Towards a Type-2 Fuzzy Description Logic for Semantic Search Engine*

Ruixuan Li, Xiaolin Sun, Zhengding Lu, Kunmei Wen, and Yuhua Li

College of Computer Science and Technology,
Huazhong University of Science and Technology,
Wuhan 430074, Hubei, China
rxli@hust.edu.cn, arthursun@smail.hust.edu.cn,
{zdlu,kmwen,yhli3}@hust.edu.cn

Abstract. Classical description logics are limited in dealing with the crisp concepts and relationships, which makes it difficult to represent and process imprecise information in real applications. In this paper we present a type-2 fuzzy version of $\mathcal{ALC}$ and describe its syntax, semantics and reasoning algorithms, as well as the implementation of the logic with type-2 fuzzy OWL. Comparing with type-1 fuzzy $\mathcal{ALC}$, system based on type-2 fuzzy $\mathcal{ALC}$ can define imprecise knowledge more exactly by using membership degree interval. To evaluate the ability of type-2 fuzzy $\mathcal{ALC}$ for handling vague information, we apply it to semantic search engine for building the fuzzy ontology and carry out the experiments through comparing with other search schemes. The experimental results show that the type-2 fuzzy $\mathcal{ALC}$ based system can increase the number of relevant hits and improve the precision of semantic search engine.

Keywords: Semantic search engine, Description logic, Type-2 fuzzy $\mathcal{ALC}$, Fuzzy ontology.

1 Introduction

As the fundament of the semantic web [1,2], ontology is playing a very important role in many applications such as semantic search [3]. Being one of the logic supports of ontology, Description logics (DLs) [4] represent the knowledge of an application domain by defining the relevant concepts of the domain (terminologies) and using these concepts to specify properties of objects and individuals which belong to the domain (the world description). As one in the family of knowledge representation (KR) formalisms, the powerful ability of describing knowledge makes DLs express the information more easily in different application domains [5]. Being established by W3C in 2004, OWL (Web Ontology Language) [2,6] becomes the standard knowledge representation markup language for the semantic web.

* This work is supported by National Natural Science Foundation of China under Grant 60403027, Natural Science Foundation of Hubei Province under Grant 2005ABA258, Open Foundation of State Key Laboratory of Software Engineering under Grant SKLSE05-07, and a grant from Huawei Technologies Co., Ltd.

G. Dong et al. (Eds.): APWeb/WAIM 2007, LNCS 4505, pp. 805–812, 2007.

Being expected to be applied in semantic web, semantic search extended the search engine with ontology. Using general ontologies, the most current semantic search engines handle the information retrieval in semantic web based on classic DLs. University of Maryland proposed SHOE [7,8], which can find the semantic annotations from web pages. Tap [9,10,11] developed by Stanford University and IBM applies the technology of semantic web into Google, which augments the results in order to increase the quality of the retrieval. Swoogle [12,13,14] is designed for the information retrieval in structured documents such as RDF (Resource Description Framework), OWL and so on. At present, more and more semantic search systems are designed based on ontology that is supported by classic DLs.

But the classical DLs can only define the crisp concepts and properties, and the certain reasoning of classic DLs means that the answer of inference only returns "True" or "False", which cannot solve the fuzzy problem of ontology system in real world. Therefore, the fuzzy DLs are designed to expand the classic DLs to make it more applicable to ontology system. At present, most fuzzy logic systems (FLSs) are based on type-1 fuzzy sets, which were proposed by Zadeh in 1965 [15]. However, it was quite late when the fuzzy sets were applied to DLs and ontology System. Without reasoning algorithm, Meghini proposed a preliminary fuzzy DL as a tool for modeling multimedia document retrieval [16]. Straccia presented the formalized Fuzzy $\mathcal{ALC}$ ($\mathcal{FALC}$) [17] in 2001, which is a type-1 fuzzy extension of $\mathcal{ALC}$. Before long, Straccia extended the $\mathcal{SHOIN(D)}$, the corresponding DL of the standard ontology description language OWL DL, to a fuzzy version [18,19].

