Front. Comput. Sci. China 2011, 5(2): 205–215
DOI 10.1007/s11704-011-0109-8

RESEARCH ARTICLE

Type-2 fuzzy description logic

Ruixuan LI (✉)[1], Kunmei WEN[1], Xiwu GU[1], Yuhua LI[1], Xiaolin SUN[1], Bing LI[2]

1 Intelligent and Distributed Computing Laboratory, School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
2 State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072, China

Abstract Description logics (DLs) are widely employed in recent semantic web application systems. However, classical description logics are limited when dealing with imprecise concepts and roles, thus providing the motivation for this work. In this paper, we present a type-2 fuzzy attributive concept language with complements (ALC) and provide its knowledge representation and reasoning algorithms. We also propose type-2 fuzzy web ontology language (OWL) to build a fuzzy ontology based on type-2 fuzzy ALC and analyze the soundness, completeness, and complexity of the reasoning algorithms. Compared to type-1 fuzzy ALC, type-2 fuzzy ALC can describe imprecise knowledge more meticulously by using the membership degree interval. We implement a semantic search engine based on type-2 fuzzy ALC and carry out experiments on real data to test its performance. The results show that the type-2 fuzzy ALC can improve the precision and increase the number of relevant hits for imprecise information searches.

Keywords description logic (DL), type-2 fuzzy attributive concept language with complements (ALC), fuzzy ontology, reasoning, semantic search engine

1 Introduction

The semantic web was proposed by Berners-Lee in 1991 and gained rapid development in the following two decades [1]. Ontology, is a formal knowledge representation, and plays a very important role in many semantic web applications, such as semantic search systems [2]. Description logic (DL) [3] is usually employed to represent the knowledge and logic of an ontology. It represents the knowledge of an application domain by defining the relevant concepts of the domain (terminology) and using these concepts to specify properties of objects and individuals that belong to the domain (world description). In the family of knowledge representation formalisms, DLs have the powerful ability of describing knowledge that makes it express the information more easily in different application domains [4]. DLs are considered the most important knowledge representation formalism unifying and giving a logical basis to the well known traditions of frame-based systems, semantic networks and KL-ONE-like languages, object-oriented representations, semantic data models, and type systems. The prototypical DL attributive concept language with complements (ALC) was introduced by Schmidt-Schaubß and Smolka [5] in 1991, and is the basis of many more expressive DLs.

A majority of semantic web applications use ALC like classic DLs to represent the knowledge and reason on the ontology. Heflin et al. [6,7] propose SHOE that can find semantic annotations from web pages. Guha et al. [8] have developed Tap, which applies the technology of semantic web to Google to increase the quality of the retrieval. Users can get the results offered by Tap only if the query matches the individuals belonging to its ontology. Ding et al. [9,10] have designed Swoogle for information retrieval in structured documents, such as resource

Received May 20, 2010; accepted July 24, 2010

E-mail: rxli@hust.edu.cn

description framework (RDF) [11] and web ontology language (OWL) [12,13]. Nevertheless, it cannot be integrated into a traditional search engine easily. At present, more and more semantic web application systems are designed using ontologies based on classic DLs. However, classical DLs can only define crisp concepts and properties. The certainty reasoning of classic DLs only returns *true* or *false* as the answer of the inference. It cannot deal with fuzzy information that meets the application requirements in real world ontology systems.

To address the above problem, fuzzy DLs are presented by extending the classic DLs to support the imprecise information processing in ontology systems. To the best of our knowledge, most of the fuzzy logic systems (FLSs) are based on type-1 fuzzy sets, which were proposed by Zadeh in 1965 [14]. The fuzzy sets have recently been applied to DLs and ontology Systems. Without a reasoning algorithm, Meghini et al. [15] proposed a preliminary fuzzy DL as a tool for modeling multimedia document retrieval. Straccia et al. [16] presented a formalized fuzzy ALC (FALC) in 2001, which is a type-1 fuzzy extension of ALC. They also extend SHOIN(D), the corresponding DL of the standard ontology description language OWL DL, to a fuzzy version [17,18]. Li et al. [19,20] also present a fuzzy extension of DLs, named ALCH, and some reasoning technique for the extended fuzzy DLs. They also study the family of extended fuzzy DLs [21]. Jiang et al. [22] presented a fuzzy description logic framework based on certainty lattices. Its main feature is that an assertion is not just *true* or *false* as in classical description logics, but certain to some degree, where the certainty value is taken from a certainty lattice. Jiang et al. [23] also gave an integration of the theories of intuitionistic fuzzy DLs and rough DLs by providing intuitionistic fuzzy rough DLs based on intuitionistic fuzzy rough set theory. Hajek [24] presented a version of fuzzy description logic based on the basic fuzzy predicate logic BL and reduced the problems of satisfiability, validity and subsumption of concepts to problems of fuzzy propositional logic that can be decidable for any continuous t-norm. In the field of combining fuzzy set theory with information search, Jin et al. [25] proposed a special ranking mechanism based on the weighed fuzzy query representation, and formulated user's search request through tightly combining fuzziness together with the user's subjective weighting importance over multiple search properties.

However, there are some limitations to type-1 fuzzy DLs. They cannot describe the imprecise knowledge of the concepts and individuals by membership degree interval. For example, we may say that the degree of a person at the age of 30 being a young person is between 0.6 and 0.9, rather than using a value, such as 0.85, to represent the degree. These types of limitations make it difficult for FALC and fuzzy SHOIN(D) to express complex fuzzy information clearly. To address these issues, we propose a type-2 fuzzy ALC and give the syntax, semantics and the reasoning algorithm for the type-2 fuzzy ALC [26]. We also propose the type-2 fuzzy OWL to build the fuzzy ontology based on type-2 fuzzy ALC logic and analyze the soundness, completeness and complexity of the reasoning algorithm. Compared with type-1 fuzzy ALC, type-2 fuzzy ALC can describe imprecise knowledge more rigorously through using membership degree intervals. In addition, we present a semantic search engine framework based on type-2 fuzzy ALC and carry out experiments on real data to test the performance of the proposed approach. The results show that the type-2 fuzzy ALC can improve the precision and increase the number of relevant hits for imprecise information search.

The main contributions of this paper are as follows:

- First, we propose the syntax and semantics of a type-2 fuzzy extension of ALC, and provide the formal axioms, which enable users to represent imprecise information in semantic web applications.
- Second, we provide reasoning algorithms in type-2 fuzzy ALC and discuss the soundness, completeness, and complexity problems.
- Third, we present a formalized ontology description language, type-2 fuzzy OWL, based on type-2 fuzzy ALC. It is a feasible implementation of type-2 fuzzy ALC using classical semantic web standards.
- Fourth, we design and implement a semantic search system based on type-2 fuzzy ALC and carry out experiments to demonstrate its performance.

