( K \) comprises a fuzzy ABox \( A \) and a fuzzy TBox \( T \). 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 abstract role assertion of the form \( \langle (a, b); R, n \rangle \) (the pair of individuals \( (a, b) \) is an instance of role reasoning. More expressive languages can be obtained by choosing a more expressive fuzzy DL such as [4].

**Syntax.** Recall that \( \text{ALCF}(D) \) is the basic DL \( \text{ALC} \) extended with functional roles (letter \( F \)) and concrete domains (letter \( D \)) allowing to deal with data types such as strings, integers and reals. In our specific fuzzy \( \text{ALCF}(D) \), the concrete domain consists of the set of regions \( \mathbb{R} \) and spatial relations over it. In general, a fuzzy concrete domain (or simply fuzzy domain) [21] is a pair \( (\Delta_D, \Phi_D) \), where \( \Delta_D \) is an interpretation domain and \( \Phi_D \) is the set of fuzzy domain predicates \( d \) with a predefined arity \( n \) and an interpretation \( d^D : \Delta^n_D \rightarrow [0,1] \), which is an \( n \)-ary fuzzy relation over \( \Delta_D \). In our specific spatial fuzzy DL, we assume that \( \Delta_D = \mathbb{R} \) and the fuzzy predicates have arity two and consist of the set of RCC relations defined over \( \mathbb{R} \) and of a set of application dependent binary spatial relations \( S \) with fixed fuzzy membership function over \( \mathbb{R} \times \mathbb{R} \), as seen in the previous section.

Now, consider pairwise disjoint alphabets (disjoint from \( \mathbb{R} \) and \( S \)) of concepts names (denoted \( A \)), abstract roles names (denoted \( R \)), concrete roles names (denoted \( T \)), abstract individual names (denoted \( a \)) and concrete individual names (denoted \( r \)). Within the alphabet of abstract and concrete roles, we have distinguished subsets of abstract functional roles names (denoted \( f \)) and concrete functional roles names (denoted \( t \)), respectively. We call functional roles also features. 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 (\( R \)), concrete roles (\( T \)) as:

\[ C, C' \rightarrow \begin{array}{c|c}
A & (\text{atomic concept}) \\
\top & (\text{top concept}) \\
\bot & (\text{bottom concept}) \\
C \cap C' & (\text{concept conjunction}) \\
C \cup C' & (\text{concept disjunction}) \\
\neg C & (\text{concept negation}) \\
\forall R.C & (\text{universal quantification}) \\
\exists R.C & (\text{existential quantification}) \\
\forall(T, T') \mathbf{d} & (\text{concrete universal quantification}) \\
\exists(T, T') \mathbf{d} & (\text{concrete existential quantification}) \\
\end{array} \]
R with degree at least n), or a fuzzy concrete role assertion of the form ⟨(a, b): R, n⟩ (individual a is related to region r via concrete role R with degree at least n), or of the form ⟨(r, r'): S, n⟩ (the pair of regions (r, r') is an instance of the spatial relation S with degree at least n). In FOL, ⟨a; C, n⟩ may be seen as a fuzzy statement of the form C(a) ≥ n, while ⟨(a, b); R, n⟩ may be seen as a fuzzy statement of the form R(a, b) ≥ n (and similarly for the other assertions). A fuzzy TBox consists of fuzzy General Concept Inclusions (fuzzy GCIs), which are expressions of the form C ⊑ D (n) (with informal meaning, the degree of subsumption between concept C and D is not less than n). In FOL, (C ⊑ D, n) may be seen as a fuzzy statement of the form (∀x.C(x) → D(x)) ≥ n, and amounts of asserting that the degree of subsumption among C and D is at least n. We will use C = D as a shorthand for (C ⊑ D, 1) and (D ⊑ C, 1).

**Semantics.** From a semantics point of view, a fuzzy interpretation I = (ΔI, T) relative to the spatial fuzzy concrete domain theory D = (R, D, ·) consists of a nonempty set ΔI (the domain), disjoint from R, and of a fuzzy interpretation function · that coincides with · on every region and fuzzy spatial relation, and it assigns: (i) to each abstract concept C a function C· : ΔI → [0, 1]; (ii) to each abstract role R a function R· : ΔI × ΔI → [0, 1] such that for all u ∈ ΔI there is an unique w ∈ ΔI on which R·(u, v) is defined; (iii) to each concrete role assertion (r, r') a function I· : [0, 1] → [0, 1] such that for all u ∈ ΔI there is an unique r ∈ R on which I·(u, r) is defined; (vii) to each abstract individual a an element in ΔI such that if a ≠ b then a· ≠ b· (Unique Name Assumption); (vii) to each concrete individual r an element in R that is r itself. A fuzzy interpretation I is a model of a fuzzy statement ⟨α, n⟩ iff α· ≥ n. The notions of logical consequence, best entailment degree and best satisfiability degree of α are as for Section II. We additionally define the Best Satisfiability Degree [4] of a concept C w.r.t. a fuzzy KB K as \(bsd(K, C) = \sup_{x \in \Delta_I} C·(x)\).