However, there are some limits in Type-1 fuzzy sets. For example the imprecision cannot be described by a crisp value clearly, which will result the loss of fuzzy information. To address the problem mentioned above, we propose a type-2 fuzzy $\mathcal{ALC}$ and try to apply it into semantic search engine. The contributions of the paper are as follows. First, we present the syntax and semantics of a type-2 fuzzy extension of $\mathcal{ALC}$, which can represent and reason fuzzy information with OWL, a formalized ontology description language. Besides the format of the axioms defined in Type-2 fuzzy $\mathcal{ALC}$, the reasoning algorithm is also proposed for semantic search. Finally, we design and realize the system of semantic search engine based on type-2 fuzzy $\mathcal{ALC}$ and carry out the experiments to evaluate the performance of the proposed search scheme.

The rest of the paper is organized as follows. Section 2 gives the condition of relative research and basic concepts of DL, typical $\mathcal{ALC}$ and type-1 fuzzy $\mathcal{ALC}$. Section 3 presents the format of the type-2 fuzzy $\mathcal{ALC}$ and the method of reasoning in type-2 fuzzy DL. Approaches for applying the type-2 fuzzy DL to deal with the description in fuzzy ontology for semantic search engine with OWL is addressed in section 4, followed by conclusions and future research of the paper.

2 Basic Concepts

$\mathcal{ALC}$ concepts and roles are built as follows. Use letter $\mathcal{A}$ for the set of atomic concepts, $\mathcal{C}$ for the set of complex concept defined by descriptions and $\mathcal{R}$ for the set of roles. Starting with: (1) A, B∈ $\mathcal{A}$ (2) C, D∈ $\mathcal{C}$ and (3) R∈ $\mathcal{R}$. The concept terms in TBox can be defined with the format as following inductively: C ⊑ f(A, B, R, ⊓, ⊔, ∀, ∃, ⊥, ⊤) (partial definition) and C ≡ f(A, B, R, ⊓, ⊔, ∀, ∃, ⊥, ⊤) (full definition). ⊥ and ⊤ are two special atomic concepts named "bottom concept" and "universe concept". The syntax and semantics of $\mathcal{ALC}$ constructors have been represented in [4].

For the reason we mentioned above, classic DL such as $\mathcal{ALC}$ cannot deal with the imprecise description. To solve this problem in DLs, Straccia presented $\mathcal{FALC}$, which is an extension of $\mathcal{ALC}$ with fuzzy features, to support fuzzy concept representation. Because Straccia used a certain number to describe the fuzzy concepts and individuals in $\mathcal{FALC}$, we call the $\mathcal{FALC}$ type-1 $\mathcal{FALC}$ [7].

3 Type-2 Fuzzy $\mathcal{ALC}$

3.1 Imprecise Axioms in Type-2 Fuzzy $\mathcal{ALC}$

Different from the type-1 fuzzy sets, type-2 fuzzy sets use an interval to show the membership. Each grade of the membership is an uncertain number in interval [0,1]. We denote the membership in type-2 fuzzy sets with $\overline{\mu_A}$ instead of μ_A in type-1, which is defined as following:

$$\overline{\mu_A}(x) = [\mu_A^L(x), \mu_A^U(x)] \tag{1}$$

In (1) we present: $\mu_A^L(x), \mu_A^U(x) : U \rightarrow [0,1]$, and $\forall x \in U, \mu_A^L(x) \leq \mu_A^U(x)$. We call $\mu_A^L(x)$ and $\mu_A^U(x)$ the primary membership and secondary membership, and x is an instance in the fuzzy sets U. It is obvious that type-2 fuzzy sets can be reduced to type-1 fuzzy sets when the primary membership equals the secondary one. So a type-1 fuzzy set is embedded in a type-2 fuzzy set.

There are two fuzzy parts in type-2 fuzzy $\mathcal{ALC}$ presented in our paper, which are the imprecise terminological axiom (TBox) and fuzzy individual membership (ABox). To built a DL system, the first thing should be done in creating TBox is to define necessary atomic concepts and roles with some symbols. It is certainly that the base symbols exist in the DL system, but the name symbols are not. In other words, the atomic concepts defined by different axioms may be imprecise, which means that the axiom may not come into existence in type-2 fuzzy $\mathcal{ALC}$ TBox.

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

$$\text{Bird}_{[0.9,0.95]} \equiv \text{Animal} \sqcap \text{FlyingObject} \tag{2}$$

(2) means that the probability value of that bird can be described with the conjunction of the Animal and FlyingObject is between 0.90 and 0.95.

Because of the certainty of the base symbols, the probability of atomic concepts Animal and FlyingObject are both 1, in the interval [1,1]. Instead of Animal $_{[1,1]}$ we define the certain atomic concept Animal without [1,1] concisely.