Section 2 gives the basic concepts of typical ALC description logics and the preliminary introduction of type-1 fuzzy ALC. Section 3 presents the syntax and semantics of type-2 fuzzy ALC. Section 4 gives the reasoning algorithms in type-2 fuzzy ALC. The type-2 fuzzy OWL, an implementation of type-2 fuzzy ALC, is discussed in Section 5. Section 6 designs a semantic search framework using a type-2 fuzzy OWL ontology. We follow this with conclusions and some potentially interesting future directions in Section 7.

2 Preliminary

2.1 Basic concepts of description logics

A knowledge system based on description logics provides facilities to set up a knowledge base, and operate and reason on the knowledge base. A knowledge base comprises two components: TBox and ABox [27]. TBox introduces the terminology, i.e., the vocabulary of an application domain, while ABox contains assertions about named individuals in terms of the specific vocabulary. The vocabulary consists of concepts denoting sets of individuals and roles indicating binary relationships between the individuals. The description logic used in the knowledge system should not only store the terminologies and assertions, but also provide services for reasoning on the knowledge base.

Description logics are a family of formal knowledge representation languages. They model concepts, roles, individuals, and their relationships [3]. A description logic system is characterized by four fundamental aspects: the set of constructs used in concept and role expressions, the kind of assertions allowed in TBox (assertions on concepts) and ABox (assertions on individuals), and the inference mechanisms for reasoning on both TBox and ABox. The classification of concepts in description logics determines sub-concept relationships (called subsumption relationships in description logics) between the concepts of a given terminology, or super-concept relationship. Thus, it allows users to structure the terminology in the form of a subsumption hierarchy. This hierarchy provides useful information in the connection between different concepts, and it can be used to speed-up the inference services.

2.2 Typical ALC and type-1 fuzzy ALC

The attributive concept language with complements (ALC) [5] is the fundamental basis for many other expressive description logics. ALC concepts and roles are built as follows. Suppose that A denotes set of atomic concepts, C denotes set of complex concepts defined by descriptions and R denotes set of roles. Starting with: 1) A, B ∈ A; 2) C, D ∈ C and 3) R ∈ R. The concept terms in TBox can be defined as the following formats inductively: C ⊑ f (A, B, R, ⊓, ⊔, ∀, ∃, ⊥, ⊤) (partial definition), and C ≡ f (A, B, R, ⊓, ⊔, ∀, ∃, ⊥, ⊤) (full definition), where ⊥ and ⊤ are two special atomic concepts named *bottom concept* and *universe concept*, respectively. The syntax and semantics of ALC constructors have previously been presented in [3].

As we have mentioned, classic description logics, such as ALC, cannot deal with the imprecise information that becomes a popular requirement in real world systems. For example, an individual can only belong to one concept to some extent, or an axiom can only be true to a certain degree in terms of a concept. To solve these types of problem in description logics, FALC, an extension of ALC with fuzzy features, is presented to support fuzzy concept representation [16]. FALC uses a certain value of numeral to describe the fuzziness of the concepts and individuals. We call this type of FALC as type-1 fuzzy ALC.

Similarly, A, C and R are defined as sets of atomic fuzzy concepts, complex fuzzy concepts, and fuzzy roles in FALC, respectively. It is easy to prove that C ⊓ D, C ⊔ D, ¬C, ∀R.C and ∃R.C are also fuzzy concepts [16,17]. The fuzzy interpretation in FALC is a pair $I = (\Delta^I, \cdot^I)$, where $\cdot^I$ is an interpretation function mapping fuzzy concepts and roles into membership degrees, such that $C^I = \Delta^I \rightarrow [0,1]$ and $R^I = \Delta^I \times \Delta^I \rightarrow [0,1]$. The most important part of FALC is the use of the membership degree function to describe the individuals that belong to fuzzy concepts. However, using a determined value to denote the membership degree has many restrictions. It can only give a limited degree of the membership; for example, the daughter is 30% like her father. However, it cannot represent a more vague relationship; for instance, the daughter is 30%–60% like her father. In fact, many membership degrees of the concepts and individuals in real world applications cannot be exactly expressed by a single value. It may fall into a range of some interval. Hence, we propose type-2 fuzzy ALC to address this problem.

3 Type-2 fuzzy ALC

3.1 Basic concepts of type-2 fuzzy sets

In this section, we propose type-2 fuzzy sets, which use an interval to represent the membership degree instead of the single value employed in type-1 fuzzy sets. In type-2 fuzzy sets, the degree of the membership is normalized to the interval of [0, 1]. We denote the membership in type-2 fuzzy sets with $\overline{\mu_A}$, instead of μ_A as used in type-1 fuzzy sets, which is defined as follows

$$\overline{\mu_A}(x) = \left[\mu_A^L(x), \mu_A^U(x)\right]. \qquad (1)$$

In Eq. (1), $\mu_A^L(x)$,$\mu_A^U(x)$: $U \rightarrow [0,1]$, and $\forall x \in U, \mu_A^L(x) \leqslant \mu_A^U(x)$. We call $\mu_A^L(x)$ and $\mu_A^U(x)$ the lower and upper bounds of the membership degree, and x is an element in the fuzzy set U. It is obvious that a type-2 fuzzy set can be reduced to a type-1 fuzzy set when the lower bound equals the upper bound of the membership degree. Therefore, the set of all possible type-1 fuzzy sets is a subset of the set of all possible type-2 fuzzy sets.

To calculate the probability value of the uncertainty with some fuzzy rules, we introduce the *triangular norm* to type-2 fuzzy sets, which is widely used in fuzzy set theory.

We call the interpretation a T triangular norm if it satisfies the following conditions:

1) $T(0, 0) = 0$, $T(1, 1) = 1$;
2) $a \leqslant c$, $b \leqslant d \Rightarrow T(a,b)T(c,d)$;
3) $T(a, b) = T(b, a)$;
4) $T(T(a, b), c) = T(a, T(b, c))$.