**Example 1.** Assume we would like to recognize whether there is a quite house in an image (see, e.g. [13]), where a quite house is defined as a house having a garden around. Assume that we have a definition for quite house and the output of a (semi-automatic) image classification tool applied to an image I:

\[
\text{QuiteHouseImage} = \text{Image} \sqcap \exists \text{depicts.House} \sqcap \exists \text{depicts.Garden}
\]

Now, assuming \(R = \{r, r'\}\) and \(S = \{\text{Around}\}\), then it can be shown that (under Zadeh semantics) \(bsd(K, i; \text{QuiteHouseImage}) = 0.8 \otimes 0.7 \otimes 0.8 \otimes 0.9 = 0.7\) and, thus, the image depicts a quite house to degree 0.7.

**V. REASONING**

We next provide reasoning algorithms. At fist, we will assume that a knowledge base will contain the set of fuzzy spatial statements (compare to Section III) \(\bigcup_{S \in \mathcal{S}} \{(r, r') \colon S, \mu_{S}(r, r')\} \mid r, r' \in R\). Second, for computational reasons, we will restrict knowledge bases being unfoldable, as in [21]. That is, a TBox T consists of a finite set of terminological axioms of the form \(A \subseteq C\) (A is sub-concept of C) or \(A = C\) (A is defined as the concept C), where A is a concept name and C is concept. We also assume that no concept A appears more than once on the left hand side of a terminological axiom and that no cyclic definitions are present in T (See [1]). We use \(A \subseteq C\) as a shorthand for \(\langle A \subseteq C, 1 \rangle\).

1. We point out that a calculus involving fuzzy GCI’s and fuzzy spatial reasoning has still to be worked out and is expected to be much more involved as already shown for the crisp case [12].
DLs [1]: (i) each terminological axiom \( A \subseteq C \in T \) can be replaced with \( A = C \sqcap A^* \), where \( A^* \) is a new concept name. Let \( K' \) the obtained knowledge base; (ii) the newly obtained \( K' \) can be expanded by substituting every concept name \( A \) occurring in \( K \), which is defined in \( T \), with its defining term in \( T \). Although, the expanded knowledge base may become of exponential size, the properties from a semantics point of view are left unchanged. Let \( K'' \) the obtained knowledge base; and (iii) each concept occurring in \( K'' \) is then transformed into NNF. 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: \( \neg T \equiv \bot, \neg(C_1 \sqcap C_2) \equiv \neg C_1 \sqcap \neg C_2, \neg \exists R.C \equiv \forall R.\neg C, \neg C \equiv C \), \( \neg(C_1 \sqcup C_2) \equiv \neg C_1 \sqcup \neg C_2, \neg \forall R.C \equiv \exists R.\neg C, \neg \forall(T,T'), \neg \exists(T,T'), \neg \exists(T,T'), \neg \exists(T,T'), \neg \forall(T,T'), \neg \exists(T,T') \), \( \exists(T,T'), \neg \forall(T,T'), \neg \exists(T,T') \). Then, we apply similar NNF transformation rules to spatial expressions \( d \) and \( \neg d \) occurring in the obtained knowledge base. All these operations do not affect the semantics due to the restrictions we made on the fuzzy constructors. Hence, we may assume that \( T = \emptyset \) and all expressions are in NNF.

The calculus we present here is inspired by the one implemented within the FuzzyDL system [4] (which follows from the one presented in [21]). Furthermore, the version we present here for ALCF, 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 calculi under Łukasiewicz logic and Zadeh logic, as in [21], [23]. A similar algorithm under product logic is expected to be more involved (see, e.g., [3]), but may easily be worked out along the lines described in [3].

Our goal is to provide a terminating procedure determining \( \text{bed}(K, \alpha) \), where \( \alpha \) is of the form \( a;C \), \( (a,b);R \), \((a,r);T \), \((r,r');S \), or \( C \subseteq D \). If \( \alpha \) is a fuzzy role assertion of the form \((a,b);R \) then, in order to determine \( \text{bed}(K, (a,b);R) \), we may reduce it to the BED for concept assertions, as
\[
\text{bed}(K, (a,b);R) = \text{bed}(K \cup \{(b:B), 1 \}), a;3R.B ,
\]
where \( B \) is a new concept (i.e., it does not occur in \( K \)). Similarly, for \((a,r);T \), we determine
\[
\text{bed}(K, (a,r);T) = \text{bed}(K \cup \{(a,r);T, 1 \}), a;3(T,t).C ,
\]
where \( t \) is a new concrete feature (the equivalence follows from the fact that \( r \) is the only value related to \( a \) via \( t \) and \( C \) is reflexive). Finally, concerning \((r,r');S \), we determine
\[
\text{bed}(K, (r,r');S) = \text{bed}(K \cup \{(a,r);T, 1 \}, (a,r);T, 1 \}, a;3(T,t).S ,
\]
where \( a \) is a new individual and \( T \) is a new concrete role (the equivalence follows from the fact that \( r, r' \) are the only values related to \( a \) via \( T \)).