Type-2 fuzzy $\mathcal{ALC}$ can represent the vagueness in atomic concept with two properties, *fuzzy:LowerDegree* and *fuzzy:UpperDegree* to describe $\mu_A^L(x)$ and $\mu_A^U(x)$.

Because it can be considered true that every atomic concept (role) is independent, we can calculate the value of *fuzzy:LowerDegree* and *fuzzy:UpperDegree* of fuzzy concept if we do not know it beforehand. For example, we want to define an atomic concept Meat-eatingBird with base symbol Meat-eatingObject with axiom (3):

$$\text{Meat-eatingBird} \equiv \text{Bird}_{[0.9,0.95]} \sqcap \text{Meat-eatingObject} \quad (3)$$

when we apply the triangular norms $T(a,b) = ab/[1+(1-a)(1-b)]$, $S(a,b) =(a+b)/(1+ab)$, we can get the value of *fuzzy:LowerDegree* (*fuzzy:UpperDegree*) of Meat-eatingBird with the follow equation: μ^L (Meat-eatingBird)=T (μ^L (Bird), μ^L (Meat-eatingObject)), as mentioned above, we know that μ^L (Bird)=0.9 and μ^L (Meat-eatingObject)=1. So μ^L (Meat-eatingBird)= (0.9×1)/[1+(1-1)(1-0.9)] = 0.9. So the membership of atomic concept Meat-eatingBird is in scope [0.9,0.95]. We call it transitivity of type-2 fuzzy $\mathcal{ALC}$.

In addition to the fuzzy TBox, the uncertainty still exists in ABox in fuzzy DLs. The assertion $\text{Bird}_{[0.9,0.95]}$(penguin) $_{[0.65,0.9]}$ means the degree that the penguin can be considered as an instant of $\text{Bird}_{[0.9,0.95]}$ is in [0.65,0.90] in a given DL. Being similar with $\mathcal{FALC}$, the ABox assertions $C^I(d)=[a, b]$, in which $0 \le a \le b \le 1$. Take atomic Bird $_{[0.9,0.95]}$ for example, The Bird(penguin) being satisfied in ABox has two pre-conditions: (1) concept Bird should be satisfied in TBox; (2) penguin belongs to bird in ABox. So we can conclude that μ^L (Bird(penguin))= μ^L (Bird)× μ^L (penguin $\in$ Bird) = T(0.65,0.90)=0.565 (so do μ^U (Bird(penguin))). So the ABox can be denoted by a set of equations with form as: $C_{[a,b]}$ (a)=[c, d] Where C=f(A, B, R, $\sqcap$, $\sqcup$, $\forall$, $\exists$, $\bot$, $\top$). For example: $\text{Bird}_{[0.9,0.95]}$(penguin)=[0.65,0.95], or $\text{Bird}_{[0.9,0.95]}(\text{penguin})_{[0.65,0.98]}$.

3.3 The Syntax and Semantics of Type-2 Fuzzy $\mathcal{ALC}$

We define $\mathcal{A}$, $\mathcal{C}$ and $\mathcal{R}$ as the set of atomic concepts, complex concepts, and roles. $C \sqcap D$, $C \sqcup D$, $\neg C$, $\forall R.C$ and $\exists R.C$ are fuzzy concept. The fuzzy interpretation in type-2 fuzzy $\mathcal{ALC}$ is a pair $I = (\Delta^I, \cdot^I)$, and $\cdot^I$ is an interpretation function that map fuzzy concept and role into a membership degree interval: $C^I = \Delta^I \to [a, b]$ and $R^I = \Delta^I \times \Delta^I \to [a, b]$, and a, b must satisfy $0 \le a \le b \le 1$. The syntax and semantic of type-2 fuzzy $\mathcal{ALC}$ is shown in Table 1. Different from $\mathcal{FALC}$ the type-2 fuzzy $\mathcal{ALC}$, Δ^I is not a set of numbers in scope [0,1] but a set of pairs, which have the form like [a, b]. And it must satisfy the inequation: $0 \le a \le b \le 1$.

