Knowledge Representation and Reasoning in Norm-Parameterized Fuzzy Description Logics

Zhao, Jidi; Boley, Harold

Publisher's version / Version de l'éditeur:

Canadian Semantic Web Technologies and Applications, pp. 27-53, 2010-07-01

Knowledge Representation and Reasoning in Norm-Parameterized Fuzzy Description Logics

Jidi Zhao[1], Harold Boley[2]

Abstract The Semantic Web is an evolving extension of the World Wide Web in which the semantics of the available information are formally described, making it more machine-interpretable. The current W3C standard for Semantic Web ontology languages, OWL, is based on the knowledge representation formalism of Description Logics (DLs). Although standard DLs provide considerable expressive power, they cannot express various kinds of imprecise or vague knowledge and thus cannot deal with uncertainty, an intrinsic feature of the real world and our knowledge. To overcome this deficiency, this paper extends a standard Description Logic to a family of norm-parameterized Fuzzy Description Logics. The syntax to represent uncertain knowledge and the semantics to interpret fuzzy concept descriptions and knowledge bases are addressed in detail. The paper then focuses on a procedure for reasoning with knowledge bases in the proposed Fuzzy Description Logics. Finally, we prove the soundness, completeness, and termination of the reasoning procedure.

1 Introduction

The Semantic Web is an evolving extension of the World Wide Web in which the semantics of the available information are formally described by logic-based standards and technologies, making it possible for machines to understand the information on the Web [3].

Jidi Zhao
Faculty of Computer Science, University of New Brunswick, Fredericton, NB, E3B 5A3 Canada, e-mail: `Judy.Zhao[AT]unb.ca`

Harold Boley
Institute for Information Technology, National Research Council of Canada, Fredericton, NB, E3B 9W4 Canada, e-mail: `Harold.Boley[AT]nrc.gc.ca`

Uncertainty is an intrinsic feature of real-world knowledge, which is also reflected in the World Wide Web and the Semantic Web. Many concepts needed in knowledge modeling lack well-defined boundaries or, precisely defined criteria. Examples are the concepts of young, tall, and cold. The Uncertainty Reasoning for the World Wide Web (URW3) Incubator Group defined the challenge of representing and reasoning with uncertain information on the Web. According to the latest URW3 draft report, uncertainty is a term intended to encompass different forms of uncertain knowledge, including incompleteness, inconclusiveness, vagueness, ambiguity, and others [18]. The need to model and reason with uncertainty has been found in many different Semantic Web contexts, such as matchmaking in Web services [20], classification of genes in bioinformatics [28], multimedia annotation [27], and ontology learning [6]. Therefore, a key research direction in the Semantic Web is to handle uncertainty.

The current W3C standard for Semantic Web ontology languages, OWL Web Ontology Language, is designed for use by applications that need to process the content of information instead of just presenting information to humans [21, 23]. It facilitates greater machine interpretability of Web content than that supported by other Web languages such as XML, RDF, and RDF Schema (RDFS). This ability of OWL is enabled by its underlying knowledge representation formalism Description Logics (DLs). Description Logics (DLs) [2][1][12] are a family of logic-based knowledge representation formalisms designed to represent and reason about the conceptual knowledge of arbitrary domains. Elementary descriptions of DL are atomic concepts (classes) and atomic roles (properties or relations). Complex concept descriptions and role descriptions can be built from elementary descriptions according to construction rules. Different Description Logics are distinguished by the kinds of concept and role constructors allowed in the Description Logic and the kinds of axioms allowed in the terminology box (TBox). The basic propositionally closed DL is $\mathcal{ALC}$ in which the letters $\mathcal{AL}$ stand for attributive language and the letter $\mathcal{C}$ for complement (negation of arbitrary concepts). Besides $\mathcal{ALC}$, other letters are used to indicate various DL extensions. More pecisely, $\mathcal{S}$ is often used for $\mathcal{ALC}$ extended with transitive roles (R^+), $\mathcal{H}$ for role hierarchies, $\mathcal{O}$ for nominals, $\mathcal{I}$ for inverse roles, $\mathcal{N}$ for number restrictions, $\mathcal{Q}$ for qualified number restrictions, and $\mathcal{F}$ for functional properties. OWL[1] has three increasingly expressive sublanguages: OWL Lite, OWL DL, and OWL Full. If we omit the annotation properties of OWL, the OWL-Lite sublanguage is a syntactic variant of the Description Logic $\mathcal{SHIF}(\mathcal{D})$ where (D) means data values or data types, while OWL-DL is almost equivalent to the $\mathcal{SHOIN}(\mathcal{D})$ DL [13]. OWL-Full is the union of OWL syntax and RDF, and known to be undecidable mainly because it does not impose restrictions on the use of transitive properties. Accordingly, an OWL-Lite ontology corresponds

[1] In the following, OWL refers to OWL 1. Similar sublanguages exist for OWL 2.

to a $\mathcal{SHIF}(\mathcal{D})$ knowledge base, and an OWL-DL ontology corresponds to a $\mathcal{SHOIN}(\mathcal{D})$ knowledge base.

Although standard DLs provide considerable expressive power, they are limited to dealing with crisp, well-defined concepts and roles, and cannot express vague or uncertain knowledge. To overcome this deficiency, considerable work has been carried out in integrating uncertain knowledge into DLs in the last decade. One important theory for such integration is Fuzzy Sets and Fuzzy Logic. Yen [33] is the first who combines fuzzy logic with term subsumption languages and proposes a fuzzy extension to a very restricted DL, called $\mathcal{FTSL}^-$. The corresponding standard DL $\mathcal{FL}^-$, as defined in [5], is actually a sublanguage of $\mathcal{ALC}$, and only allows primitive concepts, primitive roles, defined concepts formed from concept intersection, value restriction and existential quantification. Semantically, the `min` function is used to interpret the intersection between two $\mathcal{FTSL}^-$ concepts. The knowledge base in $\mathcal{FTSL}^-$ includes only fuzzy terminological knowledge in the form of $C \sqsubseteq D$, where C and D are two fuzzy concepts. The inference problem Yen is interested in is testing subsumption relationships between fuzzy concepts. A concept D subsumes a concept C if and only if D is a fuzzy superset of C, i.e., given two concepts C, D defined in the fuzzy DL, $C \sqsubseteq D$ is viewed as $\forall x.C(x) \leq D(x)$. Thus, the subsumption relationship itself is a crisp Yes/No test. A structural subsumption algorithm is given in his work. Tresp and Molitor [32] consider a more general extension of $\mathcal{ALC}$ to many-valued logics, called $\mathcal{ALC}_{\mathcal{FM}}$. The language $\mathcal{ALC}_{\mathcal{FM}}$ allows constructors including conjunction, disjunction, manipulator, value restriction, and existential qualification in the definition of complex concepts. They define the semantics of a value restriction differently from Yen's work. This work also starts addressing the issue of a fuzzy semantics of modifiers M, such as *mostly*, *more or less*, and *very*, which are unary operators that can be applied to concepts. An example is *(very)TallPerson(John)*, which means that "John is a very tall person". In both of the work by [33] and [32], knowledge bases include only fuzzy terminological knowledge. But different from Yen's work, Tresp and Molitor are interested in determining fuzzy subsumption between fuzzy concepts, i.e., given concepts C, D, they want to know to which degree C is a subset of D. Such a problem is reduced to the problem of determining an adequate evaluation for an extended ABox which corresponds to a solution for a system of inequations. The degree of subsumption between concepts is then determined as the minimum of all values obtained for some specific variable. [32] presents a sound and complete reasoning algorithm for $\mathcal{ALC}_{\mathcal{FM}}$ which basically is an extension of each completion rule in the classical tableau algorithm for standard ALC. Another fuzzy extension of $\mathcal{ALC}$ is due to [30]. In this work, the interpretation of the Boolean operators and the quantifiers is based on the `min` and `max` functions, and the knowledge base includes both fuzzy terminological and fuzzy assertional knowledge. That is, the ABox assertions are equipped with a degree from [0,1]. Thus in this context, one may also want to find out to which degree other assertions follow from the ABox, which is called a *fuzzy* *entailment problem*. A decision algorithm for such fuzzy entailment problems in this fuzzy extension of $\mathcal{ALC}$ is presented. Similar to Yen, [30] is interested in crisp subsumption of fuzzy concepts, with the result being a crisp Yes or No, instead of a fuzzy subsumption relationship. Although [31] addresses the syntax and semantics for more expressive fuzzy DLs, no reasoning algorithm for the fuzzy subsumption between fuzzy concepts is given in his work. [25] consider modifiers in a fuzzy extension of the Description Logic $\mathcal{ALCQ}$, but the knowledge base in their work only consists of the TBox. They also present an algorithm which calculates the satisfiability interval for a fuzzy concept in fuzzy $\mathcal{ALCQ}$. The recent work in [29] presents an expressive fuzzy DL language with the underlying standard DL $\mathcal{SHIN}$. As we will explain in the following section, Fuzzy Logic is in fact a family of multi-valued logics derived from Fuzzy Set Theory. Identified by the specific fuzzy operations applied in the logic, the Fuzzy Logic family consists of Zadeh Logic, Product Logic, Gödel Logic, and more. Generally speaking, all existing work uses the basic fuzzy logic known as Zadeh Logic. Surprisingly enough, little work uses other logics with the exception of [4], which considers concrete domains and provide an algorithm for fuzzy $\mathcal{ALC}(\mathcal{D})$ under product semantics, and the work by H*á*jek [8, 9], which considers a fuzzy DL under arbitrary t-norms with $\mathcal{ALC}$ as the underlying DL language.

In this paper, in order to extend standard DLs with Fuzzy Logic in a broad sense, we propose a generalized form of norm-parameterized Fuzzy Description Logics. The main contributions of this paper can be explained as follows. First, unlike other approaches except [31], which only deal with crisp subsumption of fuzzy concepts, our Fuzzy Description Logic deals with fuzzy subsumption of fuzzy concepts and addresses its semantics. We argue that fuzzy subsumption of fuzzy concepts permits more adequate modeling of the uncertain knowledge existing in real world applications. Second, almost all of the existing work employs a single set of fuzzy operations, which limits their applicability in various real-world system and knowledge requirements. We propose a set of t-norms and s-norms in the semantics of our norm-parameterized Fuzzy Description Logics, so that the interpretation of complex concept descriptions can cover different logics in the Fuzzy Logic family, such as Zadeh Logic, Lukasiewicz Logic, Product Logic, Gödel Logic, and Yager Logic. Most importantly, Product Logic interprets fuzzy intersection as the inner product of the truth degrees between fuzzy concepts and fuzzy union as the product-sum operation. It thus broadens Fuzzy Logic and sets up a connection between Fuzzy Logic in the narrow sense and Probability Theory [22]. Third, the notion of fuzzy subsumption was first proposed in [31] and used in the forms $\geq n$ and $\leq n$, where n $\in$ [0,1], but it was left unsolved how to do reasoning on fuzzy knowledge bases. In this paper, we define a Fuzzy Description Logic with a unified uncertainty intervals of the form $[l, u]$, where $l, u \in$ [0,1] and $l \leq u$, and present its reasoning procedure.

Besides the work based on Fuzzy Sets and Fuzzy Logic, there is also some work based on other approaches. Probabilistic Description Logics [16][17][19] are built on Probability Theory; [11, 27] follows Possibility Theory; and [15] proposes a framework called $\mathcal{ALC}u$, which extends the standard DL $\mathcal{ALC}$ with a combination of different uncertainties. Due to space limitation, we point readers to our work in [35] for an in-depth review of various uncertainty extensions to Description Logics.

The current paper presents the whole methodology of our proposed Fuzzy Description Logic, including the underlying Fuzzy Logics, the syntax, the semantics, the knowledge bases, the reasoning procedure, and its decidability. The rest of the paper is organized as follows. Section 2 briefly introduces the syntax and semantics of the standard Description Logic $\mathcal{ALCN}$. Section 3 reviews Fuzzy Logic and Fuzzy Set Theory. Section 4 presents the syntax and semantics of the expressive Fuzzy Description Logic $f\mathcal{ALCN}$, as well as the components of knowledge bases using this knowledge representation formalism. Section 5 explains different reasoning tasks on the knowledge base. Section 6 addresses General Concept Inclusion (GCI) axioms, the $f\mathcal{ALCN}$ concept Negation Normal Form (NNF), and the ABox augmentation. In Section 7, we put forward the reasoning procedure for the consistency checking problem of an $f\mathcal{ALCN}$ knowledge base and illustrate fuzzy completion rules. Section 8 proves the decidability of the reasoning procedure by addressing its soundness, completeness, and termination. Finally, in the last section, we summarize our main results and give an outlook on future research .

2 Preliminaries

We briefly introduce Description Logics (DLs) in the current section. Their syntax and semantics in terms of classical First Order Logic are also presented. As a notational convention, we will use a, b, x for individuals, A for an atomic concept, C and D for concept descriptions, R for atomic roles.

Concept descriptions in $\mathcal{ALCN}$ are of the form:

$C \rightarrow \top|\bot|A|\neg A|\neg C|C \sqcap D|C \sqcup D|\exists R.C|\forall R.C| \geq nR| \leq nR$

The special concept names $\top$ (*top*) and $\bot$ (*bottom*) represent the most general and least general concepts, respectively.