Now, the basic idea behind our reasoning algorithm is as follows. It is based on the observations and algorithm of Section II. So, not surprisingly, in order to determine \( \text{bed}(K, \alpha) \), we will compute it as
\[
\text{bed}(K, \alpha) = \min x \text{ such that } K \cup \{(\alpha \leq x)\} \text{ satisfiable}.
\]
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,
\[
\text{bed}(K, a;C) = \min x \text{ such that } K \cup \{(a;\neg C, 1 - x)\} \text{ satisfiable}
\]
\[
\text{bed}(K, C \subseteq D) = \min x \text{ such that } K \cup \{(b;C \sqcap \neg D, 1 - x)\} \text{ satisfiable}
\]
\[
\text{bed}(K, C) = \min x \text{ such that } K \cup \{(b:C, x)\} \text{ satisfiable}.
\]
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 \( K \) has a model by e.g. checking whether \( \text{bed}(K, b:A) = 1 \), where \( b \) and \( A \) do not occur in \( K \).

Like most of the tableau algorithms, our algorithm works on completion-forests since an ABox might contain several individuals with arbitrary roles connecting them. A completion-forest \( F \) 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 concepts. 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 \( R \) and if \( R \in \mathcal{L}(\langle v, w \rangle) \) then we consider a variable \( x_{\langle v, w \rangle};R \) representing the degree of being \( \langle v, w \rangle \) and instance of \( R \). We will assume that there is a bijection between assertions \( \alpha \) and variables \( x_{\alpha \times} \). The forest has also associated a set \( C_F \) of constraints of the form \( I \leq l', l = l' \), \( x_i \in \{0,1\}, y_i \in \{0,1\} \), where \( l, l' \) are arithmetic expressions, on the variables occurring the node labels and edge labels.

We further need a technical definition involving features (see [12], [21]). Let \( F \) be a forest, \( f \) an abstract feature such that we have two edges \( \langle v, w_1 \rangle \) and \( \langle v, w_2 \rangle \) such that \( f \) occurs in both \( \mathcal{L}(\langle v, w_1 \rangle) \) and \( \mathcal{L}(\langle v, w_2 \rangle) \). Then we call such a pair a fork. As \( f \) is a function, such a fork means that \( w_1 \) and \( w_2 \) have to be interpreted as the same individual. Such a fork can be deleted by adding both \( \mathcal{L}(\langle v, w_2 \rangle) \) to \( \mathcal{L}(\langle v, w_1 \rangle) \) and \( \mathcal{L}(w_2) \) to \( \mathcal{L}(w_1) \), and then deleting node \( w_2 \). At the beginning, we remove the forks from the initial forest. We assume that forks are eliminated as soon as they appear (as part of a rule application) with the proviso that newly generated nodes are replaced by older ones and not vice-versa. Then we start with an initialization step, which builds the initial completion-forest. Then we apply it to 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 \( F \) as follows. Consider \( K \): (i) \( F \) contains a root node \( a_i \) for each abstract individual \( a_i \) occurring in \( A \); (ii) \( F \) contains an edge \( \langle a, b \rangle \) for each \( \langle a, b \rangle;R, n \rangle \in A \), where \( R \) is an abstract role; (iii) for each \( \langle a,C \rangle, n \) \in \( A \), we add both \( C \) to \( \mathcal{L}(a) \) and \( x_{a;C} \geq n \) to \( C_F \); (iv) for each \( \langle a, b \rangle;R, n \rangle \in A \) (where \( R \) is an abstract role), we add both \( R \) to \( \mathcal{L}(a, b) \) and \( x_{a,b \times}^{R} \geq n \) to \( C_F \). (v) for each \( \langle a, b \rangle;T, n \rangle \in A \) (where \( T \) is a concrete role), we add \( x_{T,a,b}^{R} \geq n \) to the set of constraints \( C_F \); (vi) for each spatial assertion \( \langle r, r' \rangle;S, n \rangle \in A \), we add \( x_{SS,a,b} \geq n \) to the set of constraints \( C_F \). Now, \( F \) is then expanded by repeatedly applying the rules described below. The completion-forest is complete when none of the rules are applicable. Then, the bMILP problem on \( C_F \) is solved. The rules are:

\[ 1 \text{In the last equation, we use the fact that in our setting } \max x \text{ s.t. } x \in S = \min -x \text{ s.t. } x \in S. \]
There are several points that may be of interest for future research: so far, (i) we allow unfoldable TBoxes only. It remains to see whether practical reasoning algorithms exists in the case of fuzzy GCI’s (it is expected to be more involved as already shown for the crisp case [12]); (ii) our calculus applies to the case where a finite non-empty set of regions is fixed a priori. We still have to work out an algorithm for the case $\mathcal{R}$ is not defined a priori (see, e.g., [11], [12], [15]).

REFERENCES