Table 1. The syntax and semantics of type-2 fuzzy $\mathcal{ALC}$ 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	$\forall R_{[a,b]}.C_{[c,d]}$	$\forall y.S(R_{[1-b,1-a]}(x,y), C_{[c,d]}(y))$
Full existential quantification	$\exists R_{[a,b]}.C_{[c,d]}$	$\exists y.T(R_{[a,b]}(x,y), C_{[c,d]}(y))$

3.4 Reasoning in Type-2 Fuzzy $\mathcal{ALC}$

Tableau algorithms use negation to reduce subsumption to (un)satisfiability of concept descriptions instead of testing subsumption of concept descriptions directly: $C \sqsubseteq D$ iff $\neg C \sqcap D = \bot$. The fuzzy tableau begin with an ABox $A_0=\{C_{[a,b]}(\mathrm{x})_{[c,d]}\}$ to check the (un)satisfiability of concept $C_{[a,b]}$. Since $\mathcal{ALC}$ has not number restrictions, here are 5 rules presented:

$\cap$ *-rule*: if A contains $C_{[a,b]}(\mathrm{x})_{[c,d]}$, and $C_{[e,f]}(\mathrm{x})_{[g,h]}$: if $[a, b] \cap [e, f] \neq \Phi$ and $[c, d] \cap [g, h] \neq \Phi$ algorithm should extend A to $A' = A - \{C_{[a,b]}(\mathrm{x})_{[c,d]}, C_{[e,f]}(\mathrm{x})_{[g,h]}\} \sqcup \{C_{[S0(a,e),T0(b,f)]}(\mathrm{x})_{[S0(c,g),T0(d,h)]}\}$, else $A' = A - \{C_{[a,b]}(\mathrm{x})_{[c,d]}, C_{[e,f]}(\mathrm{x})_{[g,h]}\}$

$\sqcap$ *-rule*: if A contains $(C'_{[e,f]} \sqcap C''_{[g\,h]})_{[a,b]}(\mathrm{x})_{[c,d]} = (C' \sqcap C'')_{[T(T(e,f),a),T(T(g,h),b)]}(\mathrm{x})_{[c,d]}$, but not contains both $C'_{[e,f]}(\mathrm{x})_{[c,d]}$ and $C''_{[g,h]}(\mathrm{x})_{[c,d]}$, algorithm should extend A to $A' = A \sqcup \{C'_{[e,f]}(\mathrm{x})_{[c,d]}, C''_{[g,h]}(\mathrm{x})_{[c,d]}\}$.

$\sqcup$ *-rule*: if A contains $(C'_{[e,f]} \sqcup C''_{[g\,h]})_{[a,b]}(\mathrm{x})_{[c,d]} = (C' \sqcup C'')_{[S(S(e,f),a)\ S(S(g,h),b)]}(\mathrm{x})_{[c,d]}$, but neither $C'_{[e,f]}(\mathrm{x})_{[c,d]}$ nor $C''_{[g,h]}(\mathrm{x})_{[c,d]}$, the algorithm should extend A to $A' = A \sqcup \{C'_{[e,f]}(\mathrm{x})_{[c,d]}\}$ or $A'' = A \sqcup \{C''_{[g,h]}(\mathrm{x})_{[c,d]}\}$.

$\exists$ *-rule*: if A contains $(\exists R_{[e,f]}.C_{[g,h]})(\mathrm{x})_{[c,d]}$, but no individuals z such that $R_{[e,f]}(\mathrm{x}, \mathrm{z})_{[c,d]}$ and $C_{[g,h]}(\mathrm{z})_{[c,d]}$, the algorithm should extend A to $A' = A \sqcup \{R_{[e,f]}(\mathrm{x}, \mathrm{y})_{[c,d]}, C_{[g,h]}(\mathrm{y})_{[c,d]}\}$ where y is an individual not occurring in A before.

$\forall$*-rule*: if A contains $(\forall R_{[e,f]}.C_{[g,h]})(\mathrm{x})_{[c,d]}$, and $R_{[e,f]}(\mathrm{x}, \mathrm{y})_{[c,d]}$, but not $C_{[g,h]}(\mathrm{y})_{[c,d]}$, the algorithm should extend A to $A' = A \sqcup \{C_{[g,h]}(\mathrm{y})_{[c,d]}\}$.

Given two limit values: T_L and T_U, the way to decide whether the ABox in type-2 fuzzy $\mathcal{ALC}$ is unsatisfiable is different from typical tableau. In that

$\mu^{L(U)}(C) \leq T_L \Leftrightarrow C_{[0,0]}$, $\mu^{L(U)}(C) \geq T_U \Leftrightarrow C_{[1,1]}$. So the process of tableau will stop when anyone of following conditions is established:

(1) Any obvious clash ($\perp$(x), ($C \sqcap \neg C$)(x), etc.) is found in process of algorithm.