DLs have a model theoretic semantics, which is defined by interpreting concepts as sets of individuals and roles as sets of pairs of individuals. An interpretation I is a pair $I = (\Delta^I, \cdot^I)$ consisting of a domain Δ^I which is a non-empty set and an interpretation function $\cdot^I$ which maps each individual x into an element of Δ^I ($x^I \in \Delta^I$), each concept C into a subset of Δ^I ($C^I \subseteq \Delta^I$) and each atomic role R into a subset of $\Delta^I \times \Delta^I$ ($R^I \subseteq \Delta^I \times \Delta^I$). The semantic interpretations of complex concept descriptions are shown in Table 1. In the at-most restriction and the at-least restriction, $\sharp\{\cdot\}$ denotes the cardinality of a set.

A knowledge base in DLs consists of two parts: the terminological box (TBox $\mathcal{T}$) and the assertional box (ABox $\mathcal{A}$). There are two kinds of assertions

Table 1 Syntax and Semantics of $\mathcal{ALCN}$ constructors

DL Constructor	DL Syntax	Semantics
top concept	$\top$	$\triangle^I$
bottom concept	$\perp$	$\emptyset$
atomic concept	A	$A^I \subseteq \triangle^I$
concept	C	$C^I \subseteq \triangle^I$
atomic negation	$\neg A$	$\triangle^I \setminus A^I$
concept negation	$\neg C$	$\triangle^I \setminus C^I$
concept conjunction	$C \sqcap D$	$C^I \cap D^I$
concept disjunction	$C \sqcup D$	$C^I \cup D^I$
exists restriction	$\exists R.C$	$\{x \in \triangle^I \vert \exists y. < x, y >\in R^I \wedge y \in C^I\}$
value restriction	$\forall R.C$	$\{x \in \triangle^I \vert \forall y. < x, y >\in R^I \rightarrow y \in C^I\}$
at-most restriction	$\leq nR$	$\{x \in \triangle^I \vert \sharp\{y \in \triangle^I \vert R^I(x, y)\} \leq n\}$
at-least restriction	$\geq nR$	$\{x \in \triangle^I \vert \sharp\{y \in \triangle^I \vert R^I(x, y)\} \geq n\}$

in the ABox of a DL knowledge bas: concept individual and role individual. A concept individual assertion has the form $C(a)$ while a role individual assertion is $R(a, b)$. The semantics of assertions is interpreted as the assertion $C(a)$ (resp. $R(a, b)$) is satisfied by I iff $a \in C^I$ (resp. $(a, b) \in R^I$).

The TBox of a DL knowledge base has several kinds of axioms. A concept inclusion axiom is an expression of subsumption of the form $C \sqsubseteq D$. The semantics of a concept inclusion axiom is interpreted as the axiom is satisfied by I iff $\forall x \in \Delta^I, x \in C^I \rightarrow x \in D^I$. A concept equivalence axiom is an expression of the form $C \equiv D$. Its semantics is that the axiom is satisfied by I iff $\forall x \in \Delta^I, x \in C^I \rightarrow x \in D^I \textit{ and } x \in D^I \rightarrow x \in C^I$.

3 Fuzzy Set Theory and Fuzzy Logic

Fuzzy Set Theory was first introduced by Zadeh [34] as an extension to the classical notion of a set to capture the inherent vagueness (the lack of crisp boundaries of sets). Fuzzy Logic is a form of multi-valued logic derived from Fuzzy Set Theory to deal with reasoning that is approximate rather than precise. Just as in Fuzzy Set Theory the set membership values can range between 0 and 1, in Fuzzy Logic the degree of truth of a statement can range between 0 and 1 and is not constrained to the two truth values $\{false, true\}$ as in classical predicate logic [24]. Formally, a fuzzy set X is characterized by a membership function $\mu(x)$ which assigns a value in the real unit interval [0,1] to each element $x \in X$, mathematically notated as $\mu : X \rightarrow [0, 1]$. $\mu(x)$ gives us a degree of an element x belonging to a set X. Such degrees can be computed based on some specific membership functions which can be linear or non-linear. Figure 1 shows the most general form of linear membership functions, also known as a trapezoidal membership function. Formally, we define it as *trapezoidal(a,b,c,d)* with the range of the membership function being $[k_1, k_2]$. Other linear membership functions can be regarded as its special forms. Specifically, if a=b and c=d, it defines a crisp membership function, *crisp(a,c)*. If a=b=0, it defines a left-shoulder membership function *leftshoulder(c,d)*. Similarly, it defines a right-shoulder membership function *rightshoulder(a,b)* when c=d=∞. It defines a triangular membership function *triangular(a,b,d)* if b=c.

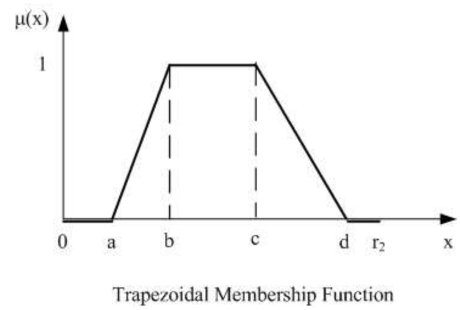

Fig. 1 Fuzzy Membership Function

For example, as shown in Figure 2, the fuzzy set *Young* is defined by a left_shoulder membership function *leftshoulder(30,50)*, while the fuzzy set *Old* is defined by a right_shoulder membership function *rightshoulder(40,70)*. Assume we know, John is 45 years old. Therefore, we have $Young(John) = 0.25$ which means the statement "John is young" has a truth value of 0.25. We also have $Old(John) = 0.17$ which means the statement "John is old" has a truth value of 0.17. But more often, we want to make vaguer statements, saying that "John is old" has a truth value of greater than or equal to 0.17. Such a statement can be written as $Old(John) \geq 0.17$. Another kind of often-used statement is less than or equal to, e.g., the truth degree for "John is young" is less than or equal to 0.25 ($Young(John) \leq 0.25$). In order to describe all of the above statements in a uniform form, we employ an interval syntax $[l, u]$ with $0 \leq l \leq u \leq 1$. Then $Young(John) = 0.25$ can be written as $Young(John)$ $[0, 0.25]$, $Young(John) \leq 0.25$ as $Young(John)$ $(0, 0.25]$, $Old(John) \geq 0.17$ as $Old(John)$ $[0.17, 1]$.

A fuzzy relation R over two fuzzy sets X_1 and X_2 is defined by a function $R : X_1 \times X_2 \rightarrow [0, 1]$. For example, the statement that "John drives 150" has a truth value of greater than or equal to 0.6 can be defined as $drives(John, 150)$ $[0.6, 1]$.

In the context of fuzzy sets and fuzzy relations, Fuzzy Logic extends the Boolean operations complement, union, and intersection defined on crisp sets and relations. The fuzzy operations in Fuzzy Logic are interpreted as mathematical functions over the unit interval [0,1]. The mathematical functions for fuzzy intersection are usually called t-norms ($t(x, y)$) and those for fuzzy union are called s-norms ($s(x, y)$). All mathematical functions satisfying cer-

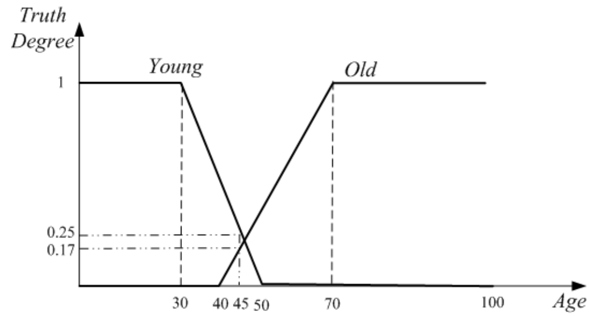

Fig. 2 A membership function for the concept Young

tain mathematical properties can serve as t-norms and s-norms. For example, in particular, a binary operation t(x,y) on the interval [0,1] is a t-norm if it is commutative, associative, non-decreasing and 1 is its unit element. The most well-known fuzzy operations for t-norms and s-norms include Zadeh Logic, Lukasiewicz Logic, Product Logic, Gödel Logic, and Yager Logic, as summarized in Table 2[2].

Table 2 Fuzzy Operations

	Zadeh Logic	Lukasiewicz Logic	Product Logic	Gödel Logic	Yager Logic
t-norm ($t(x,y)$)	$\min(x,y)$	$\max(x+y-1,0)$	$x \cdot y$	$\min(x,y)$	$min(1,(x^w+y^w)^{\frac{1}{w}})$
s-norm ($s(x,y)$)	$\max(x,y)$	$\min(x+y,1)$	$x+y-x \cdot y$	$\max(x,y)$	$1-min(1,((1-x)^w+(1-y)^w)^{\frac{1}{w}})$

Mathematical functions for fuzzy negation ($\neg x$) have to be non-increasing, and assign 0 to 1 and vice versa. There are at least two fuzzy complement operations satisfying the requirements. One is Lukasiewicz negation ($\neg x = 1 - x$) and the other is Gödel negation ($\neg x = \begin{cases} 1 & if\ x = 0 \\ 0 & else \end{cases}$).

Fuzzy implication (x⇒y) is also of fundamental importance for Fuzzy Logic but is sometimes disregarded, as we can use a straightforward way to define implication from fuzzy union and fuzzy negation using the corresponding tautology of classical logic; such implications are called S-implications (also known as the Kleene-Dienes implication). That is, denoting the s-norm as s(x,y) and the negation as $\neg x$, we have $x \Rightarrow y \equiv s(\neg x, y)$.

Another way to define implication is R-implications [10]: an R-implication is defined as a residuum of a t-norm; denoting the t-norm t(x,y) and the

[2] w satisfies $0 < w < \infty$, $w = 2$ is mostly used.

residuum $\Rightarrow$, we have $x \Rightarrow y \equiv \max\{z | t(x, z) \leq y\}$. This is well-defined only if the t-norm is left-continuous.

4 Fuzzy Description Logic

In this section, we present the syntax and semantics of $f\mathcal{ALCN}$, as well as $f\mathcal{ALCN}$ knowledge bases.

4.1 Syntax of $f\mathcal{ALCN}$

Definition 1 *Let $\mathbf{N_C}$ be a set of concept names. Let $\mathbf{R}$ be a set of role names. The set of $f\mathcal{ALCN}$ roles only consists of atomic roles. The set of $f\mathcal{ALCN}$ concepts is the smallest set such that*

1. *every concept name is a concept and*
2. *if C and D are $f\mathcal{ALCN}$ concepts and R is a $f\mathcal{ALCN}$ role, then $(\neg C)$, $(C \sqcap D)$, $(C \sqcup D)$, $(\exists R.C)$, $(\forall R.C)$, $(\geq R)$, and $(\leq R)$ are concepts.*

We can see that the syntax of this Fuzzy Description Logic is identical to that of the standard Description Logic $\mathcal{ALCN}$. But in $f\mathcal{ALCN}$, the concepts and roles are defined as fuzzy concepts (i.e. fuzzy sets) and fuzzy roles (i.e. fuzzy relations), respectively. A fuzzy concept here can be either a *primitive concept* defined by a membership function, or a *defined concept* constructed using the above fuzzy concept constructors.

4.2 Semantics of $f\mathcal{ALCN}$

Similar to standard DL, the semantics of $f\mathcal{ALCN}$ is based on the notion of interpretation. Classical interpretations are extended to the notion of fuzzy interpretations by using membership functions that range over the interval [0,1].

Definition 2 *A **fuzzy interpretation** I is a pair $I = (\Delta^I, \cdot^I)$ consisting of a domain Δ^I, which is a non-empty set, and a fuzzy interpretation function $\cdot^I$, which maps each individual x into an element of Δ^I ($x \in \Delta^I$), each concept C into a membership function $C^I : \Delta^I \to [0, 1]$, and each atomic role R into a membership function $R^I : \Delta^I \times \Delta^I \to [0, 1]$.*

Next we define the semantics of $f\mathcal{ALCN}$ constructors, including the top concept ($\top$), the bottom concept ($\bot$), concept negation ($\neg$), concept conjunction ($\sqcap$), concept disjunction ($\sqcup$), exists restriction ($\exists$), value restriction ($\forall$), and number restrictions ($\leq, \geq$). We explain how to apply the Fuzzy Logic operations of Table 2 to $f\mathcal{ALCN}$ with some examples.

The semantics of the top concept $\top$ is the greatest element in the domain Δ^I, that is, $\top^I = 1$. Note that, in standard DL, the top concept $\top \equiv A \sqcup \neg A$, while in $f\mathcal{ALCN}$, such an equivalence is not always true. After applying the s-norms $s(x, \neg x)$ of different logics in Table 2 on $A \sqcup \neg A$, only the result in Lukasiewicz Logic is still equal to 1. Thus, in $f\mathcal{ALCN}$, we have to explicitly define the top concept, stating that the truth degree of x in $\top$ is 1. Similarly, the bottom concept $\perp$ is the least element in the domain, defined as $\perp^I = 0$.

Concept negation (also known as concept complement) $\neg C$ is interpreted with a mathematical function which satisfies

1. $\neg\top^I(x) = 0, \neg\perp^I(x) = 1$.
2. self-inverse, i.e., $(\neg\neg C)^I(x) = C^I(x)$.