A triangular norm is named T-norm if $T(a, 1) = a$, $(a \in [0,1])$, while a triangular norm is called S-norm if $T(0, a) = a$, $(a \in [0,1])$. The properties of the basic T-norm and S-norm are given as follows:

For any T-norm T and S-norm S, we have

$$T_0'(a,b) = \begin{cases} a, & b = 1, \\ b, & a = 1, \\ 0, & \text{Others}, \end{cases}$$

$$S_0'(a,b) = \begin{cases} a, & b = 0, \\ b, & a = 0, \\ 1, & \text{Others}, \end{cases}$$

$$T_0(a,b) = a \wedge b,\ S_0(a,b) = a \vee b,$$
$$T_1(a,b) = a \cdot b,\ S_1(a,b) = a + b - a \cdot b,$$
$$T_2(a,b) = \frac{a \cdot b}{1 + (1-a)(1-b)},$$
$$S_2(a,b) = \frac{a+b}{1 + a \cdot b},$$
$$T^\lambda(a,b) = \frac{a \cdot b}{\lambda + (1-\lambda)(a+b-ab)},$$
$$S^\lambda(a,b) = \frac{a + b + (\lambda-2)ab}{1 + (\lambda-1)ab}, (\lambda \geqslant 0),$$
$$T^Y(a,b) = 1 - \min\left(1, \left((1-a)^Y + (1-b)^Y\right)^{1/Y}\right),$$
$$S^Y(a,b) = \min\left(1, \left(a^Y + b^Y\right)^{1/Y}\right), (Y \geqslant 1),$$
$$T_\infty(a,b) = \max(0, a+b-1),$$
$$S_\infty(a,b) = \min(1, a+b).$$

To calculate the value of the uncertainty, we can apply any pair of T-norm and S-norm to the proposed fuzzy system. Each pair of norms has different degree in the fuzzy description, and the degree can be ranked as follows

$$T_0' \leqslant T_\infty \leqslant T_2 \leqslant T_1 \leqslant T_0 \leqslant S_0 \leqslant S_1 \leqslant S_2 \leqslant S_\infty \leqslant S_0'.$$

3.2 TBox and ABox of type-2 fuzzy ALC

There are two fuzzy parts in type-2 fuzzy ALC presented in this paper, which are the imprecise terminological axiom (TBox) and fuzzy individual membership (ABox). To build a description logic system, the first step is to create TBox by defining necessary atomic concepts and roles. The atomic concepts can be divided into two sets, the name symbols, N_T, that appear in the axioms on the left side and the base symbols, B_T, that appear on the right side. Generally, no atomic concept can be defined more than once; that is, any name symbol can appear on the left side of the axioms at most once. In addition, every atomic concept can be expanded by base symbols in TBox. The atomic concepts defined by different axioms may be imprecise; that is, the axioms exist in classical description logic may not be true in type-2 fuzzy ALC TBox.

For example, given two base symbols named *Animal* and *FlyingObject*, we can define the atomic concept *Bird* in TBox with Axiom (2).

$$Bird \equiv Animal \sqcap FlyingObject. \tag{2}$$

However, we cannot ensure that the concept *Bird* is defined precisely in Axiom (2); for example, *Penguin* is a bird but cannot fly. If the probability of the bird being an animal that can fly is in [0.9, 0.95], we can change the axiom in ALC TBox into a fuzzy axiom in type-2 fuzzy ALC.

$$Bird_{[0.9,0.95]} \equiv Animal \sqcap FlyingObject. \tag{3}$$

Axiom (3) means that the probability that a *B*ird can be described with the conjunction of *Animal* and *FlyingObject* is between 0.9 and 0.95. Since the base symbols are a certainty, the probability of atomic concepts *Animal* and *FlyingObject* are both 1, in the interval [1,1]. For simplicity, we denote the certain atomic concept *Animal* without using the demarcation [1,1].

Type-2 fuzzy ALC can represent the vagueness of the atomic concepts with two properties, *fuzzy*:*LowerDegree* and *fuzzy*:*UpperDegree*. In that way, every description can be extended by using the two fuzzy properties, and the range of each property is a literal whose value is in interval [0, 1]. Hence, it is not difficult to extend the TBox in a description logic system to type-2 fuzzy ALC if we modify the default format of description in the description logic syntax with these two properties.

Triangular norms are used for calculating fuzzy degrees in a fuzzy sets system. Different triangular norms may cause different results when we calculate the fuzzy degrees among the same fuzzy concepts. Considering the property of the fuzzy degree and the computational complexity, we use the norm pair T_2 and S_2 in type-2 fuzzy ALC in this paper.

Suppose every atomic concept (role) is independent. We will calculate the *fuzzy*:*LowerDegree* and *fuzzy*: *UpperDegree* of fuzzy concepts if we do not know it beforehand. For example, we want to define an atomic concept *FleshEatingBird* with base symbol *FleshEatingObject* with Axiom (4) when we apply the norms T_2 (0.9, 1) and S_2 (0.95, 1):

$$\begin{aligned}&FleshEatingBird\\&\quad\equiv Bird_{[0.9,\ 0.95]}\sqcap FleshEatingObject.\end{aligned}\tag{4}$$

We can calculate the *fuzzy:LowerDegree* and *fuzzy: UpperDegree* of *FleshEatingBird* with the following equation:

$$\begin{aligned}&\mu^L(FleshEatingBird)\\&\quad=T_2\big(\mu^L(Bird),\mu^L(FleshEatingObject)\big),\end{aligned}$$

and

$$\begin{aligned}&\mu^U(FleshEatingBird)\\&\quad=S_2\big(\mu^U(Bird),\mu^U(FleshEatingObject)\big).\end{aligned}$$

We have

$$\mu^L(Bird)=0.9,$$

and

$$\mu^L(FleshEatingObject)=1.$$

Accordingly,

$$\begin{aligned}&\mu^L(FleshEatingBird)\\&\quad=T_2(0.9,1)=(0.9\times1)/[1+(1-1)(1-0.9)]\\&\quad=0.9.\end{aligned}$$

Similarly, we can get $\mu^U(FleshEatingBird)=0.95$. Hence, the membership degree of the atomic concept *FleshEatingBird* is [0.9, 0.95]. This type of the property of type-2 fuzzy ALC can be called transitivity.

