Modeling Fuzzy Rules with Description Logics

Sudhir Agarwal and Pascal Hitzler

Institute of Applied Informatics and Formal Description Methods (AIFB),
University of Karlsruhe (TH), Germany.
{agarwal,hitzler}@aifb.uni-karlsruhe.de

Abstract. In real application scenarios, input data and knowledge is often vague. Likewise, it is often the case that exact reasoning over data is impossible due to complex dependencies between input data and target outputs. For practical applications, however, good approximations often suffice, and efficient calculation of an approximate answer is often preferable over complex processing which may take a long time to come up with an exact answer. Fuzzy logic supports both features by providing fuzzy membership functions and fuzzy IF-THEN rule bases. In this paper, we show how fuzzy membership functions and fuzzy rules can be modeled by means of an appropriate description logic and how this can be employed for query answering.

1 Introduction

In the last decade a substantial amount of work has been carried out in the context of Description Logics (DLs)[1]. Despite their growing popularity, relative little work has been done in extending them to the management of vague information. In many application domains, not only the membership of an individual to a set is nonrigid, but also the transition between the memberships of an individual from one set to another is smooth. Consider, for example, *height* of a human. Small children grow, but when do they stop to be small? So the transition from short humans to humans of average height is rather smooth and not crisp. Such kinds of knowledge can be encoded using techniques from fuzzy logic.

Vague knowledge, i.e. rules based on fuzzy logic, are also important from the perspective of evaluating values of attributes that have very complex dependencies with other attribute values. Such rules play an important role in application domains, where a good approximation of the desired value of an attribute is acceptable. For example, consider the controlling of a train. It is desired, that when a train arrives at station, it halts at a certain fixed position. However, calculating how exactly the brake should be applied at what position in which speed so that the passengers can still sit comfortable etc. is difficult. However, considering that it is acceptable if the train stops a small distance before or after the mark, automatic control of the train is much easier.

Fuzzy logic, first introduced by Zadeh in [2, 3], provides answers to both the problems. On the one hand, the fuzzy sets allow to model vague memberships of

individuals to sets. On the other hand, fuzzy IF-THEN rules allow to evaluate good approximations of desired attribute values in a very efficient way [2, 3].

In this paper, we will show how fuzzy IF-THEN rules can be modeled using description logics with concrete domains and aggregates. The exposition serves two purposes. On the one hand, it serves as a modeling example for description logics and the usefulness of concrete domains and aggregates. On the other hand, it provides the first steps toward the implementation of fuzzy rules systems with the description logic knowledge representation paradigm. Due to the advent of the semantic web, this can be seen as a step toward the sharing of sophisticated structured knowledge by means of ontology languages.

The paper is structured as follows. First, we give a short introduction to the description logic with concrete domains and aggregates in section 2. In section 3, we show how fuzzy membership functions can be modeled and how membership can be calculated. In section 4, we show how fuzzy rules can be modeled and how the degree of fulfillment of a fuzzy rule can be calculated. In section 5, we show how the modeled fuzzy rules can be used for answering DL queries. Finally, we conclude in section 7 after discussing some related work in section 6.

2 Description Logics with Aggregates and Concrete Domains

Description logics with concrete domains was first introduced in [4]. It was then extended by aggregates in [5, 6].

Definition 1. *A concrete domain* $\mathcal{D} = (dom(\mathcal{D}), pred(\mathcal{D}))$ *consists of a set* $dom(\mathcal{D})$ *(the domain), and a set of predicate symbols* $pred(\mathcal{D})$*. Each predicate symbol* $P \in pred(\mathcal{D})$ *is associated with an arity* n *and an* n*-ary relation* $P^{\mathcal{D}} \subseteq dom(\mathcal{D})^n$*.*

Definition 2. *Let* N_C*,* N_R *and* N_F *be disjoint sets of concept, role and feature*[1] *names. The set of* $\mathcal{ALC}(\mathcal{D})$*-concepts is the smallest set such that (1) every concept name is a concept and (2) if* C *and* D *are concepts,* R *is a role or a feature name,* $P \in pred(\mathcal{D})$ *is an* n*-ary predicate name, and* $u_1, \ldots, u_n$ *are feature chains*[2]*, then* $(C \sqcap D)$*,* $(C \sqcup D)$*,* $(\neg C)$*,* $(\forall R.C)$*,* $(\exists R.C)$ *and* $P(u_1, \ldots, u_n)$ *are concepts.*

Definition 3. *The notion of a concrete domain* $\mathcal{D}$ *as introduced in Definition 1 is extended by a set of aggregation functions* $agg(\mathcal{D})$*, where each* $\Gamma \in agg(D)$ *is associated with a partial function* $\Gamma^{\mathcal{D}}$ *from the set of multisets of* $dom(\mathcal{D})$ *into* $dom(\mathcal{D})$*. Such an extended concrete domain is denoted by* Σ*.*

The set of concrete features is inductively defined as follows. (1) Each feature name $f \in N_F$ *is a concrete feature. (2) a feature chain* $f_1 \ldots f_n$ *is a concrete features, and (3) an aggregated feature* $f_1 \ldots f_n \Gamma(R \circ f)$ *is a concrete feature, where* $f, f_1 \ldots f_n$ *are feature names, and* $\Gamma \in agg(\Sigma)$ *is an aggregation function.*

[1] *Features* are just functional roles.