As we have discussed in Section 3, both Lukasiewicz negation and Gödel negation satisfy these properties. In our approach, we adopt Lukasiewicz negation ($\neg x = 1 - x$) as it reflects human being's intuitively understanding of the meaning of concept negation. For example, if we have that the statement "John is young" has a truth value of greater than or equal to 0.8 ($Young(John)$ $[0.8, 1]$), and after applying Lukasiewicz negation operator to the statement "John is not young", we have $\neg Young(John) = \neg[0.8, 1] = [0, 0.2]$.

The interpretation of concept conjunction (also called concept intersection) is defined by a t-norm as $(C \sqcap D)^I(x) = t(C^I(x), D^I(x))$ $(\forall x \in \Delta^I)$, that is, under the fuzzy interpretation I, the truth degree of x being an element of the concept $(C \sqcap D)$ is equal to the result of applying a t-norm function on the truth degree of x being an element of the concept C and the truth degree of x being an element of the concept D. For example, if we have $Young(John)$ $[0.8, 1]$ and $Tall(John)$ $[0.7, 1]$, and assume the minimum function in Zadeh Logic or Gödel Logic is chosen as the t-norm, then the truth degree that John is both young and tall is

$(Young \sqcap Tall)(John) = t_Z([0.8, 1], [0.7, 1]) = [0.7, 1]$.

If the multiplication function in Product Logic is chosen as the t-norm, then the degree of truth that John is both young and tall is

$(Young \sqcap Tall)(John) = t_P([0.8, 1], [0.7, 1]) = [0.56, 1]$.

Under Lukasiewicz Logic, the truth degree is $t_L([0.8, 1], [0.7, 1]) = [0.5, 1]$, while under Yager Logic with $w = 2$, the truth degree is $t_Y([0.8, 1], [0.7, 1]) = [1, 1]$ as $\min((0.8^2 + 0.7^2)^{1/2}, 1) = 1$ and $\min((1^2 + 1^2)^1/2, 1) = 1$.

The interpretation of concept disjunction (union) is defined by a s-norm as $(C \sqcup D)^I(x) = s(C^I(x), D^I(x))$ $(\forall x \in \Delta^I)$, that is, under the fuzzy interpretation I, the truth degree of x being an element of the concept $(C \sqcup D)$ is equal to the result of applying an s-norm function on the truth degree of x being an element of the concept C and the truth degree of x being an element of the concept D.

For example, if we have $Young(John)$ $[0.8, 1]$ and $Tall(John)$ $[0.7, 1]$, then under Zadeh Logic, the degree of truth that John is young or tall is $(Young \sqcup Tall)(John) = s_Z([0.8, 1], [0.7, 1]) = [0.8, 1]$. Under Product Logic, it is $(Young \sqcup Tall)(John) = s_P([0.8, 1], [0.7, 1]) = [0.94, 1]$. Under Lukasiewicz Logic, the truth degree is $S_L([0.8, 1], [0.7, 1]) = [1, 1]$, while under Yager Logic with $w = 2$, the truth degree is $s_Y([0.8, 1], [0.7, 1]) = [0.64, 1]$.

The semantics of exists restriction $\exists R.C$ is the result of viewing $\exists R.C$ as the open first order formula $\exists y.R(x, y) \wedge C(y)$ and the existential quantifier $\exists$ is viewed as a disjunction over the elements of the domain, defined as sup (supremum or least upper bound). Therefore, we define

$(\exists R.C)^I(x) = \sup_{y \in \Delta^I}\{t(R^I(x, y), C^I(y))\}$

Suppose in an ABox A_1, we have

$hasDisease(P001, Cancer)$ $[0.2, 1]$, $VitalDisease(Cancer)$ $[0.5, 1]$,

$hasDisease(P001, Cold)$ $[0.6, 1]$, $VitalDisease(Cold)$ $[0.1, 1]$.

Further we assume the minimum function in Zadeh Logic is chosen as the t-norm, then

$$\begin{aligned}(\exists R.C)^I(x) &= \sup\{t_Z(hasDisease(P001, Cancer), VitalDisease(Cancer)),\\ &\quad t_Z(hasDisease(P001, Cold), VitalDisease(Cold))\}\\ &= \sup\{t_Z([0.2, 1], [0.5, 1]), t_Z([0.6, 1], [0.1, 1])\}\\ &= \sup\{[0.2, 1], [0.1, 1]\} = [0.1, 1]\end{aligned}$$

That is, the truth degree for the complex concept assertion ($\exists$ *hasDisease.VitalDisease*) (*P001*) is greater than or equal to 0.1. Similarly, we can get the results under other logics.

A value restriction $\forall R.C$ is viewed as an implication of the form $\forall y \in \Delta^I, R^I(x, y) \to C^I(x)$. As explained in Section 3, both Kleene-Dienes implication and R-implication can be applied in the context of fuzzy logic. Following the semantics proposed by Hajek [7], we interpret $\forall$ as inf (infimum or greatest lower bound). Furthermore, in classical logic, $a \to b$ is a shorthand for $\neg a \vee b$; we can thus interpret $\to$ as the Kleene-Dienes implication and finally get its semantics as $(\forall R.C)^I(x) = \inf_{y \in \Delta^I}\{s(\neg R^I(x, y), C^I(y))\}$.

For example, with the ABox A_1 and Product Logic, we have

$$\begin{aligned}(\forall R.C)^I(x) &= \inf\{s_P(\neg hasDisease(P001, Cancer), VitalDisease(Cancer)),\\ &\quad s_P(\neg hasDisease(P001, Cold), VitalDisease(Cold))\}\\ &= \inf\{s_P([0, 0.8], [0.5, 1]), s_P([0, 0.4], [0.1, 1])\}\\ &= \inf\{[0.5, 1], [0.1, 1]\} = [0.5, 1]\end{aligned}$$

Similarly, we can get the results under other logics.

A fuzzy at-least restriction is of the form $\geq nR$, whose semantics

$(\geq nR)^I(x) = \sup_{y_1, \ldots, y_n \in \Delta^I, y_i \neq y_j, 1 \leq i < j \leq n} t_{i=1}^n\{R^I(x, y_i)\}$

is derived from its first order reformulation

$\exists y_1, \ldots, y_n. \wedge_{i=1}^n R(x, y_i) \bigwedge \wedge_{1 \leq i < j \leq n} y_i \neq y_j$.

The semantics states that there are at least n distinct individuals $(y_1, \ldots, y_n)$ all of which satisfy $R_I(x, y_i)$ to some given degree.

Furthermore, we define the semantics of a fuzzy at-most restriction as

$$(\leq nR)^I(x) = \neg(\geq (n+1)R)^I(x)$$
$$= \neg \sup_{y_1,\ldots,y_{n+1}\in\Delta^I, y_i\neq y_j, 1\leq i<j\leq n+1} t_{i=1}^{n+1}\{R^I(x,y_i)\}$$
$$= \inf_{y_1,\ldots,y_{n+1}\in\Delta^I, y_i\neq y_j, 1\leq i<j\leq n+1} s_{i=1}^{n+1}\{\neg R^I(x,y_i)\}$$

The semantics states that for $n+1$ role assertions $R_I(x,y_i)$ $(1 \leq i \leq n+1)$ that can be formed, at least one satisfies $\neg R_I(x,y_i)$ to some given degree.

Note that, the semantics of fuzzy at-least restriction (respectively, fuzzy at-most restriction) in [31] and [29] is a special case of our fuzzy at-least restriction (respectively, fuzzy at-most restriction).

An alternative view of the semantics of a fuzzy at-most number restriction is that there are at most n unique individuals $(y_1,\ldots,y_n)$ that satisfy $R_I(x,y_i)$ to some given degree. For example, $\leq 2R$ [0.8,1] means that there are at most two role instance assertions about any individual a: $R(a,b_1)$ and $R(a,b_2)$. Moreover, assuming $x_{R(a,b_1)}$ is the truth degree of $R(a,b_1)$ and $x_{R(a,b_2)}$ is the truth degree of $R(a,b_2)$, both $x_{R(a,b_1)}$=[0.8,1] and $x_{R(a,b_2)}$=[0.8,1] hold, but $b_1 \neq b_2$ does not necessarily hold. Furthermore, for $\leq nR$ $[l,u]$, if we find there are $n+1$ assertions satisfying the truth degree constraints, we have to find some individuals that can be merged, similar to the case in standard DL [1]. If we find there are more than n different individuals after merging, we say the concept $\leq 2R$ [0.8,1] is unsatisfiable. It is easy to see that these fuzzy semantics generalize the crisp case of standard DL where the truth degree of all assertions is [1,1].

The semantics of the complex concept descriptions axioms for $f\mathcal{ALCN}$ are summarized in Table 3.

Table 3 Syntax and Semantics of $f\mathcal{ALCN}$ constructors

Constructor	Syntax	Semantics
top concept	$\top$	$\top^I = 1$
bottom concept	$\bot$	$\bot^I = 0$
atomic negation	$\neg A$	$(\neg A)^I(x) = \neg A^I(x)$
atomic negation	$\neg C$	$(\neg C)^I(x) = \neg C^I(x)$
concept conjunction	$C \sqcap D$	$(C \sqcap D)^I = t(C^I(x), D^I(x))$
concept disjunction	$C \sqcup D$	$(C \sqcup D)^I = s(C^I(x), D^I(x))$
exists restriction disjunction	$\exists R.C$	$(\exists R.C)^I(x) = \sup_{y\in\Delta^I}\{t(R^I(x,y), C^I(y))\}$
value restriction	$\forall R.C$	$(\forall R.C)^I(x) = \inf_{y\in\Delta^I}\{s(\neg R^I(x,y), C^I(y))\}$
at-least restriction	$\geq nR$	$(\geq nR)^I(x) = \sup_{y_1,\ldots,y_n\in\Delta^I, y_i\neq y_j, 1\leq i<j\leq n} t_{i=1}^{n}\{R^I(x,y_i)\}$
at-most restriction	$\leq nR$	$(\leq nR)^I x \equiv \neg(\geq (n+1)R)^I(x)$

4.3 Knowledge Bases in $f\mathcal{ALCN}$

A fuzzy knowledge base in $f\mathcal{ALCN}$ consists of two parts: the fuzzy terminological box (TBox $\mathcal{T}$) and the fuzzy assertional box (ABox $\mathcal{A}$). The TBox contains several kinds of axioms. A fuzzy concept inclusion axiom has the form of $C \sqsubseteq D\ [l,u]$ with $0 \leq l \leq u \leq 1$, which describes that the subsumption degree of truth between concepts C and D is from l to u.

For example, the axiom

$Professor \sqsubseteq (\exists publishes.Journalpaper \sqcap \exists teaches.Graduatecourse)\ [0.8, 1]$ states that the concept *professor* is subsumed by entities that *publish* journal papers and *teach* graduate courses with a truth degree of at least 0.8.

In order to simplify the reasoning task of subsumption checking, some work on DL restricts a terminological box to introduction axioms of the form $A \sqsubseteq C$ where A is an atomic concept and C is a concept description. In this research, we permit general concept inclusion axioms (GCIs). The FOL translation of a general concept inclusion axiom $C \sqsubseteq D$ has the form $\forall x.C(x) \to D(x)$; therefore, its semantics is defined as

$(C \sqsubseteq D)^I(x) = \inf_{x \in \Delta^I} C^I(x) \to D^I(x) = \inf_{x \in \Delta^I}\{s(\neg C^I(x), D^I(x))\}$.

That is, for a fuzzy interpretation I, I satisfies $C \sqsubseteq D\ [l,u]$ iff $\forall x \in \Delta^I, l \leq \inf_{x \in \Delta^I}\{s(\neg C^I(x), D^I(x))\} \leq u$. For the above example, it means under every fuzzy interpretation I of the knowledge base, we have

$0.8 \leq s(x_{\neg Professor}, x_{(\exists publishes.Journalpaper \sqcap \exists teaches.Graduatecourse)}) \leq 1$.

We also permit a fuzzy concept equivalence axioms of the form $C \equiv D$ with the semantics $C^I = D^I$.

There are three kinds of assertions in the ABox: concept individual, role individual, and individual inequality. A fuzzy concept assertion and a fuzzy role assertion are of the form $C(a)\ [l,u]$ and $R(a,b)\ [l,u]$, respectively. An individual inequality in the $f\mathcal{ALCN}$ ABox is identical to standard DLs and has the form $a \neq b$ for a pair of individuals a and b. Given a fuzzy Interpretation I, I satisfies $C(a)\ [l,u]$ iff $l \leq C^I(a) \leq u$, I satisfies $R(a,b)\ [l,u]$ iff $l \leq R^I(a,b) \leq u$, and I satisfies $a \neq b$ iff $a \neq b$ under I.

5 Reasoning Tasks

We are interested in several reasoning tasks for an $f\mathcal{ALCN}$ knowledge base. First, **consistency checking** refers to the reasoning task of determining whether the knowledge base is consistent.