However, the axioms cannot be extended using the same method with the atomic concepts. For example, Axiom (5) has the same meaning as Axiom (6) in classic description logics, while they are not equal in type-2 fuzzy ALC using the proposed calculation method.

$$FleshEatingBird\equiv Bird\sqcap FleshEatingObject,\tag{5}$$

$$\begin{aligned}&\text{FleshEatingBird}\\&\quad\equiv Animal\sqcap FlyingObject\sqcap FleshEatingObject.\end{aligned}\tag{6}$$

For example, the μ^L (*FleshEatingBird*) and μ^U (*FleshEatingBird*) calculated by Axiom (5) is [0.9, 0.95], while the degree value calculated by Axiom (6) is [1, 1], which makes Axiom (7) and Axiom (8) different.

$$\begin{aligned}&FleshEatingBird_{[0.9,0.95]}\\&\quad\equiv Bird_{[0.9,0.95]}\sqcap FleshEatingObject_{[1,1]},\end{aligned}\tag{7}$$

$$\begin{aligned}&FleshEatingBird_{[1,1]}\\&\quad\equiv Animal_{[1,1]}\sqcap FlyingObject_{[1,1]}\\&\qquad\sqcap FleshEatingObject_{[1,1]}.\end{aligned}\tag{8}$$

In addition to the fuzzy TBox, the uncertainty also exists in ABox in type-2 fuzzy description logics. Similar to type-1 fuzzy ALC, we use the interval to represent the degree of the assertion in ABox. For example, $C^I(x)=[a,b]$, where $0\leqslant a\leqslant b\leqslant 1$. Hence, ABox can be denoted by a set of equations with the form as follows: $C_{[a,\ b]}(x)=[c,d]$, where $C=f(A,B,R,\sqcap,\sqcup,\forall,\exists,\bot,\top)$. The assertions in ABox can be denoted by the following form: $C_{[a,\ b]}(x)_{[c,\ d]}$.

Taking the atomic concept $Bird_{[0.9,\ 0.95]}$ as an example, the fact that *Bird*(*Penguin*) is satisfied in ABox has two preconditions: 1) the concept *Bird* should be satisfied in TBox; 2) the instance *Penguin* belongs to the concept *Bird* in ABox. For example, the assertion $Bird_{[0.9,0.95]}(Penguin)_{[0.65,0.9]}$ means the degree that *Penguin* can be considered as an instance of $Bird_{[0.9,0.95]}$ is [0.65, 0.9]. Therefore, $Bird_{[0.9,0.95]}(Penguin)=[0.65,0.95]$ will be satisfied in ABox.

3.3 Syntax and semantics of type-2 fuzzy ALC

We use A, C and R to denote the set of atomic concepts, complex concepts, and roles respectively. $C\sqcap D$, $C\sqcup D$, $\neg C$, $\forall R.C$ and $\exists R.C$ are fuzzy concepts. The fuzzy

Table 1 Syntax and semantics of type-2 fuzzy ALC constructors

Constructor	Syntax	Semantics
Top (universe)	$\top$	Δ^I
Bottom (nothing)	$\bot$	Φ
Atomic concept	$A_{[a,b]}$	$A^I_{[a,b]} \subseteq \Delta^I$
Atomic role	$R_{[a,b]}$	$R^I_{[a,b]} \subseteq \Delta^I \times \Delta^I$
Conjunction	$C_{[a,b]} \sqcap D_{[c,d]}$	$(C \sqcap D)^I_{[T(a,c),T(b,d)]}$
Disjunction	$C_{[a,b]} \sqcup D_{[c,d]}$	$(C \sqcup D)^I_{[S(a,c),S(b,d)]}$
Negation	$\neg C_{[a,b]}$	$C^I_{[1-b,1-a]}$
Value restriction	$\forall R_{[a,b]} \cdot C_{[c,d]}$	$\forall y.S(R_{[1-b,1-a]}(x,y),C_{[c,d]}(y))$
Full existential quantification	$\exists R_{[a,b]} \cdot C_{[c,d]}$	$\exists y.T(R_{[a,b]}(x,y),C_{[c,d]}(y))$

interpretation in type-2 fuzzy ALC is a pair $I = (\Delta^I, \cdot^I)$, and interpreter $\cdot^I$ is an interpretation function that maps fuzzy concepts and roles into a membership degree interval: $C^I = \Delta^I \rightarrow [a, b]$ and $R^I = \Delta^I \times \Delta^I \rightarrow [a, b]$, where $0 \leqslant a \leqslant b \leqslant 1$. The interpretation function $\cdot^I$ of type-2 fuzzy ALC must satisfy the following equations. For any $d \in \Delta^I$,

$$\top^I(x) = [1,1],$$
$$\bot^I(x) = [0,0],$$
$$C^I(x) = [\mu^L(C(x)),\mu^U(C(x))],$$
$$(C \sqcap D)^I(x) = [T\{\mu^L(C(x)),\mu^L(D(x))\}, T\{\mu^U(C(x)),\mu^U(D(x))\}],$$
$$(C \sqcup D)^I(x) = [S\{\mu^L(C(x)),\mu^L(D(x))\}, S\{\mu^U(C(x)),\mu^U(D(x))\}],$$
$$\neg C^I(x) = [1-\mu^U(C(x)),1-\mu^L(C(x))],$$
$$(\forall R.C)^I(x) = inf^I_{d' \in \Delta}[S\{1-\mu^U(\mathrm{R}(x,x')),\mu^L(C(x'))\}, S\{1-\mu^L(\mathrm{R}(x,x')),\mu^U(C(x'))\}],$$
$$(\exists R.C)^I(x) = sup^I_{d' \in \Delta}[T\{\mu^L(\mathrm{R}(x,x')),\mu^L(C(x'))\}, T\{\mu^U(\mathrm{R}(x,x')),\mu^U(C(x'))\}].$$

The syntax and semantics of type-2 fuzzy ALC are shown in Table 1. Different from type-1 fuzzy ALC, the Δ^I of type-2 fuzzy ALC is not a set of numbers in [0, 1], but a set of pairs, which have the form $[a,b]$, where $0 \leqslant a \leqslant b \leqslant 1$.

4 Reasoning with type-2 fuzzy ALC

Reasoning technique plays a very important role in a logic. In the past decade, it has attracted much attention in the field of description logics. However, reasoning with description logics can be much more complicated than other logics since description logics have a strong ability to describe the world. Reasoning is also essential to type-2 fuzzy ALC and can be even more challenging. In this section, we will present reasoning techniques in type-2 fuzzy ALC.

