General Concept Inclusions in Fuzzy Description Logics

Giorgos Stoiilos\(^1\) and Umberto Straccia\(^2\) and Giorgos Stamou\(^3\) and Jeff Z. Pan\(^4\)

Abstract. Fuzzy Description Logics (fuzzy DLs) have been proposed as a language to describe structured knowledge with vague concepts. A major theoretical and computational limitation so far is the inability to deal with General Concept Inclusions (GCIs), which is an important feature of classical DLs. In this paper, we address this issue and develop a calculus for fuzzy DLs with GCIs.

1 INTRODUCTION

Description Logics (DLs) \(^2\) are a logical reconstruction of the so-called frame-based knowledge representation languages, with the aim of providing a simple well-established Tarski-style declarative semantics to capture the meaning of the most popular features of structured representation of knowledge. Nowadays, DLs have gained even more popularity due to their application in the context of the Semantic Web, as the theoretical counterpart of OWL DL (the W3C standard for specifying ontologies, see \([9]\) for details).

Fuzzy DLs \(^{15, 18, 23, 24}\) extend classical DLs by allowing to deal with fuzzy/vague/imprecise concepts such as “Candia is a creamy white rose with dark pink edges to the petals”, “Jacaranda is a hot pink rose”, and “Calla is a very large, long white flower on thick stalks”. Such concepts involve so-called fuzzy or vague concepts, like “creamy”, “dark”, “hot”, “large” and “thick”, for which a clear and precise definition is not possible.

The problem to deal with imprecise concepts has been addressed several decades ago by Zadeh \([25]\), which gave birth in the meanwhile to the so-called fuzzy set and fuzzy logic theory and a huge number of real life applications exists. Despite the popularity of fuzzy set theory, relative little work has been carried out in extending DLs towards the representation of imprecise concepts, notwithstanding DLs can be considered as a quite natural candidate for such an extension \([1, 3, 4, 5, 6, 11, 12, 14, 15, 16, 17, 18, 20, 21, 22, 23, 24, 26]\).

A major theoretical and computational limitation so far of fuzzy DLs is the inability to deal with General Concept Inclusions (GCIs), which is an important feature of classical DLs; e.g., GCIs are necessary to represent domain and range constraints. In this paper, we address this issue and develop a calculus for fuzzy DLs with GCIs.

In the next section, we briefly recall basic concepts of DLs and fuzzy DLs, while in Section 3 we present a sound and complete calculus dealing with GCIs. Section 4 concludes.

2 PRELIMINARIES

DLs basics. DLs \(^2\) are a family of logics for representing structured knowledge. Each logic is identified by a name made of labels, which identify the operators allowed in that logic. Major DLs are the so-called logic ALC \([13]\) and is used as a reference language whenever new concepts are introduced in DLs, SHOIN(D), which is the logic behind the ontology description language OWL DL and SH(L,T,F)(D), which is the logic behind OWL Lite, a slightly less expressive language than OWL DL (see \([9]\)). DLs can be seen as a restricted First Order Language with unary and binary predicates. For the sake of our purpose we deal here with ALC, whose syntax and semantics is described in Table 1. (an interpretation \(I = (\Delta^T, \Delta^D)\) has domain \(\Delta^T\) and maps concepts into subsets of \(\Delta^T\), maps roles into subsets of \(\Delta^T \times \Delta^T\) and maps individuals into elements of \(\Delta^T\).