In order to define what it means for a knowledge base to be consistent, we first explain the general idea of how the $f\mathcal{ALCN}$ reasoning procedure works and then give several formal definitions. The reasoning procedure derives new assertions through applying completion rules (cf. Table 4) in an arbitrary order, adds derived assertions to an extended ABox A_i^ε, and at the same time adds corresponding constraints that incorporate the semantics of the assertions to a constraint set $\mathcal{C}_j$ in the form of (linear or non-linear) inequations. The reasoning procedure stops when either A_i^ε contains a clash or no further rule can be applied to A_i^ε.

Definition 3 *Let A_i^ε be an extended ABox obtained by applications of the completion rules. Let $\mathcal{C}_i$ be a constraint set obtained by applications of the completion rules. An extended ABox A_i^ε is called **complete** if no more completion rule can be applied to A_i^ε.*

Definition 4 *For a constraint set $\mathcal{C}_i$ with respect to a complete ABox A_i^ε, let $Var(\mathcal{C}_i)$ be the set of variables occurring in $\mathcal{C}_i$. If the system of inequations in $\mathcal{C}_i$ is solvable, the result of the constraint set, i.e., the mapping $\Phi : Var(\mathcal{C}) \rightarrow [0,1]$, is called a **solution**.*

Definition 5 *We say there is a **clash** in the extended ABox A_i^ε iff one of the following situations occurs:*

1. $\{\bot(a)\ [l,u]\} \subseteq A_i^\varepsilon$ *and* $0 < l \leq u$
2. $\{\top(a)\ [l,u]\} \subseteq A_i^\varepsilon$ *and* $l \leq u < 1$
3. $\{A(a)\ [l_1,u_1], A(a)\ [l_2,u_2]\} \subseteq A_i^\varepsilon$ *and* $(u_1 < l_2$ *or* $u_2 < l_1)$
4. $\{(\leq nR)(a)\ [l,u]\} \cup \{R(a,b_i)\ [l_i,u_i] | 1 \leq i \leq n+1\} \cup \{b_i \neq b_j | 1 \leq i < j \leq n+1\} \subseteq A_i^\varepsilon$ *and* $\{[l_i,u_i] \subseteq [l,u] | 1 \leq i \leq n+1\}$

For example, if a knowledge base contains both assertions *Tall(John)* [0,0.2] and *Tall(John)* [0.7,1], since 0.2<0.7, the third clash will be triggered.

Note that we do not make the unique names assumption for individuals in the ABox. Since number restrictions may lead to the identification of different individual names, we therefore define explicit inequality assertions of the form: $b_i \neq b_j$ for two individuals b_i and b_j, as in clash 4.

The following is our definition of a model of an $f\mathcal{ALCN}$ knowledge base.

Definition 6 *Let A_i^ε be the extended ABox obtained by applications of the completion rules. Let $I = (\Delta^I, \cdot^I)$ be a fuzzy interpretation, $\Phi : Var(\mathcal{C}) \rightarrow [0,1]$ be a solution, x_α be the variable representing the truth degree of assertion α. For each concept assertion $C(a)$, $C^I(a) = \Phi(x_{C(a)})$. For each role assertion $R(a,b)$, $R^I(a,b) = \Phi(x_{R(a,b)})$. The pair $< I, \Phi >$ is a **model** of the extended ABox A_i^ε if the following properties hold:*

$\forall a, b \in \Delta^I$,

1. *if* $\{(\neg C)(a)\} \in A_i^\varepsilon$, *then* $\Phi(x_{(\neg C)(a)}) = 1 - \Phi(x_{C(a)})$,
2. *if* $\{(C \sqcap D)(a)\} \in A_i^\varepsilon$, *then* $\Phi(x_{(C \sqcap D)(a)}) = t(\Phi(x_{C(a)}), \Phi(x_{D(a)}))$,
3. *if* $\{(C \sqcup D)(a)\} \in A_i^\varepsilon$, *then* $\Phi(x_{(C \sqcup D)(a)}) = s(\Phi(x_{C(a)}), \Phi(x_{D(a)}))$,
4. *if* $\{(\exists R.C)(a)\} \in A_i^\varepsilon$, *then* $\Phi(x_{(\exists R.C)(a)}) = \sup_{b \in \Delta^I}\{t(\Phi(x_{R(a,b)}), \Phi(x_{C(a)}))\}$,
5. *if* $\{(\forall R.C)(a)\} \in A_i^\varepsilon$, *then* $\Phi(x_{(\forall R.C)(a)}) = \inf_{b \in \Delta^I}\{s(1 - \Phi(x_{R(a,b)}), \Phi(x_{C(a)}))\}$,
6. *if* $\{(\geq nR)(a)\} \in A_i^\varepsilon$, *then there are at least n distinct individuals* $(y_1, \ldots, y_n)$ *all of which satisfy* $\Phi(x_{R_I(x,y_i)}$, *and*
7. *if* $\{(\leq nR)(a)\} \in A_i^\varepsilon$, *then there are at most n distinct individuals* $(y_1, \ldots, y_n)$ *all of which satisfy* $\Phi(x_{R_I(x,y_i)}$.

Definition 7 *Let $KB =< T, A >$ be a $f\mathcal{ALCN}$ knowledge base where $\mathcal{T}$ is the fuzzy TBox and $\mathcal{A}$ is the fuzzy ABox. Let $I = (\Delta^I, \cdot^I)$ be a fuzzy interpretation, and $\Phi : Var(\mathcal{C}) \to [0, 1]$ be a solution. If there exists a model $< I, \Phi >$ for the extended ABox resulting from $KB =< T, A >$, we say the knowledge base $KB =< T, A >$ is* ***consistent****. If there is no such model, we call the knowledge base inconsistent.*

The second reasoning task is **instance checking**, which determines the degree to which an assertion is true. That is, let α be an assertion $C(a)$. We want to check whether $KB \models \alpha$ and to what degree the entailment is true. Such a problem can be reduced to the consistency problem. We first check whether $KB \cup \{\neg C(a)\ (0, 1]\}$ is consistent, and then solve the corresponding constraint set.

The third reasoning task is **subsumption checking**, which determines the subsumption degree between concepts C and D. That is, let α be an assertion $C \sqsubseteq D$. We want to check whether $KB \models \alpha$ and its truth degree. Such a problem can also be reduced to the consistency problem. We first check whether $KB \cup \{C \sqcap \neg D\ (0, 1]\}$ is consistent, and then solve the corresponding constraint set.

Another interesting reasoning task is the classification of $f\mathcal{ALCN}$ knowledge base, i.e. computing all fuzzy subsumptions between concepts in a given knowledge base. An intuitive way is to choose each pair of concepts and check their subsumption. Classification problem can thus be further reduced to consistency checking. There are some ways to optimize such a reasoning procedure for classification, but we do not consider this issue in this paper.

6 GCI, NNF, and ABox Augmentation

For the following note that, since GCI is the general form of concept subsumption, and concept equation is the general form of concept definition, we only use the general forms here. Let C and D be concept descriptions, R be roles, a and b be individuals. In this section we discuss how to deal with General Concept Inclusion (GCI) axioms, get the $f\mathcal{ALCN}$ concept Negation Normal Form (NNF), and augment the Extended ABox step by step.

First, every GCI axiom in the TBox is transformed into its normal form. That is, replace each axiom of the form $C \sqsubseteq D$ [l,u] with $\top \sqsubseteq \neg C \sqcup D$ [l,u]. As mentioned in the previous section, we consider GCIs in knowledge bases. In standard DL, we have the identity $C \sqsubseteq D \iff \perp \equiv C \sqcap \neg D$ [12]. Negating both sides of this equality gives $\top \equiv \neg C \sqcup D$. Accordingly, in fuzzy DL, we have $C \sqsubseteq D\ [l, u] \iff \top \sqsubseteq \neg C \sqcup D\ [l, u]$.

Second, transform every concept description (in the TBox and the ABox) into its Negation Normal Form (NNF). The NNF of a concept can be obtained by applying the following equivalence rules:

$\forall a, b \in \Delta^I$,

$$\neg\neg C\ [l,u] \equiv C\ [l,u]\ , \tag{1}$$

$$\neg(C \sqcup D)\ [l,u] \equiv \neg C \sqcap \neg D\ [l,u]\ , \tag{2}$$

$$\neg(C \sqcap D)\ [l,u] \equiv \neg C \sqcup \neg D\ [l,u]\ , \tag{3}$$

$$\neg\exists R.C\ [l,u] \equiv \forall R.\neg C\ [l,u]\ , \tag{4}$$

$$\neg\forall R.C\ [l,u] \equiv \exists R.\neg C\ [l,u]\ , \tag{5}$$

$$\neg(\leq nR)\ [l,u] \equiv \geq (n+1)R\ [l,u]\ , \tag{6}$$

$$\neg(\geq nR)\ [l,u] \equiv \begin{cases} \leq (n-1)R\ [l,u] & n > 1 \\ \bot & n = 1 \end{cases}\ , \tag{7}$$

Note that all of the above equivalence rules can be proved for the different logics in Table 2 as a consequence of choosing Lukasiewicz negation and Kleene-Dienes implication in explaining the semantics of $f\mathcal{ALCN}$ concepts. If we use R-implication and/or Gödel negation, such equivalence rules do not necessarily hold. Here we only show the equivalence of rule 2.

Proof: Besides applying Lukasiewicz negation, by applying the min t-norm and the max s-norm in Zadeh Logic, we have that the left side of rule 2 is equal to $1 - max(x, y)$ where x is the truth degree of $C(a)$ and y is the truth degree of $D(a)$, and the right side of rule 2 is equal to $min(1-x, 1-y)$. Since $1 - max(x, y) = min(1 - x, 1 - y)$, rule 2 holds under Zadeh Logic.

By applying the t-norm and the s-norm in Lukasiewicz Logic, we have that the left side of rule 2 is equal to $1 - min(x + y, 1)$ and the right side of rule 2 is equal to $max(1 - x - y, 0)$. Since $1 - min(x + y, 1) = max(1 - x - y, 0)$, rule 2 holds under Lukasiewicz Logic.

By applying the t-norm and the s-norm in Product Logic, we have that the left side of rule 2 is equal to $1 - (x + y - xy)$ and the right side of rule 2 is equal to $(1 - x)(1 - y)$. Since $1 - (x + y - xy) = (1 - x)(1 - y)$, rule 2 holds under Product Logic.

By applying the t-norm and the s-norm in Yager Logic with $w = 2$, we have that the left side of rule 2 is equal to $min(1, ((1 - x)^2 + (1 - y)^2)^{1/2})$ and the right side of rule 2 is also equal to $min(1, ((1 - x)^2 + (1 - y)^2)^{1/2})$. So rule 2 holds under Yager Logic.

When applying Lukasiewicz negation as the fuzzy complement operation, Zadeh Logic and Gödel Logic have the same t-norm and s-norm. So rule 2 holds under Gödel Logic.

Therefore, rule 2 holds under the different logics in Table 2.

$\square$

Third, augment the ABox $\mathcal{A}$ with respect to the TBox $\mathcal{T}$. This step is also called eliminating the TBox. That is, for each individual a in $\mathcal{A}$ and each axiom $\top \sqsubseteq \neg C \sqcup D$ [l,u] in $\mathcal{T}$, add $(\neg C \sqcup D)(a)$ [l,u] to $\mathcal{A}$. The resulting ABox after finishing this step is called the initial extended ABox, denoted by A_0^{ε}.

Now for a $f\mathcal{ALCN}$ $KB =< \mathcal{T}, \mathcal{A} >$, by following the above steps, we eliminate the TBox by augmenting the ABox and get an extended ABox $\mathcal{A}$ in which all concepts occurring are in NNF. Next, we initialize the set of constraints $\mathcal{C}_0$ and get ready for apply the reasoning procedure for the consistency checking problems. $\mathcal{C}_0$ is initialized as follows: For each concept assertion $\{C(a)\ [l,u]\} \in A_0^\varepsilon$, add $\{l \leq x_{C(a)} \leq u\}$ into $\mathcal{C}_0$; for each role assertion $\{R(a,b)\ [l,u]\} \in A_0^\varepsilon$, add $\{l \leq x_{R(a,b)} \leq u\}$ into $\mathcal{C}_0$.

Finally, the reasoning procedure continues to check whether A_i^ε contains a clash. If it detects clashes in all of the or-branches, it returns the result that the knowledge base is inconsistent. If the complete ABox does not contain an obvious clash, the reasoning procedure continues to solve the inequations in the constraint set $\mathcal{C}_j$. If the system of these inequations is unsolvable, the knowledge base is inconsistent, and consistent otherwise.

As for the other reasoning problems, including instance checking and subsumption checking, the values returned from the system of inequations, if solvable, serve as the truth degrees of these entailment problems.

7 Reasoning Procedure

The main component of the $f\mathcal{ALCN}$ reasoning procedure consists of a set of completion rules. Like standard DL tableau algorithms, the reasoning procedure for $f\mathcal{ALCN}$ consistency checking problem tries to prove the consistency of an extended ABox $\mathcal{A}$ by constructing a model of $\mathcal{A}$, which, in the context of Fuzzy Description Logic, is a fuzzy interpretation $I = (\triangle^I, \cdot^I)$ with respect to a solution Φ. Such a model has the shape of a forest, a collection of trees with nodes corresponding to individuals, root nodes corresponding to named individuals, and edges corresponding to roles between individuals. Each node is associated with a node label, $\mathcal{L}$(individual). But unlike in standard DL where a node is labeled only with concepts, each node in $f\mathcal{ALCN}$ is associated with a label that consists of a pair of elements $\langle concept, constraint\rangle$, to show the concept assertions for this individual and its corresponding constraints. Furthermore, each edge is associated with an edge label, $\mathcal{L}$(individual$_1$, individual$_2$) which consists of a pair of elements $\langle role, constraint\rangle$, instead of simply being labeled with roles as in standard DL.