[2] A *feature chain* [4] u is a chain of functional roles, e.g. (*father age*).

Finally, $\mathcal{ALC}(\Sigma)$ concepts are obtained from $\mathcal{ALC}(\mathcal{D})$-concepts by allowing additionally the use of concrete features f_i in predicated restrictions $P(f_1 \ldots f_n)$.

Note, that the description logic $\mathcal{ALC}(\Sigma)$ is not decidable. For details about the semantics of $\mathcal{ALC}(\mathcal{D})$ and $\mathcal{ALC}(\Sigma)$, and proof of undecidability of $\mathcal{ALC}(\Sigma)$, we refer to [4–6].

3 Fuzzy Membership Functions

One of the biggest advantages of fuzzy logic is that fuzzy rules appeal to human intuition, as they contain linguistic variables and linguistic terms like *temperature = cold* rather than precise values like *temperature* $= 5°C$. A fuzzy rule engine mimics human reasoning as it works on such linguistic variables and linguistic terms. Formally, a linguistic term is a membership function that maps each possible value of the linguistic variable to a real number between 0 and 1. Each linguistic term of a linguistic variable covers a range of possible values which the linguistic variable can take, and the real advantage of fuzzy inferencing lies in the smooth transition between linguistic terms covering adjacent value ranges.

3.1 Modeling Fuzzy Membership Functions

Figure 1 shows the linguistic terms *cold*, *comfortable* and *warm* modeled as membership functions for a linguistic variable *Temperature*.

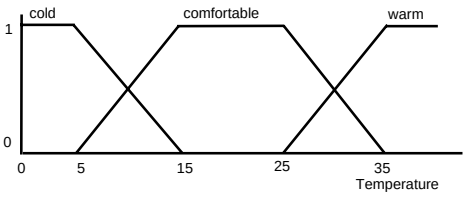

Fig. 1. Example Membership Functions

Now let $\mathbb{R}_{[0,1]}$ denote the set of real numbers between 0 and 1. For a concept v and a linguistic term t, we define a membership function μ_t^v as a finite and non-empty set of points[3] (x, y) in $\mathbb{R} \times \mathbb{R}_{[0,1]}$, where x is a number representing[4] an individual of the concept v. We will define next a concept *Point*. For this purpose, we introduce two concrete functional roles x and y, which assign the first respectively second coordinate to the point. So we define the concept *Point* as

$$Point \sqsubseteq \exists x.\mathbb{R} \sqcap \exists y.\mathbb{R}_{[0,1]}.$$

Similarly, we define a concept μ that denotes the set of membership functions. We do this by means of a (non-functional) role p which assigns points to individuals, as follows:[5]

$$\mu \sqsubseteq \exists p.Point.$$

[3] I.e. we allow only membership functions which are piecewise linear.

[4] Identifying individuals with real numbers simply serves to make computations simpler, though it may appear to be counterintuitive in some cases.

[5] $\mu \sqsubseteq \geq 2\, p.Point$ would be more precise if qualified number restrictions are available.

For a concept v, being viewed as a linguistic variable and having linguistic terms $t_1, \ldots, t_n$, we add n instances $\mu^v_{t_1}, \ldots, \mu^v_{t_n}$ of μ with corresponding roles and points. We interpret the set of points associated with some $\mu^v_{t_i}$ as a piecewise linear function. For a concept v being viewed as a linguistic variable, we denote the set of its linguistic terms by v^*.

3.2 Calculating the Membership to a Fuzzy Set

Our membership functions are just sets of points in $\mathbb{R}^2$. Two points with adjacent x-coordinates can be interpreted as a straight line. For a line between (x_1, y_1) and (x_2, y_2), and a given x, we calculate y as[6]

$$y = \begin{cases} \frac{y_1 - y_2}{x_1 - x_2}(x - x_1) + y_1, \text{ if } x_1 \leq x < x_2, \\ 0, \text{ otherwise.} \end{cases} \tag{1}$$

We model this by defining a concept YL, to capture the ys as described above.