An ALC knowledge base is defined as a pair \(\Sigma = (T, A)\), where \(T\) is called a TBox and \(A\) an ABox. A TBox is a finite set of general inclusion axioms (GCIs for short) of the form, \(C \sqsubseteq D\) and \(A\) is a finite set of concept and role assertions of the form \(C(a)\) and \((a, b) : R\), respectively. For example \(T\) could contain an axioms of the form HappyFather \(\sqsubseteq\) hasChild11 and Female, and \(A\) an assertion of the form Tom : HappyFather. An interpretation \(I\) satisfies \(T\) if \(C^I \subseteq D^I\) for all GCIs in \(T\), then \(I\) is called a model of \(T\), and it satisfies \(A\) if \(a^I \in C^I \subseteq (\Delta^T)\) for all concept (role) assertions in \(A\). Then \(I\) is called a model of \(A\). An interpretation satisfies an ALC KB \(\Sigma = (T, A)\) if it satisfies both \(T\) and \(A\); then \(I\) is called a model of \(\Sigma\). A concept \(C\) is subsumed by a concept \(D\), written \(C \sqsubseteq D\), with respect to \((A, B)\) for all models \(I\) of \(T\), it holds that \(C^I \subseteq D^I\). An ABox \(A\) is consistent (inconsistent) \(w.r.t.\) a TBox \(T\) if there exists (does not exist) a model \(I\) of \(T\), that is also a model of \(A\). Finally, \(\Sigma\) entails an ALC assertion \(\phi\), written \(\Sigma\models \phi\), if each model of \(\Sigma\) is a model of \(\phi\).

Fuzzy DLs basics. Fuzzy DLs \([18]\) extend classical DLs by allowing to deal with fuzzy/vague/imprecise concepts. The main idea underlying fuzzy DLs is that an assertion \(a : C\) means that the constant \(a\) is an instance of concept \(C\), rather being interpreted as either true or false, will be mapped to a truth value \(n \in [0, 1]\) (the rationals in the interval \([0, 1]\)). The intended meaning is that \(n\) indicates to which extent \(a\) is a \(C\). From a syntax point of view, concepts, roles, individuals and concept inclusion axioms are as for ALC. In place of assertions, we have fuzzy assertions \([18]\), which are of the form \(\langle a \triangleright \geq n \rangle\), where \(a\) is an assertion, \(n \in [0, 1]\) and \(\geq n\) is one of \(\geq, \leq, >, <\). For instance, \(\langle a \triangleright 0.5 \rangle\) allows to state that individual \(a\) is an instance of concept \(C\) at least to degree \(n\). Similarly for role assertions, a Fuzzy Knowledge Base, \(\Sigma = (T, A)\), is as for the crisp case, except that now \(A\) is a set of fuzzy assertions rather than assertions only. From a semantics point of view, a fuzzy interpretation \(I = (\Delta^T, \Delta^D)\) has domain \(\Delta^T\), but now maps a concept \(C\) into a function \(C^I : \Delta^T \rightarrow [0, 1]\) and a role \(R\) into a function \(R^I : \Delta^T \times \Delta^T \rightarrow [0, 1]\). For \(d \in \Delta^T\), \(C^I(d)\) gives us the degree of \(d\) being an instance of the concept \(C\) (similarly for roles).

The semantics is summarized in Table 1. A fuzzy interpretation \(I\) satisfies a fuzzy TBox \(T\) if \(\forall x \in \Delta^T, C^I(x) \leq D^I(x)\) for all

---

\(^1\) National and Technical University of Athens
\(^2\) ISTI - Italian National Research Council at Pisa
\(^3\) National and Technical University of Athens
\(^4\) Department of Computing Science, The University of Aberdeen, UK
### Table 1. ALC and fuzzy ALC.