(2) All rules ($\sqcap$-*rule*, etc.) have been executed.

(3) Any fuzzy clash ($C_{[0,0]}$ (x)=[*c, d*],$C_{[a,b]}$ (x)=[*c, d*], $C_{[c,d]}$ (x)=[*a, b*]with $a \leq b \leq T_L$, $C_{[a,b]}$ (x) and $C_{[c,d]}$ (x) with the intervals [*a, b*] and [*c, d*] do not overlap) happened in process of algorithm.

4 The Semantic Search Engine Based on Type-2 Fuzzy Ontology

4.1 Architecture of Type-2 Fuzzy Semantic Search Engine

The natural languages in daily communication often have imprecise information. We call the queries including fuzzy concepts fuzzy queries. To handle these fuzzy queries, semantic search engines based on ontology must extend their knowledge bases on fuzzy ontologies such as the fuzzy semantic search engine proposed in this paper. Fig. 1 shows the architecture of type-2 fuzzy semantic search engine.

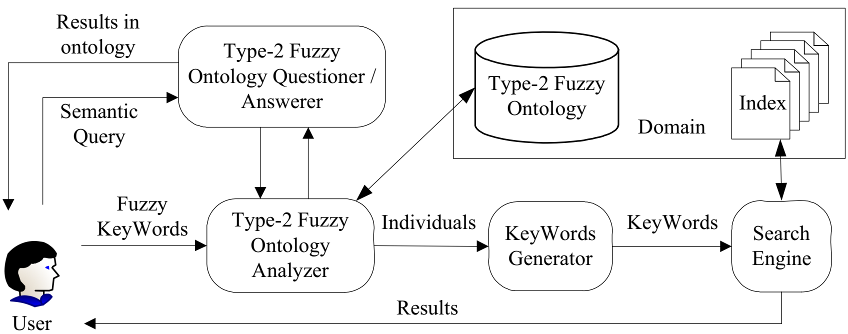

Fig. 1. Architecture of Type-2 fuzzy semantic search engine

In this framework, users can propose their query in two ways: they can ask the type-2 fuzzy ontology analyzer with keywords or fuzzy keywords. On the other hand, users can also search the ontology by issuing the semantic query to type-2 fuzzy ontology questioner (answerer) with keywords or other interfaces. Thus, users can communicate with ontology directly with the recalls formed by individuals or classes to make the queries precisely, which are sent to type-2 fuzzy ontology analyzer later. Then type-2 fuzzy ontology analyzer can generate individuals that satisfy to query and send these answers to keywords generator to combine proper keywords. At last, the traditional search engine will find the results from index with these keywords and return the hits to users.

4.2 Experiments and Analysis

Based on the framework introduced above, we have implemented the type-2 fuzzy search engine. Supported by the fuzzy ontology reasoner, the semantic search engine based on type-2 fuzzy $\mathcal{ALC}$ can improve the relativity of the responses to query. The experiment is carried out in the scope of all resources available in Huazhong University of Science and Technology, including almost 7000 web pages indexed from different departments and 2400 documents. The type-2 fuzzy ontology analyzer, answer, keywords generator and the search engine are all implemented with java. The ontology has built with protégé.

We chose a group of keywords to retrieve information from indexes, and then picked out the relevant hits (hits those are relevant to the retrieval) from result set and counted the average of them. Fig. 2 shows that the semantic search engine based on ontology (including classic and fuzzy ontology) can expand the relevant hits greatly when there is no imprecise information in keywords. The reason is that ontology generates more keywords with its individuals. However, the number of relevant hits of search engine based on classic ontology decreases rapidly when we add more fuzzy keywords such as “very”, “young” into the keywords group. Compared to classic ontology, semantic search engine based on type-2 fuzzy ontology can accommodate itself to fuzzy keywords much better. We carry out an experiment on the precision (the fraction of the retrieved documents which is relevant) of the semantic search engine for that reason. Fig. 3 represents that the precision of semantic search engine based on classic ontology increases slower than the one based on type-2 fuzzy ontology when the number of nodes increases in otology. That means the precision of search engine will be improved if type-2 fuzzy $\mathcal{ALC}$ is applied.