4.1 Reasoning in fuzzy TBox with modifiers

In Section 3, we gave the basic concepts, syntax and semantics of type-2 fuzzy ALC. We use the same symbols with the same meaning in this section. There are two main reasoning rules in TBox of type-2 fuzzy ALC: negation rules and subsumption rules. We first propose the negation rules for reasoning in type-2 fuzzy ALC as follows.

$$\neg\neg C_{[a,b]} = C_{[a,b]},$$
$$\neg(C_{[a,b]} \sqcap D_{[c,d]}) = C_{[1-b\ 1-a]} \sqcup D_{[1-d\ 1-c]},$$
$$\neg(C_{[a,b]} \sqcup D_{[c,d]}) = C_{[1-b\ 1-a]} \sqcap D_{[1-d\ 1-c]},$$
$$\neg\forall R_{[a,b]} \cdot C_{[c,d]} = \exists R_{[a,b]} \cdot C_{[1-d\ 1-c]},$$
$$\neg\exists R_{[a,b]} \cdot C_{[c,d]} = \forall R_{[a,b]} \cdot C_{[1-d\ 1-c]}.$$

The subsumption reasoning rules in type-2 fuzzy ALC are similar to those in classic description logics except that the membership should also be taken into account. That means the $C_{[a,b]} \sqsubseteq D_{[c,d]}$ has two preconditions. One is that C is a subclass of D, in other words, $C_{[a,b]} = C_{[a,b]} \sqcap D_{[a,b]}$. The other is that a, b, c and d must satisfy: $0 \leqslant a \leqslant c \leqslant d \leqslant 1$ and $0 \leqslant a \leqslant b \leqslant d \leqslant 1$.

In addition, there are many more modifiers in type-2 fuzzy ALC than in classic description logics, such as *very*, *most*, and *less*, which are used to describe the different membership degree of the concepts and roles in TBox. The definition of the modifiers *very*, *most*, and *less* are as follows.

$$C \rightarrow (very)or(most)C : \mu^L((very)C) = (\mu^L(C))^2, \mu^U((very)C) = (\mu^U(C))^2, \tag{9}$$

$$C \rightarrow (less)C : \mu^L((less)C) = \sqrt{\mu^L(C)}, \mu^U((less)C) = \sqrt{\mu^U(C)}. \tag{10}$$

We can give the definition of other modifiers by using the method in Eqs. (9) and (10). These modifiers will help us describe the vague properties of the concepts and roles more clearly and exactly.

4.2 Fuzzy tableau algorithms

Tableau algorithms are the most famous and basic algorithms in description logic reasoning. We first introduce a brief process of the reasoning in tableau algorithms. Tableau algorithms use negation to reduce subsumption to the (un)satisfiability of concept descriptions instead of directly testing subsumption of concept descriptions. For example, $C \sqsubseteq D$ if and only if $\neg C \sqcap D = \perp$. We can check whether the concept is unsatisfiable through those algorithms. The fuzzy tableau algorithms begin with an ABox $A_0 = \{C_{[a,\ b]}(x)_{[c,\ d]}\}$ to check the (un)satisfiability of concept $C_{[a,\ b]}$. Since ALC has no number restrictions, we present five rules for fuzzy tableau algorithms as follows.

1) $\cap$-*rule*: if ABox A contains $C_{[a,\ b]}(x)_{[c,\ d]}$, and $C_{[e,\ f]}(x)_{[g,\ h]}$, where $[a, b] \cap [e, f] \neq \Phi$ and $[c, d] \cap [g, h] \neq \Phi$, the fuzzy tableau algorithms should extend A to

$$A' = A - \{C_{[a,b]}(x)_{[c,d]}, C_{[e,f]}(x)_{[g,h]}\}$$
$$\sqcup \{C_{[S0(a,e),T0(b,f)]}(x)_{[S0(c,g),T0(b,f)]}(x)_{[S0(c,g),T0(d,h)]}\},$$

otherwise

$$A' = A - \{C_{[a,\ b]}(x)_{[c,\ d]}, C_{[e,\ f]}(x)_{[g,\ h]}\}.$$

2) $\sqcap$-*rule*: if ABox A contains

$$(C'_{[e,f]} \sqcap C''_{[g\ h]})_{[a,b]}(x)_{[c,d]}$$
$$= (C' \sqcap C'')_{[T(T(e,f),a),T(T(g,h),b)]}(x)_{[c,d]},$$

but does not contain both $C'_{[e,f]}(x)_{[c,d]}$ and $C''_{[g,h]}(x)_{[c,d]}$, the fuzzy tableau algorithms should extend A to

$$A' = A \sqcup \{C'_{[e,f]}(x)_{[c,d]}, C''_{[g,h]}(x)_{[c,d]}\}.$$

3) $\sqcup$-*rule*: if ABox A contains

$$(C'_{[e,f]} \sqcup C''_{[g\ h]})_{[a,b]}(x)_{[c,d]}$$
$$= (C' \sqcup C'')_{[S(S(e,f),a)S(S(g,h),b)]}(x)_{[c,d]},$$

but neither $C'_{[e,f]}(x)_{[c,d]}$ nor $C''_{[g,h]}(x)_{[c,d]}$, the fuzzy tableau algorithms should extend A to

$$A' = A \sqcup \{C'_{[e,f]}(x)_{[c,d]}\}$$

or

$$A'' = A \sqcup \{C''_{[g,h]}(x)_{[c,d]}\}.$$

4) $\exists$-*rule*: if ABox A contains $(\exists R_{[e,f]} \cdot C_{[g,h]})(x)_{[c,d]}$, but no individuals z such that $R_{[e,\ f]}(x, z)_{[c,\ d]}$ and $C_{[g,\ h]}(z)_{[c,\ d]}$, the fuzzy tableau algorithms should extend A to

$$A' = A \sqcup \{R_{[e,f]}(x,y)_{[c,d]}, C_{[g,h]}(y)_{[c,d]}\},$$

where y is an individual not previously appearing in A.

5) $\forall$-*rule*: if ABox A contains $(\forall R_{[e,f]} \cdot C_{[g,h]})(x)_{[c,d]}$, and $R_{[e,\ f]}(x, y)_{[c,\ d]}$, but not $C_{[g,\ h]}(y)_{[c,\ d]}$, the fuzzy tableau algorithms should extend A to

$$A' = A \sqcup \{C_{[g,h]}(y)_{[c,d]}\}.$$

As we know, the $\sqcup$-*rule* will result in more than one ABox. Hence, we introduce the finite set $S = \{A_0, A_1, \ldots, A_n\}$ to represent the result ABoxes after enforcing the above reasoning rules on A_0 and its follow-up result ABoxes $A_1, \ldots, A_n$. We state that S is consistent if any ABox A_i ($i = 0\ldots n$) in S is consistent. Suppose we get a final set S_k by using the above fuzzy reasoning rules on A_0. If S_k has no obvious conflicts, such as $C_{[0,\ 0]}(x)$ or $\perp(x)$, we can consider $C_{[a,\ b]}$ is satisfiable.