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Classical Semantics</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>C, D</td>
<td>(top concept)</td>
<td>⊑</td>
</tr>
<tr>
<td>⊥</td>
<td>(bottom concept)</td>
<td>⊑</td>
</tr>
<tr>
<td>A</td>
<td>(atomic concept)</td>
<td>⊑ A</td>
</tr>
<tr>
<td>C ∪ D</td>
<td>(concept conjunction)</td>
<td>⊑ C ∪ D</td>
</tr>
<tr>
<td>⊓</td>
<td>(concept disjunction)</td>
<td>⊓ C ⊓ D</td>
</tr>
<tr>
<td>¬C</td>
<td>(concept negation)</td>
<td>⊑ ¬C</td>
</tr>
<tr>
<td>∃R.C</td>
<td>(existential quantification)</td>
<td>⊑ ∃R.C</td>
</tr>
<tr>
<td>∀R.C</td>
<td>(universal quantification)</td>
<td>⊑ ∀R.C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fuzzy Semantics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T^f(x) \in C</td>
<td>1</td>
</tr>
<tr>
<td>A^f(x) \in C</td>
<td>0</td>
</tr>
<tr>
<td>(C_1 ⊓ C_2)^f(x) = min(C_1^f(x), C_2^f(x))</td>
<td>(∀R.C)^f(x) = inf_{y \in A} (min(R^f(x, y), C^f(y)))</td>
</tr>
</tbody>
</table>

C ⊑ D ∈ T, then I is called a model of T, and it satisfys a fuzzy ABox if C^f(a^2) \sqsubseteq n (R^f(a^2), b^2 |n|, for all (a : C |n|) (\{(a, b) : R |n|\}) in A. Then I is called a model of A. A fuzzy ALC KB \Sigma = (T, A) is consistent if there exists a model I of A and T.

Given a fuzzy KB \Sigma, and a fuzzy assertion \psi (resp. a GCI C ⊑ D), we say that \Sigma entails \psi (resp. C ⊑ D), denoted \Sigma |\models \psi (resp. \Sigma |\models C ⊑ D), if each model of \Sigma is a model of \psi (resp. C ⊑ D). Finally, given \Sigma and a fuzzy assertion \alpha, it is of interest to compute \alpha’s best lower and upper truth value bounds. The greatest lower bound of \alpha w.r.t. \Sigma (denoted glb(\Sigma, \alpha)) is glb(\Sigma, \alpha) = sup\{n | \Sigma |\models \langle \alpha \geq n \rangle \} where sup = 0. Similarly, the least upper bound of \alpha w.r.t. \Sigma (denoted lub(\Sigma, \alpha)) is lub(\Sigma, \alpha) = inf\{n | \Sigma |\models \langle \alpha \leq n \rangle \} where inf = 1. Determining the glb is called the Best Truth Value Bound (BTBV) problem. Basic inference problems are: (i) Check if a fuzzy KB is consistent, i.e. has a model. (ii) Check if D subsumes C w.r.t. \Sigma, i.e. \Sigma |\models C ⊑ D. (iii) Check if a is instance of C to degree \geq n, i.e. \Sigma |\models \langle a:C \geq n \rangle (Similarly for the other relations ≤, > and <). (iv) Determine glb(\Sigma, a:C).

We recall that all the inference problems can be reduced to the consistency problem [18]: (i) Concerning the entailment problem, it can be verified that it can be reduced to the inconsistency problem: (T, A) |\models \emptyset iff (T, A \cup \{\langle a < n \rangle \}) is inconsistent, and similarly for the other relations ≤, > and <. (ii) Concerning the BTBV problem, it holds that lub(\Sigma, a:C) = 1 − glb(\Sigma, \alpha−C), i.e. the lub can be determined through the glb (and vice-versa). Furthermore, the computation of the glb can be determined by relying on a finite number of entailment tests. First, for \Sigma = (T, A), we define X^A = \{0, 0.5, 1\} \cup \{n | \alpha \in A\} and N^A = X^A \cup \{1 - n | n \in A\}. Then glb(\Sigma, a:C) = max\{n | n \in N^A and \Sigma |\models \langle a:C \geq n \rangle\} and lub(\Sigma, \alpha−C) = min\{n | n \in N^A and \Sigma |\models \langle a:C \leq n \rangle\}.

Concerning the subsumption problem, we have that \Sigma |\models C ⊑ D iff I |\models (\langle a:C \geq n \rangle, \langle a:D < n \rangle), with n \in \{n_1, n_2\}, n_1 \in (0, 0.5], n_2 \in (0.5, 1] (e.g., we may choose n_1 = 0.25, n_2 = 0.75), is not consistent. So, the subsumption problem can reduced to the consistency problem as well.