Let λ be the constraint attached to an assertion α. The variable x_α denotes the truth degree of an assertion α. After getting the initial extended ABox A_0^ε and the initial set of constraints $\mathcal{C}_0$, the reasoning procedure expands the ABox and the constraint set by repeatedly applying the completion rules defined in Table 4. Such an expansion in the reasoning procedure is completed when (1) A_i^ε contains a clash or (2) none of the completion rules is applicable.

Table 4 Completion Rules of the Tableau Procedure

The concept negation rule **Condition**: $\mathcal{A}_i^\varepsilon$ contains $\neg A(a)\ \lambda$, but does not contain $A(a)\ \neg\lambda$. **Action**: If λ is not the variable $x_{\neg A(a)}$, $C_{j+1} = C_j \cup \{x_{\neg A(a)} = [l,u]\} \cup \{x_{A(a)} = [1-u, 1-l]\}$, $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_i^\varepsilon \cup \{A(a)\ [1-u, 1-l]\}$. Otherwise, $C_{j+1} = C_j \cup \{x_{\neg A(a)} = 1 - x_{A(a)}\}$, $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_i^\varepsilon \cup \{A(a)\ x_{A(a)}\}$
The concept conjunction rule **Condition**: $\mathcal{A}_i^\varepsilon$ contains $(C \sqcap D)(a)\ \lambda$, but $\mathcal{C}_j$ does not contain $t(x_{C(a)}, x_{D(a)}) = \lambda$. **Action**: $\mathcal{C}_{j+1} = \mathcal{C}_j \cup \{t(x_{C(a)}, x_{D(a)}) = \lambda\}$. If λ is not the variable $x_{(C \sqcap D)(a)}$, $C_{j+1} = C_{j+1} \cup \{x_{(C \sqcap D)(a)} = \lambda\}$. If $\mathcal{A}_i$ does not contain $C(a)\ x_{C(a)}$, $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_i^\varepsilon \cup \{C(a)\ x_{C(a)}\}$. If $\mathcal{A}_i$ does not contain $D(a)\ x_{D(a)}$, $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_i^\varepsilon \cup \{D(a)\ x_{D(a)}\}$.
The concept disjunction rule **Condition**: $\mathcal{A}_i^\varepsilon$ contains $(C \sqcup D)(a)\ \lambda$, but $\mathcal{C}_j$ does not contain $s(x_{C(a)}, x_{D(a)}) = \lambda$. **Action**: $\mathcal{C}_{j+1} = \mathcal{C}_j \cup \{s(x_{C(a)}, x_{D(a)}) = \lambda\}$. If λ is not the variable $x_{(C \sqcup D)(a)}$, $C_{j+1} = C_{j+1} \cup \{x_{(C \sqcup D)(a)} = \lambda\}$. If $\mathcal{A}_i$ does not contain $C(a)\ x_{C(a)}$, $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_i^\varepsilon \cup \{C(a)\ x_{C(a)}\}$. If $\mathcal{A}_i$ does not contain $D(a)\ x_{D(a)}$, $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_i^\varepsilon \cup \{D(a)\ x_{D(a)}\}$.
The exists restriction rule **Condition**: $\mathcal{A}_i^\varepsilon$ contains $(\exists R.C)(a)\ \lambda$, and a is not blocked. **Action**: If there is no individual name b such that $\mathcal{C}_j$ contains $t(x_{C(b)}, x_{R(a,b)}) = x_{(\exists R.C)(a)}$, then $\mathcal{C}_{j+1} = \mathcal{C}_j \cup \{t(x_{C(b)}, x_{R(a,b)}) = \lambda\}$. If $\mathcal{A}_i$ does not contain $C(b)\ x_{C(b)}$, $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_i^\varepsilon \cup \{C(b)\ x_{C(b)}\}$. If $\mathcal{A}_i$ does not contain $R(a,b)\ x_{R(a,b)}$, $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_i^\varepsilon \cup \{R(a,b)\ x_{R(a,b)}\}$. For each axiom $\top \sqsubseteq \neg C \sqcup D\ [l,u]$ in the TBox, add $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_{i+1}^\varepsilon \cup \{(\neg C \sqcup D)(b)\ [l,u]\}$. If λ is not the variable $x_{(\exists R.C)(a)}$, then if there exists $x_{(\exists R.C)(a)} = \lambda'$ in $\mathcal{C}_j$, then $\mathcal{C}_{j+1} = \mathcal{C}_{j+1} \backslash \{x_{(\exists R.C)(a)} = \lambda'\} \cup \{x_{(\exists R.C)(a)} = sup(\lambda, \lambda')\}$, else add $\mathcal{C}_{j+1} = \mathcal{C}_{j+1} \cup \{x_{(\exists R.C)(a)} = \lambda\}$.
The value restriction rule **Condition**: A_i^ε contains $(\forall R.C)(a)\ \lambda$ and $R(a,b)\ \lambda'$. **Action**: $A_{i+1}^\varepsilon = A_i^\varepsilon \cup \{C(b)\ x_{C(b)}\}$, $\mathcal{C}_{j+1} = \mathcal{C}_j \cup \{s(x_{C(b)}, x_{\neg R(a,b)}) = x_{(\forall R.C)(a)}\}$. If λ is not the variable $x_{(\forall R.C)(a)}$, then if there exists $x_{(\forall R.C)(a)} = \lambda''$ in $\mathcal{C}_j$, add $\mathcal{C}_{j+1} = \mathcal{C}_{j+1} \backslash \{x_{(\forall R.C)(a)} = \lambda''\} \cup \{x_{(\forall R.C)(a)} = inf(\lambda, \lambda'')\}$, otherwise, add $\mathcal{C}_{j+1} = \mathcal{C}_{j+1} \cup \{x_{(\forall R.C)(a)} = \lambda\}$.
The at-least rule **Condition**: $\mathcal{A}_i^\varepsilon$ contains $(\geq nR)(a)\ \lambda$, a is not blocked, and there are no individual names $b_1, \ldots, b_n$ such that $R(a, b_i)\ \lambda_i\ (1 \leq i \leq n)$ are contained in $\mathcal{A}_i^\varepsilon$. **Action**: $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_i^\varepsilon \cup \{R(a, b_i)\ \lambda \mid 1 \leq i \leq n\} \cup \{b_i \neq b_j \mid 1 \leq i \leq n\}$, $\mathcal{C}_{j+1} = \mathcal{C}_j \cup \{t(x_{R(a,b_1)}, \ldots, x_{R(a,b_n)}) = \lambda\}$.
The at-most rule **Condition**: $\mathcal{A}_i^\varepsilon$ contains $n+1$ distinguished individual names $b_1, \ldots, b_{n+1}$ such that $(\leq nR)(a)\ \lambda$, $R(a, b_i)\ \lambda_i\ (1 \leq i \leq n+1)$ are contained in $\mathcal{A}_i^\varepsilon$ and $b_i \neq b_j$ is not in $\mathcal{A}_i^\varepsilon$ for some $i \neq j$, and if λ is not the variable $x_{\leq nR(a)}$ and for any $i\ (1 \leq i \leq n+1)$, λ_i is not the variable $x_{R(a,b_i)}$, $\lambda_i \subseteq \lambda$ holds. **Action**: For each pair b_i, b_j such that $j > i$ and $b_i \neq b_j$ is not in $\mathcal{A}_i^\varepsilon$, the ABox $\mathcal{A}_{i+1}^\varepsilon$ is obtained from $\mathcal{A}_i^\varepsilon$ and the constraint set $\mathcal{C}_{j+1}$ is obtained from $\mathcal{C}_j$ by replacing each occurrence of b_j by b_i, and if λ_i is the variable $x_{R(a,b_i)}$, $\mathcal{C}_{j+1} = \mathcal{C}_{j+1} \cup \{x_{R(a,b_i)} = \lambda\}$.

Here we give some examples to explain some of the completion rules. Examples for explaining the reasoning procedure in detail are omitted here for space reasons. Interested readers can refer to [37].

For example, if the extended ABox A_i^ε contains $\neg Young(John)$ $[0.8, 1]$. The reasoning procedure adds $Young(John)$ $[0, 0.2]$ into A_i^ε and $x_{\neg Young(John)} = [0.8, 1]$ and $x_{Young(John)} = [0, 0.2]$ into the constraint set.

Assume the user specifies Zadeh Logic. If the extended ABox A_i^ε contains $(Young \sqcap Tall)(John)$ $[0.8, 1]$, by applying the concept conjunction rule, the reasoning procedure adds $Young(John)$ $x_{Young(John)}$ and $Tall(John)$ $x_{Tall(John)}$ into A_i^ε, and adds $min(x_{Young(John)}, x_{Tall(John)}) = [0.8, 1]$ into the constraint set. If the user specifies Product Logic, then the reasoning procedure instead adds the constraint $x_{Young(John)} * x_{Tall(John)} = [0.8, 1]$ into the constraint set.

If the extended ABox A_i^ε contains $(Young \sqcup Tall)(John)$ $[0.8, 1]$, by applying the concept disjunction rule, the reasoning procedure adds $Young(John)$ $x_{Young(John)}$ and $Tall(John)$ $x_{Tall(John)}$ into A_i^ε, and adds $max(x_{Young(John)}, x_{Tall(John)}) = [0.8, 1]$ into the constraint set if Zadeh Logic is chosen. If the user specifies Product Logic, then the reasoning procedure adds the constraint $x_{Young(John)} + x_{Tall(John)} - x_{Young(John)} * x_{Tall(John)} = [0.8, 1]$ into the constraint set instead.

Assume the user specifies Zadeh Logic. If the extended ABox A_i^ε contains $(\forall hasDisease.Disease)(P001)$ $[0.8, 1]$ and $hasDisease(P001, Cancer)$ $[0, 0.6]$, by applying the value restriction rule, the reasoning procedure adds $Disease(Cancer)$ $x_{Disease(Cancer)}$ into A_i^ε, and adds $max(x_{\neg hasDisease(P001,Cancer)}, x_{Disease(Cancer)}) = x_{(\forall hasDisease.Disease)(P001)}$ and $x_{(\forall hasDisease.Disease)(P001)} = [0.8, 1]$ into the constraint set. Now assume the extended ABox A_i^ε contains another assertion $(\forall hasDisease.Disease)(P001)$ $[0.7, 1]$; the reasoning procedure will replace $x_{(\forall hasDisease.Disease)(P001)} = [0.8, 1]$ in the constraint set with $x_{(\forall hasDisease.Disease)(P001)} = [0.7, 1]$.

If the extended ABox A_i^ε contains $(\leq 2 hasDisease)(P001)$ $[0.6, 1]$, $hasDisease(P001, Disease1)$ $[0.6, 1]$, $hasDisease(P001, Disease2)$ $[0.6, 1]$, $hasDisease(P001, Disease3)$ $[0.7, 1]$, $Disease1 \neq Disease2$ and $Disease1 \neq Disease3$, by applying the at-most number restriction rule, the reasoning procedure replaces $hasDisease(P001, Disease3)$ $[0.7, 1]$ with $hasDisease(P001, Disease2)$ $[0.7, 1]$, $Disease1 \neq Disease3$ with $Disease1 \neq Disease2$ in the extended ABox A_i^ε, and replaces replace $x_{hasDisease(P001,Disease3)} = [0.7, 1]$ in the constraint set with $x_{hasDisease(P001,Disease2)} = [0.7, 1]$.

The completion rules in Table 4 are a set of consistency-preserving transformation rules. Each time the reasoning procedure applies a completion rule, it either detects a clash or derives one or more assertions and constraints. In the reasoning procedure, the application of some completion rules, including the role existential restrictions and at-least number restrictions, may lead to nontermination. Therefore, we have to find some blocking strategy to ensure the termination of the reasoning procedure.

Definition 8 *Let a, b be anonymous individuals in the extended ABox A_i^ε; let $A_i^\varepsilon(a)$ (respectively, $A_i^\varepsilon(b)$) be all the assertions in A_i^ε that are related to the individual a (respectively, b); let $\mathcal{C}_j(a)$ (respectively, $\mathcal{C}_j(b)$) be all the constraints in $\mathcal{C}_j$ that are related to a (respectively, b), $\mathcal{L}(a) = \{A_i^\varepsilon(a), \mathcal{C}_j(a)\}$*

and $\mathcal{L}(b) = \{A_i^{\varepsilon}(b), \mathcal{C}_j(b)\}$ be the node labels for a and b. An individual b is said to be ***blocked*** *by a if $\mathcal{L}(b) \subseteq \mathcal{L}(a)$.*

8 Soundness, Completeness, and Termination of the Reasoning Procedure for *f$\mathcal{ALCN}$*

Extending results for standard DL [2][1], the following lemmas show that the reasoning procedure for $f\mathcal{ALCN}$ is sound and complete. Together with the proof of termination, it is shown that the consistency of an $f\mathcal{ALCN}$ knowledge base is decidable. Note that our proof can be viewed as a norm-parameterized version of the soundness, completeness, and termination of the algorithm presented in [30] for $f\mathcal{ALC}$ as well as the $f\mathcal{ALCN}$ counterpart in [29].