The way to decide whether the ABox in type-2 fuzzy ALC is unsatisfiable is different from typical tableau algorithms. Suppose there are two limited values T_L and T_U. If the probability of any fuzzy concept or individual is less than T_L, we can consider the probability to be 0. It means that if $\mu^{L(U)}(C) \leqslant T_L$, the axiom to describe concept C will not exist in the description logic system. On the contrary, the probability of C will be 1 if $\mu^{L(U)}(C) \geqslant T_U$. Thus, the reasoning process of the fuzzy tableau algorithms will stop when any one of the following conditions is established.

1) Any obvious conflicts, such as $\perp(x)$ and $(C \sqcap \neg C)(x)$, are found in the reasoning results.

2) Any fuzzy conflicts occur in the reasoning process; for example, $C_{[0,\ 0]}(x) = [c, d]$, $C_{[a,\ b]}(x) = [c, d]$, $C_{[c,\ d]}(x) = [a, b]$ with $a \leqslant b \leqslant T_L$, $C_{[a,b]}(x)$ and $C_{[c,d]}(x)$ without overlap of the intervals $[a, b]$ and $[c, d]$.

3) All rules, including $\cap$-*rule*, $\sqcap$-*rule*, $\sqcup$-*rule*, $\exists$-*rule* and $\forall$-*rule*, have been executed.

4.3 Soundness, completeness and complexity issues

In this section, we will discuss the soundness, completeness and complexity issues of reasoning in type-2 fuzzy ALC.

● **Soundness** Assume that S is obtained from the finite set of ABoxes S_1 by applying the reasoning rules. Then S is consistent if and only if S_1 is consistent. It is easy to prove the soundness of the fuzzy reasoning algorithms in this way: if any ABox, A_i, in S_1 is conflict-free, S_1 has at least one model, which makes S conflict-free and $C_{[a,\ b]}$ satisfiable.

● **Completeness** As we know, any complete and conflict-free ABox, A, has a model. It is obvious that if we start with $A_0 = \{C_{0[a,\ b]}\ (x)_{[c,\ d]}\}$, we can obtain a finite individual tree whose root belongs to $C_{0[a,\ b]}$ and the concept $C_{0[a,\ b]}$ is satisfiable. The completeness can be satisfied since the proposed rules can reduce satisfiability of a type-2 FALC concept $C_{0[a,\ b]}$ (in negation normal form) to consistency of a finite set, S, of complete ABoxes.

● **Complexity** Since the number restriction is not considered in the fuzzy reasoning rules and every branch of the individual tree is independent, the complexity of the fuzzy tableau algorithms is $O(2n)$, which is much lower than the classic tableau algorithms that must handle $O(2^{n+1}-1)$ individuals.

5 Type-2 fuzzy OWL

Knowledge in the semantic web is usually structured in the form of ontology. This leads to considerable efforts to develop a suitable ontology language, culminating in the design of the OWL [12]. The OWL language consists of three sub-languages, namely OWL Lite, OWL DL and OWL Full, which have gradually increasing expression ability. OWL DL is widely used in many fields to represent the axioms and assertions. Its corresponding description logic is SHOIN(D +) [18]. Similar to the relationship between OWL DL and SHOIN (D +), we propose the type-2 fuzzy OWL to implement the type-2 fuzzy ALC. There are some differences between type-2 fuzzy OWL and classic OWL in the abstract syntax, such as *Description* and *Fact* in type-2 fuzzy OWL. We present the main part of the syntax of type-2 fuzzy OWL as follows.

Description:: = ‘classID (’ [classID]
{‘fuzzy: LowerDegree(’ Lowerdegree‘)’}
{‘fuzzy:UpperDegree(’ Upperdegree‘)’}
‘(’ Lowerdegree⩽Upperdegree‘)’‘)’
|restriction|‘unionOf (’ {description } ‘)’
|‘intersectionOf (’ {description } ‘)’
|‘complementOf (’ { description } ‘)’
|‘one of (’ { individualID } ‘)’
Fact:: = Individual
Individual:: = ‘Individual (’ [individualID] { annotation }
{
‘type (’type‘)’ {‘fuzzy:LowerDegree(’ Lowerdegree‘)’}
{‘fuzzy:UpperDegree(’ Upperdegree‘)’}
‘ }’
‘(’ Lowerdegree⩽Upperdegree‘)’‘)’
Lowerdegree::= degree
Upperdegree::= degree
degree::= ‘degree(’
real-number-between-0-and-1-inclusive‘)’…

The assertions in ABox we discussed above can also be implemented with type-2 fuzzy OWL. For example, we assert two instances *Eagle* and *Penguin* as follows. Suppose the membership degree is [0.84, 0.93] for the fact that *Eagle* can be considered as an instance of *FleshEatingObject*, and the membership degree is [0.22, 0.28] since *Penguin* can also be considered as an instance of *FleshEatingObject*. There are two equations to show the probability of that *Eagle* and *Penguin* are instances of *FleshEatingBird*, as shown in Eqs. (11) and (12).

$$FleshEatingBird_{[0.9,0.95]}(Eagle) = [T_2(0.84,0.9), T_2(0.93,0.95)] = [0.74,0.88], \quad (11)$$

$$FleshEatingBird_{[0.9,0.95]}(Penguin) = [T_2(0.22,0.9), T_2(0.28,0.95)] = [0.18,0.26]. \quad (12)$$

However, ABox in type-2 fuzzy ALC does not store the above equations as its assertions. The uncertain information stored in ABox is only the predefined membership degree instead of the calculated degree. We can get the result degree by using some reasoning rules and equations like Eqs. (11) and (12). The sample of type-2 fuzzy OWL for the instances *Eagle* and *Penguin* is presented as follows:

<*FleshEatingBird* fuzzy:LowerDegree = 0.84 fuzzy:UpperDegree = 0.93 rdf:ID =“*Eagle*”>

<*FleshEatingBird* fuzzy:LowerDegree = 0.22 fuzzy:UpperDegree = 0.28 rdf:ID =“*Penguin*”>…

6 Semantic search engine based on type-2 fuzzy ontology

6.1 Framework of a semantic search engine based on type-2 fuzzy ontology

Daily communications in real world, such as conversations using the natural languages, are full of imprecise information. To better model the real world problems, we need to not only employ the fuzzy description logic, but also need a query language to represent the vague requirements. We call the queries including fuzzy concepts and individuals fuzzy queries. To handle these fuzzy queries, ontology-based semantic search engine should extend their knowledge base to support fuzzy ontologies. In this paper, we implement a fuzzy semantic search engine using type-2 fuzzy OWL. Fig. 1 shows the framework of the semantic search engine based on a type-2 fuzzy ontology.