In all previous approaches to fuzzy DLs [18, 6, 14] decision procedures for the consistency, the entailment and the BTBV problem are given for various DL languages, but with restrictions on the form of concept inclusion axioms in a TBox T. More precisely, T was considered to be simple, i.e. cyclic axioms are not allowed, while concept inclusions were restricted to those of the form A ⊑ B, where A is an atomic concept. Both GICs and cyclic axioms are considered important for the classical case and, thus, should be provided in the fuzzy variant as well. For instance, classical definitions allow us to consider definitions such as

<table>
<thead>
<tr>
<th>Human</th>
<th>HasParent.Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeometricElement</td>
<td>HasPart.GeometricElement.</td>
</tr>
</tbody>
</table>
in order to satisfy \( \{ (\alpha \land C \geq n) \} \), we set \( C^2(\alpha^2) = n \), while to satisfy \( \{ (\alpha \land C > n) \} \), we set \( C^2(\alpha^2) = n + \epsilon \), for a sufficiently small \( \epsilon \in [0, 1] \).

In the following, we assume that an ABox \( A \) has been normalized, i.e. fuzzy assertions of the form \( (\alpha \land C > n) \) are replaced by \( (\alpha \land C \geq n + \epsilon) \) and those of the form \( (\alpha \land C < n) \), by \( (\alpha \land C \leq n - \epsilon) \). Please note that in a normalized fuzzy KB we allow the degree to range in \([-\epsilon, 1 + \epsilon]\) in place of \([0, 1]\). It can be proved that the process of normalization is satisfiability preserving.

**Proposition 2** Let \( \Sigma = \langle T, A \rangle \) be a fuzzy knowledge base. Then \( \Sigma \) is satisfiable if and only if its normalized variant is satisfiable.

### 3.1 A fuzzy tableau for fuzzy ALC

We have seen that the inference problems in fuzzy DLs can be reduced to the consistency checking problem. Similar to crisp DLs, our tableau algorithm checks the consistency 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 consistent or to detect that the KB is inconsistent. The fuzzy tableau we present here can be seen as an extension of the tableau presented in [8], and is inspired by the one presented in [14]. Without loss of generality, we assume that all concepts \( C \) are in negation normal form (NNF) [7], i.e. negations occur in front of atomic concepts only.\(^8\) In the following, \( \preceq \in \{ [\geq], \leq \} \), while we also use \( \preceq \) to denote the reflection of \( \preceq \), e.g. if \( a \preceq b \), then \( b \preceq a \).

**Definition 1** Given \( \Sigma = \langle T, A \rangle \), let \( R_c \) be the set of roles occurring in \( \Sigma \) and let \( sub(\Sigma) \) be the set of named concepts appearing in \( \Sigma \). A fuzzy tableau \( T \) for \( \Sigma \) is a quadruple \((S, \mathcal{L}, \mathcal{E}, V)\) such that: \( S \) is a set of elements, \( \mathcal{L} : S \times \mathcal{L}(\Sigma) \rightarrow [0, 1]^\mathcal{L} \) maps each element and concept, to a membership degree (the degree of the element being an instance of the concept), and \( \mathcal{E} : \mathcal{R} \times \mathcal{E}(\Sigma) \rightarrow [0, 1]^\mathcal{E} \) maps each role of \( \mathcal{E} \) and pair of elements to the membership degree of the pair being an instance of the role, and \( V \) : \( \mathcal{I} \rightarrow S \) maps individuals occurring in \( A \) to elements in \( S \). For all \( s, t \in S, C, E \in \mathcal{L}(\Sigma) \), and \( R \in R_c, T \) has to satisfy:

1. \( \mathcal{L}(s, \perp) = 0 \) and \( \mathcal{L}(s, \top) = 1 \) for all \( s \in S \).
2. If \( \mathcal{L}(s, \neg C) \preceq \top \), then \( \mathcal{L}(s, C^\top) \preceq 1 \).
3. If \( \mathcal{L}(s, \neg C) \preceq \bot \) and \( \mathcal{L}(s, \top) \leq n \), then \( \mathcal{L}(s, C) \geq n + \epsilon \).
4. If \( \mathcal{L}(s, \neg C) \preceq \bot \) and \( \mathcal{L}(s, \top) \leq n \), then \( \mathcal{L}(s, C) \leq n + \epsilon \).
5. If \( \mathcal{L}(s, \neg C) \preceq \bot \) and \( \mathcal{L}(s, \top) \leq n \), then \( \mathcal{L}(s, C) \leq n \).
6. If \( \mathcal{L}(s, \neg C) \preceq \bot \) and \( \mathcal{L}(s, \top) \leq n \), then \( \mathcal{L}(s, C) \leq n \).
7. If \( \mathcal{L}(s, \neg R.C) \preceq \bot \) and \( \mathcal{L}(s, R) \leq n \), then \( \mathcal{L}(s, R, s) \leq n \) for all \( s \in S \).
8. If \( \mathcal{L}(s, \neg R.C) \preceq \bot \) and \( \mathcal{L}(s, R) \leq n \), then \( \mathcal{L}(s, R, s) \leq n \) for all \( s \in S \).
9. If \( \mathcal{L}(s, R.C) \preceq \bot \) and \( \mathcal{L}(s, R) \leq n \), then \( \mathcal{L}(s, R, s) \leq n \) for all \( s \in S \).
10. If \( \mathcal{L}(s, \neg R.C) \preceq \bot \) and \( \mathcal{L}(s, R) \leq n \), then \( \mathcal{L}(s, R, s) \leq n \) for all \( s \in S \).
11. If \( \mathcal{L}(s, \neg R.C) \preceq \bot \) and \( \mathcal{L}(s, R) \leq n \), then \( \mathcal{L}(s, R, s) \leq n \) for all \( s \in S \).
12. If \( \mathcal{L}(s, \neg R.C) \preceq \bot \) and \( \mathcal{L}(s, R) \leq n \), then \( \mathcal{L}(s, R, s) \leq n \) for all \( s \in S \).

**Proposition 3** \( \Sigma = \langle T, A \rangle \) is consistent if and only if there exists a fuzzy tableau for \( \Sigma \).

---

\(^8\) A fuzzy ALC concept can be transformed into an equivalent one in NNF by pushing negations inwards using a combination of the De Morgan laws and the equivalences \( \neg \exists R.C \equiv \forall R.\neg C \land \neg \forall R.C \equiv \exists R.\neg C \).
applying the completion rules from Table 2. The completion forest is complete when, for some node \( x \) (edge \((x, y)\), \( L(x) \) (\( L(x, y) \))) contains a clash, or none of the completion rules in Table 2 are applicable. The algorithm stops when a clash occurs; it answers ‘\( \Sigma \) is consistent’ if the completion rules can be applied in such a way that they yield a complete and clash-free completion forest, and ‘\( \Sigma \) is inconsistent’ otherwise.

From Table 2, we can see that for an arbitrary fuzzy assertion of the form \((a \in D \land n)\) either value \( n \) or its complement \( 1 - n \) appear in the expansion of a node \( x \) where \( D \in L(x) \). The finite property of the membership degrees makes blocking possible in our algorithm.