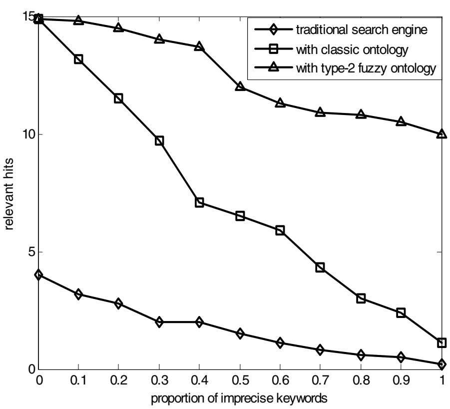

Fig. 2. Relevant hits -imprecision graph

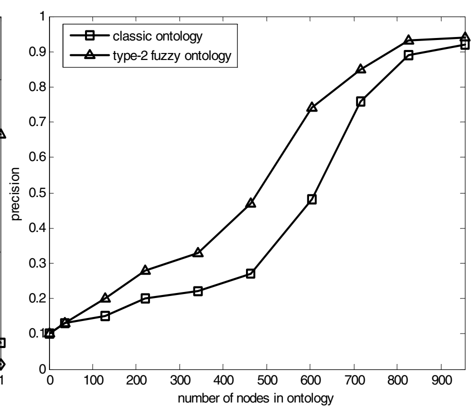

Fig. 3. Precision-nodes graph

5 Conclusions and Future Work

As the fundament of type-2 fuzzy DLs, type-2 $\mathcal{ALC}$ is introduced of its syntax, semantics, reasoning algorithm and application in this paper. Comparing with the type-1 fuzzy

$\mathcal{ALC}$, the type-2 fuzzy $\mathcal{ALC}$ can deal with the imprecise knowledge much better. Besides semantic search, there are many applications based on DLs need to handle fuzzy information such as trust management. Our approach can be applied in those domains to enrich its representation meaning and reasoning abilities. Future work includes the research of type-2 fuzzy $\mathcal{ALCN}$, $\mathcal{SHOIN(D)}$ and the reasoning algorithms.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. The Scientific American 284(5) (2001) 34-43
2. Horrocks, I., Patel-Schneider, P.F., van Harmelen, F.: From SHIQ and RDF to OWL: The Making of a Web Ontology Language. Journal of Web Semantics 1(1) (2003) 7-26
3. Guha, R., McCool, R., Miller, E.: Semantic Search. In: Proceedings of the 12th International World Wide Web Conference (WWW 2003). Budapest, Hungary (2003) 700-709
4. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F.: The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003) 47-100
5. Calvanese, D., Lenzerini, M., Nardi, D.: Unifying Class-Based Representation Formalisms. Journal of Artificial Intelligence Research 11(2) (1999) 199-240
6. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.L., Patel-Schneider, P.F. (ed.): L.A.S.: OWL Web Ontology Language Reference (2004)
7. Heflin, J.D.: Towards the Semantic Web: Knowledge Representation in a Dynamic Distributed Environment. PhD Thesis, University of Maryland (2001)
8. Heflin, J., Hendler, J.: Searching the Web with Shoe. In: AAAI-2000 Workshop on AI for Web Search. Austin, Texas, USA (2000)
9. Guha, R., McCool, R.: TAP: A Semantic Web Test-bed. Journal of Web Semantics 1(1) (2003) 32-42
10. Guha, R., McCool, R.: The Tap Knowledge Base. http://tap.stanford.edu/
11. Guha, R., McCool, R.: Tap: Towards a Web of Data. http://tap.stanford.edu/
12. Ding, L., Finin, T., Joshi, A., et al.: Swoogle: A Search and Metadata Engine for the Semantic Web. In: CIKM'04. Washington DC, USA (2004)
13. Finin, T., Mayfield, J., Joshi, A., et al.: Information Retrieval and the Semantic Web. In: Proceedings of the 38th Hawaii International Conference on System Sciences (2005)
14. Mayfield, J. Finin, T.: Information Retrieval on the Semantic Web: Integrating Inference and Retrieval. In: 2004 SIGIR Workshop on the Semantic Web. Toronto (2004)
15. Zadeh, L. A.: Fuzzy Sets. Information and Control 8(3) (1965) 338-353
16. 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. California (1997) 89-94
17. Straccia, U.: Reasoning within Fuzzy Description Logics. Journal of Artificial Intelligence Research 14 (2001) 137-166
18. Straccia, U.: Transforming Fuzzy Description Logics into Classical Description Logics. In: Proceedings of the 9th European Conference on Logics in Artificial Intelligence. Lisbon, (2004) 385-399
19. 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