The semantic search engine based on a type-2 fuzzy ontology consists of six components: type-2 fuzzy ontology analyzer, type-2 fuzzy ontology questioner / answerer, keywords generator, search engine, type-2 fuzzy ontology, and document indexes. In this framework, users can propose their queries in two ways. One way is to ask the type-2 fuzzy ontology analyzer with keywords or fuzzy keywords. These keywords will be processed by the ontology analyzer and then sent to the keyword generator. The other way is to issue the semantic queries to type-2 fuzzy ontology questioner/answerer. The semantic queries will be parsed by the ontology analyzer. The parsed queries represented by ontology language will be sent to the ontology base. The corresponding names of the concepts and individuals will be sent to the keywords generator and then the search engine will perform the search on the classic document indexes by using these generated keywords.

Thus, users cannot only communicate with ontology base by forming the queries with concepts and individuals directly, but also issue keywords to a classic search engine to search documents. The results will contain two parts. One is the ontology results returned by the ontology base, which are represented in ontology language. The other is the document results returned by the classic search engine, which are represented as document lists.

6.2 Experiments and analysis

We implement the type-2 fuzzy semantic search engine based on the proposed framework. We use the type-2 fuzzy ALC to handle the imprecise information, such as *very*, *most*, and *high* as input by users. The ontology base is built using type-2 fuzzy OWL and the vague information of the concepts and individuals is modeled as an interval in the fuzzy ontology. We use academic resources as the test data set, which are collected from the departmental websites of Huazhong University of Science and Technology, including 7000 web pages and 2400 documents. The document type can be txt, xml, rdf, doc, and pdf. The type-2 fuzzy ontology analyzer, questioner, answer, keyword generator, and search engine are all implemented in Java. The ontology is built with Protégé [28] and contains 212 TBox nodes and 995 ABox nodes.

Since the information grows explosively on the Internet, precision plays a more important role than the recall rate for a search engine. Hence, we carry out experiments to test the precision of the implemented semantic search engine rather than its recall rate. We choose a group of keywords to retrieve information from indexes, and then pick out the relevant hits from the result set to test the precision of the query. Fig. 2 represents the precision of the semantic search engine based on classic ontology and type-2 fuzzy ontology. From the figure we can see that the precision grows faster with the number of nodes in the ontology under the type-2 fuzzy ontology. This means the precision of the search engine will be improved if a type-2 fuzzy ALC is applied.

Additionally, we carry out experiments to test how the

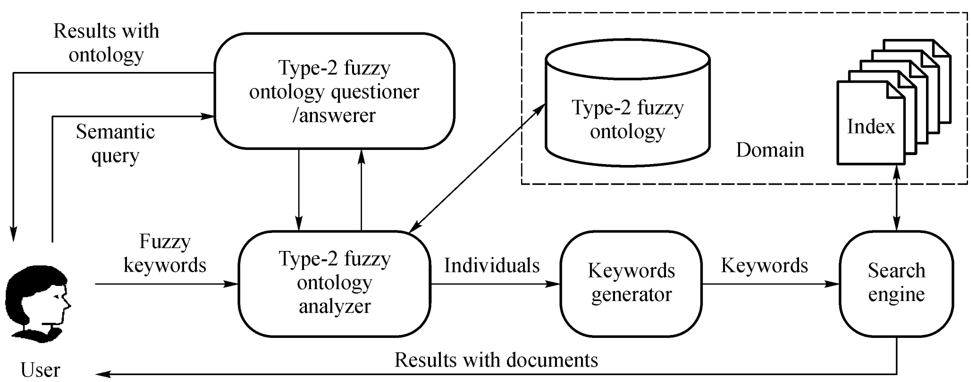

Fig. 1 Framework of semantic search engine based on type-2 fuzzy ontology

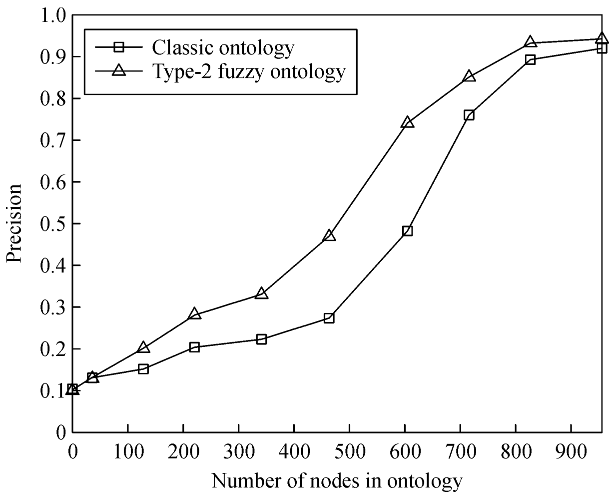

Fig. 2 Relevant hits-imprecision graph

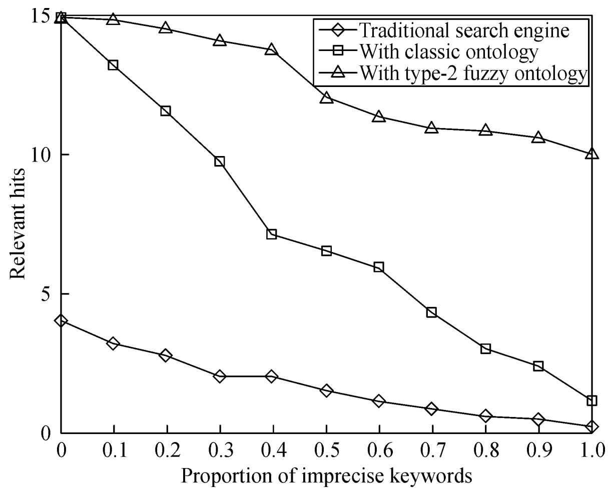

Fig. 3 Precision-nodes graph

imprecise information in the keywords affects the performance of the search engine. We compare traditional, classic ontology, and type-2 fuzzy ontology search engines. The results are shown in Fig. 3. From the figure we can see that both ontology based search engines expand the number of relevant hits greatly when there are fewer imprecise search terms. The reason being is that the ontology analyzer will generate more keywords with the individuals, which will provide more qualifiers for the query. However, the number of relevant hits of the search engine based on a classic ontology decreases rapidly when we add more fuzzy keywords into the query, such as *very*, *most*, and *young*. The type-2 fuzzy ontology based search engine, on the other hand, can process this imprecise information much better; that is to say that, it can improve the relevance of the query results.