Lemma 1 ***Soundness*** *Assume that $\mathcal{A}_{i+1}^{\varepsilon}$ is obtained from the extended $f\mathcal{ALCN}$ ABox $\mathcal{A}_i^{\varepsilon}$ by application of a completion rule, then $\mathcal{A}_{i+1}^{\varepsilon}$ is consistent iff $\mathcal{A}_i^{\varepsilon}$ is consistent.*

Proof:

$\Longrightarrow$ This is straightforward. Let $\mathcal{C}_i$ and $\mathcal{C}_{i+1}$ be the constraint set associated with the extended ABox $\mathcal{A}_i^{\varepsilon}$ and $\mathcal{A}_{i+1}^{\varepsilon}$, respectively. Let $I = (\Delta^I, \cdot^I)$ be a fuzzy interpretation, and $\Phi : Var(\mathcal{C}) \to [0,1]$ be a solution. From the definition of consistency, we know if $\mathcal{A}_{i+1}^{\varepsilon}$ is consistent, there should exist a model $< I, \Phi >$. Since $\mathcal{A}_i^{\varepsilon} \subseteq \mathcal{A}_{i+1}^{\varepsilon}$ and $\mathcal{C}_i \subseteq \mathcal{C}_{i+1}$, $< I, \Phi >$ is also a model of $\mathcal{A}_i^{\varepsilon}$, therefore, $\mathcal{A}_i^{\varepsilon}$ is consistent.

$\Longleftarrow$ This is a consequence of the definition of the completion rules. Let C and D be concept descriptions, a and b be individual names, and R be an atomic role. Let $< I, \Phi >$ be a model of $\mathcal{A}_i^{\varepsilon}$. Now we show the interpretation I also satisfies the new assertions when any of the completion rules is triggered. Hereafter, let λ denote a variable for a truth degree.

Case: When $\mathcal{A}_i^{\varepsilon}$ contains $(\neg C)(a)\ \lambda$, we apply the conjunction rule and obtain the extended ABox $\mathcal{A}_{i+1}^{\varepsilon} = \mathcal{A}_i^{\varepsilon} \cup \{C(a)\ x_{C(a)}\}$ and the constraint set $\mathcal{C}_{i+1} = \mathcal{C}_i \cup \{1 - x_{C(a)} = x_{\neg C(a)} = \lambda\}$. As $< I, \Phi >$ is a model of $\mathcal{A}_i^{\varepsilon}$, I satisfies $\{(\neg C)(a)\ \lambda\}$, that is, $\{(\neg C)^I(a) = \lambda\}$. Based on the semantics of concept negation, we know that $(C)^I(a) = 1 - (\neg C)^I(a)$. Therefore, $(C)^I(a)) = 1 - \lambda$ holds under the interpretation I. Let v_1 be a value in $1 - \lambda$, we have $C^I(a) = v_1$. Hence, I also satisfies $\{C(a)\ v_1\}$.

Case: When $\mathcal{A}_i^{\varepsilon}$ contains $(C \sqcap D)(a)\ \lambda$, we apply the conjunction rule and obtain the extended ABox $\mathcal{A}_{i+1}^{\varepsilon} = \mathcal{A}_i^{\varepsilon} \cup \{C(a)\ x_{C(a)}, D(a)\ x_{D(a)}\}$ and the constraint set $\mathcal{C}_{i+1} = \mathcal{C}_i \cup \{t(x_{C(a)}, x_{D(a)}) = \lambda\}$. As $< I, \Phi >$ is a model of $\mathcal{A}_i^{\varepsilon}$, I satisfies $\{(C \sqcap D)(a)\ \lambda\}$, that is, $\{(C \sqcap D)^I(a) = \lambda\}$. Based on the semantics of concept conjunction, we know that $(C \sqcap D)^I(a) = t(C^I(a), D^I(a))$. Therefore, $t(C^I(a), D^I(a)) = \lambda$ holds under the interpretation I. It is easily verified that there are values v_1, v_2 ($v_1, v_2 \in [0,1]$) which satisfy $t(v_1, v_2) = \lambda$

[3]. Therefore, we have $C^I(a) = v_1$ and $D^I(a) = v_2$. Hence, I also satisfies both $\{C(a)\ v_1\}$ and $\{D(a)\ v_2\}$.

Case: When $\mathcal{A}_i^\varepsilon$ contains $(C \sqcup D)(a)\ \lambda$, the disjunction rule is applied and we obtain either the extended ABox $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_i^\varepsilon \cup \{C(a)\ x_{C(a)}, D(a)\ x_{D(a)}\}$ in the cases of Product Logic and Yager Logic, or, two extended ABoxes: $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_i^\varepsilon \cup \{C(a)\ x_{C(a)}\}$ and $\mathcal{A}_{i+1}^{\varepsilon'} = \mathcal{A}_i^\varepsilon \cup \{D(a)\ x_{D(a)}\}$ in the cases of other logics in Table 2. We also obtain the constraint set $\mathcal{C}_{i+1} = \mathcal{C}_i \cup \{s(x_{C(a)}, x_{D(a)}) = \lambda\}$. As $< I, \Phi >$ is a model of $\mathcal{A}_i^\varepsilon$, I satisfies $\{(C \sqcup D)(a)\ \lambda\}$, that is, $\{(C \sqcup D)^I(a) = \lambda\}$. Based on the semantics of concept conjunction, we know that $(C \sqcup D)^I(a) = s(C^I(a), D^I(a))$. Therefore, $s(C^I(a), D^I(a)) = \lambda$ holds under the interpretation I. It is easily verified that there are values v_1, v_2 ($v_1, v_2 \in [0, 1]$) which satisfy $s(v_1, v_2) = \lambda$ [4]. Therefore, we have $C^I(a) = v_1$ or $D^I(a) = v_2$. Hence, I also satisfies either $\{C(a)\ v_1\}$ or $\{D(a)\ v_2\}$, or both.

Case: When $\mathcal{A}_i^\varepsilon$ contains $(\exists R.C)(a)\ \lambda$, the role exists restriction rule is applied. There are two possible augmentations.

(1)If there exists an individual name b such that $C(b)\ x_{C(b)}$ and $R(a, b)\ x_{R(a,b)}$ are in $\mathcal{A}_i^\varepsilon$, but $\mathcal{C}_i$ does not contain $t(x_{C(b)}, x_{R(a,b)}) = x_{(\exists R.C)(a)}$, then $\mathcal{C}_{i+1} = \mathcal{C}_i \cup \{t(x_{C(b)}, x_{R(a,b)}) = x_{(\exists R.C)(a)}\}$; If λ is not the variable $x_{((\exists R.C)(a)}$, then if there exists $x_{((\exists R.C)(a)} = \lambda'$ in $\mathcal{C}_i$, then $\mathcal{C}_{i+1} = \mathcal{C}_{i+1} \backslash \{x_{((\exists R.C)(a)} = \lambda'\} \cup \{x_{((\exists R.C)(a)} = sup(\lambda, \lambda')\}$, else add $\mathcal{C}_{i+1} = \mathcal{C}_{i+1} \cup \{x_{(\exists R.C)(a)} = \lambda\}$. There is no new assertion, thus it is straightforward that if $\mathcal{A}_{i+1}^\varepsilon$ is consistent, then $\mathcal{A}_i^\varepsilon$ is consistent.

(2) If there is no individual name b such that $C(b)\ x_{C(b)}$ and $R(a, b)\ x_{R(a,b)}$ are in $\mathcal{A}_i^\varepsilon$, and $\mathcal{C}_j$ does not contain $t(x_{C(b)}, x_{R(a,b)}) = x_{(\exists R.C)(a)}$, then we obtain the extended ABox $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_i^\varepsilon \cup \{C(b)\ x_{C(b)}, R(a, b)\ x_{R(a,b)}\}$ and the constraint set $\mathcal{C}_{i+1} = \mathcal{C}_i \cup \{t(x_{C(b)}, x_{R(a,b)}) = x_{(\exists R.C)(a)}\}$. In this case, we want to show I also satisfies both $R(a, b)$ to some degree and $C(b)$ to some degree. As $< I, \Phi >$ is a model of $\mathcal{A}_i^\varepsilon$, I satisfies $(\exists R.C)(a)\ \lambda$, that is, $(\exists R.C)^I(a)\ \lambda$. Based on the semantics of role exists restriction, we know that $(\exists R.C)^I(a, b) = \sup_{b \in \Delta^I}\{t(R^I(a, b), C^I(b))\}$. Therefore, $\sup_{b \in \Delta^I}\{t(R^I(a, b), C^I(b))\} = \lambda$ holds under the interpretation I. It is easily verified that there are an individual b and values v_1, v_2 ($v_1, v_2 \in [0, 1]$) which satisfy $t(R^I(a, b), C^I(b))$. Therefore, we have $R^I(a, b) = v_1$ and $C^I(b) = v_2$. Hence, I also satisfies $\{R(a, b)\ v_1\}$ and $\{C(b)\ v_2\}$.

Case: When $\mathcal{A}_i^\varepsilon$ contains $(\forall R.C)(a)\ \lambda$, the role value restriction rule is applied. Then, for every individual b that is an R-successor of individual a, we obtain $\mathcal{A}_{i+1}^\varepsilon = \mathcal{A}_i^\varepsilon \cup \{C(b)\ x_{C(b)}\}$, $\mathcal{C}_{i+1} = \mathcal{C}_i \cup \{s(x_{C(b)}, x_{\neg R(a,b)}) = x_{(\forall R.C)(a)}\}$. If λ is not the variable $x_{((\forall R.C)(a)}$, then if there exists $x_{((\forall R.C)(a)} = \lambda'$ in $\mathcal{C}_i$,

[3] For different t-norms in Fuzzy Logic (the minimum function, the maximum function, the product function, and the Yager-and function), it is easy to show that we can always find such a pair of values.

[4] For different s-norms in Fuzzy Logic (the minimum function, the maximum function, the product-sum function, and the yager-or function), it is again easy to show that we can always find such a pair of values.

add $\mathcal{C}_{i+1} = \mathcal{C}_{i+1} \backslash \{x_{((\forall R.C)(a)} = \lambda^{'}\} \cup \{x_{((\forall R.C)(a)} = inf(\lambda, \lambda^{'})\}$, otherwise, add $\mathcal{C}_{i+1} = \mathcal{C}_{i+1} \cup \{x_{(\forall R.C)(a)} = \lambda\}$.

As $< I, \Phi >$ is a model of $\mathcal{A}_i^{\varepsilon}$, I satisfies $(\forall R.C)(a)\ \lambda$, that is, $(\forall R.C)^I(a) = \lambda$. For every individual b that is an R-successor of a, I satisfies $R(a,b)\ \lambda'$, that is, $R^I(a,b) = \lambda'$. Based on the semantics of role value restriction, we know that, $(\forall R.C)^I(a) = \inf_{b \in \Delta^I}\{s(\neg R^I(a,b), C^I(b))\}$. Therefore,$\inf_{b \in \Delta^I}\{s(\neg R^I(a,b), C^I(b))\} = \lambda$. Therefore, for every individual b that is an R-successor of a, $s(\neg\lambda', C^I(b)) = \lambda$ holds under the interpretation I. Hence, for each of these individuals b, we can find a value v_1 ($v_1 \in [0,1]$) which satisfies $s(\neg\lambda', C^I(b)) = \lambda$. Therefore, we have $C^I(b) = v_1$. Hence, I also satisfies $\{C(b)\ v_1\}$.

Case: When $\mathcal{A}_i^{\varepsilon}$ contains $(\geq nR)(a)\ \lambda$, the at-least number restriction rule is applied. We obtain $\mathcal{A}_{i+1}^{\varepsilon} = \mathcal{A}_i^{\varepsilon} \cup \{R(a,b_i)\ \lambda | 1 \leq i \leq n\} \cup \{b_i \neq b_j | 1 \leq i < j \leq n)\}$ and $\mathcal{C}_{i+1} = \mathcal{C}_i \cup \{x_{R(a,b_i)} = \lambda | 1 \leq i \leq n)\}$. Based on the semantics of at-least number restriction, we know that $(\geq nR)^I(x) = \sup_{y_1,\ldots,y_n \in \Delta^I, y_i \neq y_j, 1 \leq i < j \leq n} t_{i=1}^n\{R^I(x,y_i)\}$. Since it is easy to see from the application of the at-least number restriction rule, we can form at least n pairs (a,b_i) for which $R(a,b_i)^I = \lambda_i$ and $\sup_{y_1,\ldots,y_n \in \Delta^I, y_i \neq y_j, 1 \leq i < j \leq n} t_{i=1}^n\{\lambda_i\} = \lambda$. Hence, $(\geq nR)^I(a) = \lambda$ and I satisfies $(\geq nR)(a)\ \lambda$.