### Example 1
Let us show how the blocking condition works on the cyclic fuzzy KB, \( \Sigma \), \( \Sigma \) is satisfiable and \( N^A = \{0, 0.5, 1\} \cup \{0.4, 0.6\} \). We start with a root node \( x^0 \), with label \( L(x^0) = \{\text{HotPinkRose} \geq 0.6\} \). By applying rule (\( \neg \)) to node \( x^0 \), HotPinkRose \( \subseteq \neg \text{NextGenHotPinkRose} \). \( \Sigma \) is satisfiable and \( N^A = \{0, 0.5, 1\} \cup \{0.4, 0.6\} \). We start with a root node \( x^0 \), with label \( L(x^0) = \{\text{HotPinkRose} \geq 0.6\} \). By applying rule (\( \neg \)) to node \( x^0 \), HotPinkRose \( \subseteq \neg \text{NextGenHotPinkRose} \). Continuing with node \( x^0 \), we apply rule (\( \geq \)) to \( \neg \text{NextGenHotPinkRose} \), create a new edge \((x^0, y^1)\) with \( L(x^0, y^1) = \{\text{HotPinkRose} \geq 0.6\} \). By applying rule (\( \neg \)) to node \( x^0 \), we update the label \( L(y^1) \) with \( L(y^1) = \{\text{HotPinkRose} \geq 0.6\} \). Now, \( y^1 \) is a nextGen-successor of \( x^0 \) and \( L(y^1) = \{\text{HotPinkRose} \geq 0.6\} \) and, thus, \( y^1 \) is directly blocked.

### Example 2
We show that \( \Sigma \) is complete if it contains a clash or none of the completion rules in Table 2 are applicable. The algorithm stops when a clash occurs; it answers ‘\( \Sigma \) is consistent’ if the completion rules can be applied in such a way that they yield a complete and clash-free completion forest, and ‘\( \Sigma \) is inconsistent’ otherwise.

### Table 2. Tableaux expansion rules

<table>
<thead>
<tr>
<th>Rule</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(( \neg ))</td>
<td>if 1. ( \langle C, \langle x, n \rangle \rangle \in L(x) ), and 2. ( \langle C, \langle x, n \rangle \rangle \notin L(x) ), then ( L(x) = {\langle C, \langle x, n \rangle \rangle } ).</td>
</tr>
<tr>
<td>(( \geq ))</td>
<td>if 1. ( \langle C_1 \cup C_2, \geq n \rangle \in L(x) ) and 2. ( \langle C_1, \geq n \rangle \notin L(x) ), then ( L(x) = {\langle C_1, \geq n \rangle } ).</td>
</tr>
<tr>
<td>(( \leq ))</td>
<td>if 1. ( \langle C_1 \cup C_2, \leq n \rangle \in L(x) ) and 2. ( \langle C_1, \leq n \rangle \notin L(x) ), then ( L(x) = {\langle C_1, \leq n \rangle } ).</td>
</tr>
<tr>
<td>(( \geq ))</td>
<td>if 1. ( \langle C_1 \cup C_2, \geq n \rangle \in L(x) ) and 2. ( \langle C_1, \geq n \rangle \notin L(x) ), then ( L(x) = {\langle C_1, \geq n \rangle } ).</td>
</tr>
<tr>
<td>(( \leq ))</td>
<td>if 1. ( \langle C_1 \cup C_2, \leq n \rangle \in L(x) ) and 2. ( \langle C_1, \leq n \rangle \notin L(x) ), then ( L(x) = {\langle C_1, \leq n \rangle } ).</td>
</tr>
</tbody>
</table>

which contains a clash, while in the latter case we update the label \( L(x^0) \) with \( L(x^0) = \{\langle C, \geq 0.3 + \epsilon \rangle, \langle D, \leq 0.3 \rangle \} \) which also contains a clash. No complete, clash-free forest can be obtained, thus the algorithm answers with ‘inconsistent’.

### Proposition 4 (Termination)
For each fuzzy ALC KB \( \Sigma \), the tableau algorithm terminates.

**Proof:** [Sketch] Termination is a result of the properties of the expansion rules, as in the classical case [8]. More precisely we have the following observations. (i) The expansion rules never remove nodes from the tree or concepts from node labels or change the edge labels. (ii) Successors are only generated by the rules \( \geq \) and \( \leq \). For any node and for each concept these rules are applied at most once. (iii) Since nodes are labelled with nonempty subsets of \( \text{sub}(\Sigma) \), obviously there is a finite number of possible labellings for a pair of nodes. Thus, the condition of blocking will be applied in any path of the tree and consequently any path will have a finite length.