7 Conclusion

We have proposed a type-2 fuzzy description logic to represent and infer the fuzzy information widely existing in real world applications. The syntax, semantics and reasoning algorithms of type-2 fuzzy ALC have been introduced as the fundamentals of type-2 fuzzy description logic. We have also tested the performance of the semantic search engine with type-2 fuzzy ALC through experiments. Compared to the traditional search engine and semantic search engine with classic ontology, we have found that the proposed type-2 fuzzy ALC search engine is more capable of dealing with the imprecise knowledge. Many other applications, besides semantic search engine, need to process fuzzy information, such as trust management in distributed systems. Our approach can be applied in different domains to strengthen their representation and reasoning ability. Our future work will include the research of type-2 fuzzy ALCN, type-2 fuzzy SHOIN (D) and the corresponding reasoning algorithms.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 60873225, 60873083, and 70771043), the National High Technology Research and Development Program of China (2007AA01Z403), the Natural Science Foundation of Hubei Province (2009CDB298), the Natural Science Foundation of Hubei Province for Distinguished Young Scholars (2008CDB351), the Wuhan Youth Science and Technology Chenguang Program (200950431171), the Open Foundation of State Key Laboratory of Software Engineering (SKLSE20080718), the Innovation Fund of Huazhong University of Science and Technology (2010MS068, Q2009021).

References

1. Stix G. The mice that warred: natural selection. Scientific American, 2001, 284(6): 34–35
2. Guha R, McCool R, Miller E. Semantic search. In: Proceeding of 12th International World Wide Web Conference. 2003, 700–709
3. Baader F, Calvanese D, McGuinness D L, Patel-Schneider P. The Description Logic Handbook: Theory, Implementation, and Applications. London: Cambridge University Press, 2003, 47–100
4. Calvanese D, Lenzerini M, Nardi D. Unifying class-based representation formalisms. Journal of Artificial Intelligence Research, 1999, 11(2): 199–240
5. Schmidt-Schaubß M, Smolka G. Attributive concept descriptions with complements. Artificial Intelligence, 1991, 48(1): 1–26
6. Heflin J D. Towards the semantic web: knowledge representation in a dynamic distributed environment. Dissertation for the Doctoral Degree. College Park: University of Maryland, 2001
7. Heflin J, Hendler J. Searching the web with shoe. In: Proceedings of AAAI-2000 Workshop on AI for Web Search. 2000, 35–40

8. Guha R, McCool R. TAP: A semantic web test-bed. Journal of Web Semantics, 2003, 1(1): 81–87
9. Ding L, Finin T, Joshi A, Pan R, Cost R S, Peng Y, Reddivari P, Doshi V, Sachs J. Swoogle: a search and metadata engine for the semantic web. In: Proceedings of 13th ACM International Conference on Information and Knowledge Management. 2004, 294–303
10. Finin T W, Mayfield J, Joshi A, Cost R S, Fink C. Information retrieval and the semantic web. In: Proceedings of 38th Hawaii International Conference on System Sciences. 2005
11. Manola F, Miller E. Resource Description Framework (RDF), W3C Recommendation. W3C, 2004, http://www.w3.org/RDF/
12. McGuinness D L, van Harmelen F. OWL Web Ontology Language Overview, W3C Recommendation. W3C, 2004, http://www.w3.org/TR/owl-features/
13. Brockmans S, Colomb R M, Haase P, Kendall E F, Wallace E K, Welty C, Xie G T. A model driven approach for building OWL DL and OWL full ontologies. In: Proceeding of 5th International Semantic Web Conference. 2006, 187–200
14. Zadeh L A. Fuzzy sets. Information and Control, 1965, 8(3): 338–353
15. Meghini C, Sebastiani F, Straccia U. Reasoning about the form and content for multimedia objects. In: Proceedings of AAAI 1997 Spring Symposium on Intelligent Integration and Use of Text, Image, Video and Audio. 1997, 89–94
16. Straccia U. Reasoning within fuzzy description logics. Journal of Artificial Intelligence Research, 2001, 14: 137–166
17. Straccia U. Description logics with fuzzy concrete domains. In: Proceedings of 21st Conference in Uncertainty in Artificial Intelligence. 2005, 559–567
18. Straccia U. Towards a fuzzy description logic for the semantic web. In: Proceedings of the 1st Fuzzy Logic and the Semantic Web Workshop, Marseille, 2005, 3–18
19. Li Y, Lu J, Xu B, Kang D, Jiang J. A fuzzy extension of description logic ALCH. In: Proceedings of 4th Mexican International Conference on Artificial Intelligence. 2005, 152–161
20. Li Y, Xu B, Lu J, Kang D, Xu J. Reasoning technique for extended fuzzy description logics. In: Proceedings of 17th IEEE International Conference on Tools with Artificial Intelligence. 2005, 665–666
21. Li Y, Xu B, Lu J, Kang D, Wang P. A family of extended fuzzy description logics. In: Proceedings of 29th Annual International Computer Software and Applications Conference. 2005, 221–226
22. Jiang Y, Tang Y, Wang J, Deng P, Tang S. Expressive fuzzy description logics over lattices. Knowledge-Based Systems, 2010, 23(2): 150–161
23. Jiang Y, Tang Y, Wang J, Tang S. Reasoning within intuitionistic fuzzy rough description logics. Information Sciences, 2009, 179 (14): 2362–2378
24. Hajek P. Making fuzzy description logic more general. Fuzzy Sets and Systems, 2005, 154(1): 1–15
25. Jin H, Ning X, Jia W, Wu H, Lu G. Combining weights with fuzziness for intelligent semantic web search. Knowledge-Based Systems, 2008, 21(7): 655–665
26. Li R, Sun X, Lu Z, Wen K, Li Y. Towards a type-2 fuzzy description logic for semantic search engine. In: Proceedings of joint conference of 9th Asia-Pacific Web Conference and 8th International Conference on Web-Age Information Management. 2007, 805–812
27. De Giacomo G, Lenzerini M. TBox and ABox reasoning in expressive description logics. In: Proceedings of 5th International Conference on the Principles of Knowledge Representation and Reasoning. 1996, 316–327
28. The Protégé Ontology Editor and Knowledge Acquisition System. http://protege.stanford.edu/