Case: When $n+1$ distinguished individual names $b_1, \ldots, b_{n+1}$ such that $(\leq nR)(a)\ \lambda$ and $R(a,b_i)\ \lambda_i$ $(1 \leq i \leq n+1)$ are contained in $\mathcal{A}_i^{\varepsilon}$, $b_i \neq b_j$ is not in $\mathcal{A}_i^{\varepsilon}$ for some $i \neq j$, the at-most number restriction rule is applied. For each pair b_i, b_j such that $j > i$ and $b_i \neq b_j$ is not in $\mathcal{A}_i^{\varepsilon}$, the ABox $\mathcal{A}_{i+1}^{\varepsilon}$ is obtained from $\mathcal{A}_i^{\varepsilon}$ and the constraint set $\mathcal{C}_{i+1}$ is obtained from $\mathcal{C}_i$ by replacing each occurrence of b_j by b_i, and if λ_i is the variable $x_{R(a,b_i)}$, $\mathcal{C}_{i+1} = \mathcal{C}_{i+1} \cup \{x_{R(a,b_i)} = \lambda\}$.

As $< I, \Phi >$ is a model of $\mathcal{A}_i^{\varepsilon}$, I satisfies $(\leq nR)(a)\ \lambda$, that is $(\leq nR)^I(a) = \lambda$. Based on the semantics of at-most number restriction, we know that if there are $n+1$ R-role assertions $R(a,b_i)$ $(i \in \{1, 2, ldots, n+1\})$ that can be formed from $\mathcal{A}_{i+1}^{\varepsilon}$, for which $R(a,b_i)^I = \lambda_i$, there would be at least one pair (a,b_k) for which $\lambda_k = \neg\lambda$ holds. Applying negation on both side of the equation, we thus have $\neg R(a,b_i)^I = \neg\neg\lambda = \lambda$ holds. This is equal to $\inf_{y_1,\ldots,y_{n+1} \in \Delta^I, y_i \neq y_j, 1 \leq i < j \leq n+1} s_{i=1}^{n+1}\{\neg R^I(x,y_i)\} = \lambda$. Therefore, we finally have that $(\leq nR)^I(x) = \lambda$, and I also satisfies $(\leq nR)(a)\ \lambda$.

$\square$

Lemma 2 ***Completeness*** *Any complete and clash-free $f\mathcal{ALCN}$ ABox $\mathcal{A}$ with a solvable constraints set $\mathcal{C}$ has a model.*

Proof:

Let $\mathcal{A}$ be a complete and clash-free ABox and $\mathcal{C}$ be the constraint set associated with $\mathcal{A}$. Since $\mathcal{A}$ is clash-free and complete, and the constraint set $\mathcal{C}$ is solvable, there exists a solution $\Phi : Var(\mathcal{C}) \rightarrow [0,1]$ to the constraint set $\mathcal{C}$.

Let's define the fuzzy interpretation I of the ABox $\mathcal{A}$ as follows:

(1) the domain $\triangle^I$ consists of all the individual names occurring in $\mathcal{A}$;

(2) for all atomic concepts A, we define $A^I(a) = \Phi(x_{A(a)})$ where a is an individual name in $\mathcal{A}$ and $\Phi(x_{A(a)})$ denotes the truth degree of the variable $x_{A(a)}$.

(3) for all atomic roles R_0 we define $R_0^I(a,b) = \Phi(x_{R_0(a,b)})$ where a and b are individual names in $\mathcal{A}$ and $\Phi(x_{R(a,b)})$ denotes the truth degree of the variable $x_{R(a,b)}$.

To show that the pair $< I, \Phi >$ is a model of $\mathcal{A}$, we need to prove all the concept and role assertions in $\mathcal{A}$ can be interpreted by I, using induction techniques on the structure of an $f\mathcal{ALCN}$ concept C.

If C is an atomic concept, then we have C^I according to its definition.

If C is of the form $C = \neg A$, $\mathcal{A}$ contains $\{\neg A(a)\ \lambda\}$. Since $\mathcal{A}$ is complete, we know the concept negation rule has been applied, thus $\mathcal{A}$ contains $\{A(a)\ \neg\lambda\}$ and $\mathcal{C}$ contains $x_{A(a)} = \neg\lambda$ and $x_{\neg A(a)} = \lambda$. By the induction hypothesis we know that I can interpret $A(a)\ \neg\lambda$, that is, $A^I(a) = \neg\lambda$. Based on the semantics of concept negation, we have $(\neg A)^I(a) = \neg A^I(a)$. Therefore, we obtain $(\neg A)^I(a) = \neg(\neg\lambda) = \lambda$; thus concept assertions of the form $\{\neg A(a)\ \lambda\}$ are correctly interpreted by I.

If C is of the form $C \sqcap D$, $\mathcal{A}$ contains $\{(C \sqcap D)(a)\ \lambda\}$. Since $\mathcal{A}$ is complete, we know the concept conjunction rule has been applied, thus $\mathcal{A}$ contains $\{C(a)\ x_{C(a)}\}$ and $\{D(a)\ x_{D(a)}\}$ and $\mathcal{C}$ contains $t(x_{C(a)}, x_{D(a)}) = \lambda$. By the induction hypothesis we know that I can interpret $C(a)\ x_{C(a)}$ and $D(a)\ x_{D(a)}$, that is, $C^I(a) = \Phi(x_{C(a)})$ and $D^I(a) = \Phi(x_{D(a)})$ where $\Phi(x_{C(a)})$ and $\Phi(x_{D(a)})$ denote the truth degrees of the variables $x_{C(a)}$ and $x_{D(a)}$, respectively. As Φ is a solution to the constraint set $\mathcal{C}$, $t(\Phi(x_{C(a)}), \Phi(x_{D(a)})) = \lambda$ holds. Therefore, $t(C^I(a), D^I(a)) = \lambda$. On the other hand, based on the semantics of concept conjunction, we have $(C \sqcap D)^I(a) = t(C^I(a), D^I(a))$. Therefore, we obtain $(C \sqcap D)^I(a) = \lambda$; thus concept assertions of the form $(C \sqcap D)(a)\ \lambda$ are correctly interpreted by I.

If C is of the form $C \sqcup D$, $\mathcal{A}$ contains $\{(C \sqcup D)(a)\ \lambda\}$. Since $\mathcal{A}$ is complete, we know the concept disjunction rule has been applied; thus $\mathcal{A}$ contains $\{C(a)\ x_{C(a)}\}$ or $\{D(a)\ x_{D(a)}\}$ and $\mathcal{C}$ contains $s(x_{C(a)}, x_{D(a)}) = \lambda$. By the induction hypothesis we know that I can interpret $C(a)\ x_{(a)}$ and $D(a)\ x_{D(a)}$, that is, $C^I(a) = \Phi(x_{C(a)})$ and $D^I(a) = \Phi(x_{D(a)})$ where $\Phi(x_{C(a)})$ and $\Phi(x_{D(a)})$ denote the truth degrees of the variables $x_{C(a)}$ and $x_{D(a)}$, respectively. Note that for a concept assertion $C(a)$, $C^I(a) = \Phi(x_{C(a)}) = 0$ still means I can interpret $C(a)$. As Φ is a solution to the constraint set $\mathcal{C}$, $s(\Phi(x_{C(a)}), \Phi(x_{D(a)})) = \lambda$ holds. Therefore, $s(C^I(a), D^I(a)) = \lambda$. On the other hand, based on the semantics of concept conjunction, we have $(C \sqcup D)^I(a) = s(C^I(a), D^I(a))$. Therefore, we obtain $(C \sqcup D)^I(a) = \lambda$; thus concept assertions of the form $(C \sqcup D)(a)\ \lambda$ are correctly interpreted by I.

If R is an atomic role, then we have R^I according to its definition.

If C is of the form $\exists R.C$, $\mathcal{A}$ contains $\{(\exists R.C)(a)\ \lambda\}$. Since $\mathcal{A}$ is complete, we know the role exists restriction rule has been applied. There could be three cases when applying the role exists restriction rule. (1) A new individual b was generated. $\mathcal{A}$ contains $\{R(a,b)\ x_{R(a,b)}\}$ and $\{C(b)\ x_{C(b)}\}$,

$\mathcal{C}$ contains $t(x_{R(a,b)}, x_{C(b)}) = \lambda$; (2) An individual b and $R(a,b)$ λ' already exist in $\mathcal{A}$. Then we have $\mathcal{A}$ contains $\{C(b)\ x_{C(b)}\}$ and $\mathcal{C}$ contains $t(x_{R(a,b)}, x_{C(b)}) = \lambda$ as well as $x_{R(a,b)} = \lambda'$ if λ' is not the variable $x_{R(a,b)}$; (3) a was blocked by some ancestor. In all these cases, we can find at least one individual b such that $C(b)\ x_{C(b)}$ and $R(a,b)\ x_{R(a,b)}$ is in $\mathcal{A}$, and $t(x_{R(a,b)}, x_{C(b)}) = \lambda$ is in $\mathcal{C}$. By the induction hypothesis, we know that I can interpret $C(b)\ x_{C(b)}$ and $R(a,b)\ x_{R(a,b)}$, that is, $C^I(b) = \Phi(x_{C(b)})$ and $R^I(a,b) = \Phi(x_{R(a,b)})$ where $\Phi(x_{C(b)})$ and $\Phi(x_{R(a,b)})$ denote the truth degrees of the variables $x_{C(b)}$ and $x_{R(a,b)}$, respectively. As Φ is a solution to the constraint set $\mathcal{C}$, $\sup_{b\in\Delta^I}\{t(x_{\Phi(C(b))}, x_{\Phi(R(a,b))})\} = \lambda$ holds. Therefore, $\sup_{b\in\Delta^I}\{t(x_{C^I(b)}, x_{R^I(a,b)})\} = \lambda$. On the other hand, based on the semantics of concept conjunction, we have $(\exists R.C)^I(a) = \sup_{b\in\Delta^I}\{t(x_{C^I(b)}, x_{R^I(a,b)})\}$. Therefore, we obtain $(\exists R.C)^I(a) = \lambda$; thus concept assertions of the form $(\exists R.C)(a)\ \lambda$ are correctly interpreted by I.

If C is of the form $\forall R.C$, $\mathcal{A}$ contains $\{(\forall R.C)(a)\ \lambda\}$. Since $\mathcal{A}$ is complete, we know the value restriction rule has been applied. Thus $\{C(b)\ x_{C(b)}\}$ is in $\mathcal{A}$ and $s(\neg x_{R(a,b)}, x_{C(b)}) = \lambda$ is in $\mathcal{C}$ for every individual b with $\{R(a,b)\ \lambda'\}$ in $\mathcal{A}$. $\mathcal{C}$ also contains $x_{R(a,b)} = \lambda'$ if λ' is not a variable. By the induction hypothesis, we know that I can interpret $C(b)\ x_{C(b)}$ for each b, that is, $C^I(b) = \Phi(x_{C(b)})$ for each b where $\Phi(x_{C(b)})$ denote the truth degree of the variable $x_{C(b)}$. As Φ is a solution to the constraint set $\mathcal{C}$, $\inf_{b\in\Delta^I}\{s(x_{\Phi(C(b))}, \neg x_{\Phi(R(a,b))})\} = \lambda$ holds. Therefore, $\inf_{b\in\Delta^I}\{s(x_{C^I(b)}, \neg x_{R^I(a,b)})\} = \lambda$. On the other hand, based on the semantics of value restriction, we have $(\forall R.C)^I(a) = \inf_{b\in\Delta^I}\{s(x_{C^I(b)}, \neg x_{R^I(a,b)})\}$. Therefore, we obtain $(\forall R.C)^I(a) = \lambda$; thus concept assertions of the form $(\forall R.C)(a)\ \lambda$ are correctly interpreted by I.

If C is of the form $\geq nR$, $\mathcal{A}$ contains $\{\geq nR(a)\ \lambda\}$. Since $\mathcal{A}$ is complete, we know the at-least number restriction rule has been applied. Thus $\{R(a,b_i)\ \lambda | 1 \leq i \leq n\}$ is in $\mathcal{A}$. By the induction hypothesis, we know that I satisfies $R(a,b_i)\ \lambda$ with $1 \leq i \leq n$, that is, $R^I(a,b_i) = \lambda$. Thus we can form at least n pairs (a,b_i) for which $R(a,b_i)^I = \lambda$ holds. Therefore, $\sup_{b_1,\ldots,b_n\in\Delta^I, b_i\neq b_j, 1\leq i<j\leq n} t_{i=1}^{n}\{R^I(a,b_i)\} = \lambda$. On the other hand, based on the semantics of at-least number restriction, we have $(\geq nR)^I(a) = \sup_{b_1,\ldots,b_n\in\Delta^I, b_i\neq b_j, 1\leq i<j\leq n} t_{i=1}^{n}\{R^I(a,b_i)\}$. Therefore, we obtain $(\geq nR)^I(a) = \lambda$; thus the interpretation I satisfies concept assertions of the form $(\geq nR)(a)\ \lambda$.