### Proposition 5 (Soundness)
If the expansion rules can be applied to an fuzzy ALC KB \( \Sigma \) such that they yield a complete and clash-free completion-forest, then \( \Sigma \) has a fuzzy tableau for \( \Sigma \).

**Proof:** [Sketch] Let \( F \) be a complete and clash-free completion-forest constructed by the tableau algorithm for \( \Sigma \). A fuzzy tableau \( T = (S, L, E, V) \) can be defined as follows:

\[
S = \{x \mid x \text{ is a node in } F, \text{ and } x \text{ is not blocked}\},
\]

\[
L(x, \langle \rangle) = L(x, \langle \rangle) = 1, \text{ for all } x \in S.
\]

\[
L(x, C) = \text{max}(\langle C, \geq n_1 \rangle), \text{ for all } x \in F \text{ not blocked},
\]

\[
L(x, \neg A) = 1 - L(x, A), \text{ for all } x \in F \text{ not blocked},
\]

\[
E(R, (x, y)) = \text{max}(\langle R, \geq n_1 \rangle), \text{ with } \langle R, \geq n_1 \rangle \in L(x),
\]

\[
V(n_i) = x^0, \text{ where } x^0 \text{ is a root node}
\]

where \( \text{max} \) returns the maximum degree \( n \) out of the set of triples of the form \( \langle A, \geq n_1 \rangle \), or 0 if no such triple exists. It can be shown that \( T \) is a fuzzy tableau for \( \Sigma \).
Proposition 6 (Completeness) Consider a fuzzy $\mathcal{ALC} KB \Sigma$. If $\Sigma$ has a fuzzy tableau, then the expansion rules can be applied in such a way that the tableau algorithm yields a complete and clash-free completion-forest for $\Sigma$.

Proof: [Sketch] Let $T = (S, L, E, V)$ be a fuzzy tableau for $\Sigma$. Using $T$, we trigger the application of the expansion rules such that they yield a completion-forest $F$ that is both complete and clash-free. Similarly to [8] we can define a mapping $\pi$ which maps nodes of $F$ to elements of $S$, and guide the application of the non-deterministic rules $\subseteq, \cup_\pi$ and $\subseteq_\pi$. Our method slightly differs from the one used in crisp DLs [8] in the following way. Using the membership degree of a node to a concept, found in the fuzzy tableau, we create artificial triples that are tested against conjunction with the candidate triples that the non-deterministic rule can insert in the completion-forest. The triples that don’t cause a conjunction can be added. The modified rules, which are used to guide such an expansion, are presented in Table 3. The modified rules together with the termination property ensure the completeness of the algorithm. □

Table 3. The $\subseteq_\pi$, $\cup_\pi$ and $\subseteq_\pi$ rules

4 CONCLUSIONS

Fuzzy DLs extend crisp DLs to deal with vague concepts. None of the work on fuzzy DLs so far presented a correct and complete calculus for cyclic TBoxes and general concept inclusions, which are important features of current crisp DL systems. We overcome to this limitation by providing a tableau for fuzzy $\mathcal{ALC}$ with GCIs.

Major topics for future research are indeed the extension of the fuzzy tableau algorithm to expressive DL languages such as fuzzy $\mathcal{SHIQ}$ or SHOIN($\mathcal{D}$) and the development of a system supporting this language. In the former case, such algorithm can be based directly on the ones presented for the fuzzy $\mathcal{SL}$ and $\mathcal{SH}$ DLs [14, 15] and the rules for nominals, for SHOIN [10] and for fuzzy SHOIN [16].

ACKNOWLEDGEMENTS

The work of Giorgos Stoilos, Giorgos Stamou and Jeff Z. Pan was partially supported by the FP6 Network of Excellence EU project Knowledge Web (IST-2004-507482).

REFERENCES