If C is of the form $\leq nR$, $\mathcal{A}$ contains $\{\leq nR(a)\ \lambda\}$. Since $\mathcal{A}$ is complete, we know the at-most number restriction rule has been applied. Thus for each pair b_i, b_j with $j > i$, the inequality $b_i \neq b_j$ is not in A_i^ε, the ABox A_i^ε is obtained and the constraint set $\mathcal{C}_i$ is obtained by replacing each occurrence of b_j by b_i, and if λ_i is the variable $x_{R(a,b_i)}$, add $\{x_{R(a,b_i)} = \lambda\}$ to $\mathcal{C}_i$. By the induction hypothesis, we know that I can interpret the resultiing n assertions $R(a,b_i)\ x_{R(a,b_i)}$ with $1 \leq i \leq n$, that is, $R^I(a,b_i) = \Phi(x_{R(a,b_i)})$, where Φ denote the truth degrees of the variables $R(a,b_i)\ x_{R(a,b_i)}$ for $(1 \leq i \leq n)$. As Φ is a solution to the constraint set $\mathcal{C}$, we have $\Phi(x_{R(a,b_i)}) \subseteq \lambda$ for $1 \leq i \leq n$.

Therefore, $x_{R^I(a,b_i)} = \lambda$ holds for $(1 \leq i \leq n)$. On the other hand, based on the semantics of at-most number restriction, we know if there are at most n pairs (a, b_i) for which $R(a, b_i)^I = \lambda$ holds, we have $(\leq nR)^I(a) = \lambda$. Thus, concept assertions of the form $(\leq nR)(a)\ \lambda$ are correctly interpreted by I.

□

In order to prove the termination of the $f\mathcal{ALCN}$ reasoning procedure, we first review the definition of $sub(D)$ given in [12]:

1. if D is an atomic concept, then $sub(D) = \{D\}$;
2. if D is of the form $C \sqcap D$, then $sub(D) = \{C \sqcap D\} \cup sub(C) \cup sub(D)$;
3. if D is of the form $C \sqcup D$, then $sub(D) = \{C \sqcup D\} \cup sub(C) \cup sub(D)$;
4. if D is of the form $\exists R.C$, then $sub(D) = \{\exists R.C\} \cup sub(C)$;
5. if D is of the form $\forall R.C$, then $sub(D) = \{\forall R.C\} \cup sub(C)$;
6. if D is of the form $\geq R$, then $sub(D) = \{\geq R\}$;
7. if D is of the form $\leq R$, then $sub(D) = \{\leq R\}$;

From the definition, we know that $sub(D)$ is the closure of subexpressions of D. When testing the consistency of an extended ABox $\mathcal{A}$, the concepts derived from the tableau procedure are restricted to subsets of any concept D (i.e., $sub(D)$) in $\mathcal{A}$. Therefore, we have $sub(\mathcal{A}) = \cup_{\forall D \in \mathcal{A}} sub(\mathcal{D})$.

Lemma 3 ***Termination*** *Let $\mathcal{A}$ be an f$\mathcal{ALCN}$ ABox. The tableaux procedure for f$\mathcal{ALCN}$ always terminates when started from $\mathcal{A}$.*

Proof: Let $R_{\mathcal{A}}$ be the set of roles occurring in $\mathcal{A}$. Let $C_{\mathcal{A}} = |sub(\mathcal{A})|$, $n_{\max} = \max\{n| \geq nR \in sub(\mathcal{A})\}$. The termination of our tableau procedure is a consequence of the same properties that ensure termination in the case of the standard $\mathcal{ALCN}$ DL. These properties are shown as follows [14]:

1. The only completion rule that remove assertions from the extended ABox is the at-most number restriction rule, which merges the assertions of an individual b with one of its ancestors a and thus individual b is blocked.

2. New individuals are only generated by the role exists restriction rule and the at-least number restriction rule. For each individual in the extended ABox, the rules can only be applied once. sub($\mathcal{A}$) contains at most s role exists restrictions which generates at most s successors. Each of these successors can further have $n_{\max}$ edges due to the at-least number restrictions. Therefore, the out-degree of the tree generated from the tableau procedure is bounded by $C_{\mathcal{A}} * n_{\max}$.

3. There is a finite number of possible labels for a pair of nodes and an edge, since concepts are taken from sub($\mathcal{A}$). Thus, there are at most $2^{C_{\mathcal{A}} * n_{\max}}$ possible labels for a pair of nodes and an edge. Hence, if a path is of length at least $2^{C_{\mathcal{A}} * n_{\max}}$, there must exist two nodes along the path that have the same node labels, and hence blocking occurs. Since a path cannot grow longer once a blocking takes place, paths are of length at most $2^{C_{\mathcal{A}} * n_{\max}}$.

□

9 Conclusion and Future Work

In this paper we propose an extension to Description Logics based on Fuzzy Set Theory and Fuzzy Logic. The syntax and semantics of the proposed Description Logic $f\mathcal{ALCN}$ are explained in detail. We further address different reasoning tasks on $f\mathcal{ALCN}$ knowledge bases. We present a sound and complete reasoning procedure that always terminates and its completion rules.

The $f\mathcal{ALCN}$ DL adopts a norm-parameterized way to cover different logics in the Fuzzy Logic family, currently Zadeh Logic, Lukasiewicz Logic, Product Logic, Gödel Logic, and Yager Logic. Such an approach allows the interpretation of different kinds of uncertain knowledge existing in real world applications. Furthermore, $f\mathcal{ALCN}$ knowledge bases can express fuzzy subsumption of fuzzy concepts of the form $C \sqsubseteq D\ [l, u]$, which allows generalized modeling of uncertain knowledge.

Description Logics constitute a family of descriptive languages with different expressiveness and decidability/efficiency. For reasons of simplicity, our fuzzy Description Logic $f\mathcal{ALCN}$ does not yet include transitive roles, inverse roles and other non-$\mathcal{ALC}$ constructors. We have also considered $\mathcal{ALCHIN}$ as a super language of $\mathcal{ALCN}$ and introduced a fuzzy version $f\mathcal{ALCHIN}$ [36]. Part of our ongoing work considers further fuzzy extensions to more expressive $\mathcal{S}$-style (i.e., $\mathcal{ALCR}^{+}$) Description Logics.

One of the main practical directions for future work is the implementation of the fuzzy reasoner, which involves a lot of technical designing decisions. We are implementing an $f\mathcal{ALCN}$ reasoner using SWI-Prolog. Our prototype reasoner is based on the $\mathcal{ALC}$ reasoner ALCAS [26] which supports $\mathcal{ALC}$ DL reasoning with an OWL abstract syntax. Our extensions to ALCAS provides functionalities to check consistency as well as fuzzy concept and subsumption entailments of a $f\mathcal{ALCN}$ knowledge base.

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, Cambridge, MA (2003)
2. Baader, F., Sattler, U.: An overview of tableau algorithms for description logic. Studia Logica **69**(1), 5–40 (2001)
3. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. Scientific American **284**(5), 34–44 (2001). URL `http://web.ebscohost.com/ehost/detail?vid=1&hid=102&sid=40d1a318-d0ac-41c6-9854-0ffde44a63eb%40sessionmgr109`
4. Bobillo, F., Straccia, U.: A fuzzy description logic with product t-norm. In: Proceedings of the IEEE International Conference on Fuzzy Systems (Fuzz IEEE-07), pp. 652C–657. IEEE Computer Society (2007)
5. Brachman, R.J., Levesque, H.J.: The tractability of subsumption in frame-based description languages. In: Proceedings AAAI-1984, pp. 34–37. AAAI Press (1984)

6. Haase, P., Völker, J.: Ontology learning and reasoning - dealing with uncertainty and inconsistency. In: Proceedings of Uncertainty Reasoning for the Semantic Web, pp. 45–55 (2005)
7. Hájek, P.: Metamathematics of fuzzy logic. Kluwer (1998)
8. Hájek, P.: Making fuzzy description logics more expressive. Fuzzy Sets Syst. **154**(1), 1–15 (2005)
9. Hájek, P.: What does mathematical fuzzy logic offer to description logic?, pp. 91–100. Fuzzy Logic and the Semantic Web, Capturing Intelligence. Elsevier (2006)
10. Hajek, P.: Fuzzy logic. In: The Stanford Encyclopedia of Philosophy. Stanford University (2009). URL `http://plato.stanford.edu/archives/spr2009/entries/logic-fuzzy/`
11. Hollunder, B.: An alternative proof method for possibilistic logic and its application to terminological logics. International Journal of Approximate Reasoning **12**(2), 85–109 (1995)
12. Horrocks, I.: Optimising tableaux decision procedures for description logics (1997). ANNOTE: AKA: Horrocks97b
13. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From $\mathcal{SHIQ}$ and RDF to OWL: The making of a web ontology language. J. of Web Semantics **1**(1), 7–26 (2003)
14. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for very expressive description logics. Logic Journal of the IGPL **8**(3), 239–264 (2000)
15. Hsueh-Ieng, P.: Uncertainty management for description logic-based ontologies. Ph.D. thesis, University of Concordia (2008)
16. Jaeger, M.: Probabilistic reasoning in terminological logics. In: Proc. of the 4th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR94), pp. 305–316 (1994)
17. Koller, D., Levy, A., Pfeffer, A.: P-classic: A tractable probabilistic description logic. In: Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97), pp. 390–397 (1997)
18. Laskey, K.J., Laskey, K.B., Costa, P.C.G., Kokar, M.M., Martin, T., Lukasiewicz, T.: W3c incubator group report. Tech. Rep. http://www.w3.org/2005/Incubator/urw3/wiki/DraftFinalReport, W3C (05 March, 2008)
19. Lukasiewicz, T.: Expressive probabilistic description logics. Artificial Intelligence **172**(6/7), 852–883 (2008)
20. Martin-Recuerda, F., Robertson, D.: Discovery and uncertainty in semantic web services. In: Proceedings of Uncertainty Reasoning for the Semantic Web, p. 188 (2005)
21. McGuinness, D.L., van Harmelen, F.: Owl web ontology language overview (2004). URL `http://www.w3.org/TR/owl-features/`
22. Montagna, F., Marini, C., Simi, G.: Product logic and probabilistic ulam games. Fuzzy Sets Syst. **158**(6), 639–651 (2007). DOI http://dx.doi.org/10.1016/j.fss.2006.11.007
23. Motik, B., Grau, B.C., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C.: Owl 2 web ontology language profiles (2009). URL `http://www.w3.org/TR/owl2-profiles/#Reasoning_in_OWL_2_RL_and_RDF_Graphs_using_Rules`
24. Novák, V., Perfilieva, I., Mockor, J.: Mathematical principles of fuzzy logic. Dodrecht: Kluwer Academic (1999)
25. Sánchez, D., Tettamanzi, A.G.: Fuzzy quantification in fuzzy description logics, pp. 135–160. Fuzzy Logic and the Semantic Web, Capturing Intelligence. Elsevier (2006)
26. Spencer, B.: ALCAS: An ALC Reasoner for CAS. http://www.cs.unb.ca/ bspencer/cs6795swt/alcas.prolog (2006). URL `http://www.cs.unb.ca/~bspencer/cs6795swt/alcas.prolog`

27. Stamou, G., van Ossenbruggen, J., Pan, J.Z., Schreiber, G.: Multimedia annotations on the semantic web. IEEE MultiMedia **13**, 86–90 (2006). DOI http://doi.ieeecomputersociety.org/10.1109/MMUL.2006.15
28. Stevens, R., Aranguren, M.E., Wolstencroft, K., Sattlera, U., Drummond, N., Horridge, M., Rectora, A.: Using owl to model biological knowledge. International Journal of Human-Computer Studies **65**(7), 583–594 (2007)
29. Stoilos, G., Stamou, G., Pan, J.Z., Tzouvaras, V., Horrocks, I.: Reasoning with very expressive fuzzy description logics. Journal of Artificial Intelligence Research **30**, 273–320 (2007)
30. Straccia, U.: Reasoning within fuzzy description logics. Journal of Artificial Intelligence Research **14**, 137–166 (2001)
31. Straccia, U.: Towards a fuzzy description logic for the semantic web (preliminary report). In: 2nd European Semantic Web Conference (ESWC-05), Lecture Notes in Computer Science, pp. 167–181. Springer Verlag (2005)
32. Tresp, C.B., Molitor, R.: A description logic for vague knowledge. In: Proc. of the 13th Eur. Conf. on Artificial Intelligence (ECAI'98), pp. 361–365 (1998)
33. Yen, J.: Generalizing term subsumption languages to fuzzy logic. In: Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91), pp. 472–477 (1991)
34. Zadeh, L.A.: Fuzzy sets. Information and Control **8**(3), 338–353 (1965)
35. Zhao, J.: Uncertainty and Rule Extensions to Description Logics and Semantic Web Ontologies, chap. 1, p. 22. Advances in Semantic Computing. Technomathematics Research Foundation (2010). Accepted
36. Zhao, J., Boley, H.: A Reasoning Procedure for the Fuzzy Description Logic fALCHIN. In: Proc. Second Canadian Semantic Web Working Symposium, Kelowna, pp. 46–59 (2009)
37. Zhao, J., Boley, H., Du, W.: Knowledge Representation and Consistency Checking in a Norm-Parameterized Fuzzy Description Logic. In: D.S. Huang, K.H. Jo, H.H. Lee, H.J. Kang, V. Bevilacqua (eds.) ICIC (2), *Lecture Notes in Computer Science*, vol. 5755, pp. 111–123. Springer (2009). URL `http://dx.doi.org/10.1007/978-3-642-04020-7`