$$YL \sqsubseteq \exists y.\mathbb{R}_{[0,1]}$$

Now suppose that there are k membership functions for a concept v. We define k relations from the concept v to the concept YL as follows:

$$v \sqsubseteq \bigsqcap_{i \in \{1,\ldots,k\}} \exists yl_{\mu^v_{t_i}} . YL$$

An instance of a concept v will be in as many relation instances of $yl_{\mu^v_{t_i}}$ with instances of YL as there are lines in the membership function $\mu^v_{t_i}$. The membership of an instance of the concept v to a membership function $\mu^v_{t_i}$ is then just the sum of all such ys over all the lines of $\mu^v_{t_i}$. To capture this information, we define a functional role m_{μ_i} from v to $\mathbb{R}_{[0,1]}$ as follows

$$v \sqsubseteq \bigsqcap_{i \in \{1,\ldots,k\}} \exists m_{\mu^v_{t_i}} . \mathbb{R}_{[0,1]}.$$

Finally, to ensure that the membership of an instance of the concept v via a membership function $\mu^v_{t_i}$ is equal to the sum of all ys over all the lines of $\mu^v_{t_i}$, we set

$$v \sqsubseteq \bigsqcap_{i \in \{1,\ldots,k\}} P_{=}(m_{\mu^v_{t_i}}, sum\{yl_{\mu^v_{t_i}} \circ y\}) \tag{2}$$

where the predicate $P_{=}(x, y)$ is true iff $x = y$, and *sum* is the aggregate function which computes the sum of its inputs.

Figure 2 shows the calculation of the membership of a concrete temperature value *12* to the membership functions *cold* and *comfortable*. The membership

[6] Note, the condition $x < x_2$ (instead of $x \leq x_2$) ensures that a given x does not lie on more than one line.

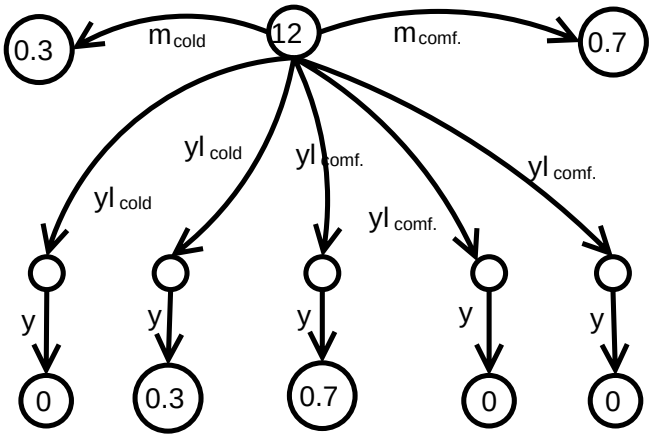

Fig. 2. Membership Calculation

function *cold* consists of three points (i.e. two lines) and the membership function *comfortable* consists of four points (i.e. three lines) (cf. figure 1). So, the instance *12* has two relations of type yl_{cold} and three relations of type $yl_{comfortable}$ to the instances of YL.[7] The values connected to an instance of YL via the role y are calculated according to equation 1. Now, the memberships of the instance *12* via membership functions m_{cold} and $m_{comfortable}$ are calculated according to the equation in axiom 2, i.e. $m_{cold} = 0{+}0.3 = 0.3$ and $m_{comfortable} = 0.7{+}0{+}0 = 0.7$.

4 Fuzzy Rules

A *fuzzy IF-THEN rule* consists of an IF part (antecedent) and a THEN part (consequent). The antecedent is a combination of terms, whereas the consequent is exactly one term. In the antecedent, the terms can be combined by using fuzzy conjunction, disjunction and negation. A *term* is an expression of the form $X = T$, where X is a linguistic variable and T is one of its linguistic terms.

4.1 Modeling Fuzzy Rules

Since terms are the elementary building blocks of a fuzzy rule, we start with modeling terms. As described above, a term consists of two parts, a linguistic variable and a linguistic term. So, we model a concept *Term* as

$$Term \sqsubseteq \exists r.\top \sqcap \exists f.\mu,$$

where the roles r and m are functional roles assigning linguistic variable resp. linguistic term.

[7] To save space, we abbreviate *comfortable* by *comf.* and do not show the structure related to the membership function *warm*.

Terms can be combined via conjunction, disjunction and negation to term expressions. Further, a term expression is fulfilled by an individual to a certain degree. So, we define concepts $\mathit{TermExp}$, $\mathit{TermExp}_\wedge$, $\mathit{TermExp}_\vee$ and $\mathit{TermExp}_\neg$ and extend the definition of the concept Term as follows, introducing also the corresponding roles.

$$\begin{aligned}
\mathit{TermExp} &\sqsubseteq \exists \mathit{degree}.\mathbb{R}_{[0,1]} \\
\mathit{TermExp}_\wedge &\sqsubseteq \mathit{TermExp} \sqcap \exists \mathit{conjunct}.\mathit{TermExp} \\
\mathit{TermExp}_\vee &\sqsubseteq \mathit{TermExp} \sqcap \exists \mathit{disjunct}.\mathit{TermExp} \\
\mathit{TermExp}_\neg &\sqsubseteq \mathit{TermExp} \sqcap \exists \mathit{operand}.\mathit{TermExp} \\
\mathit{Term} &\sqsubseteq \mathit{TermExp}
\end{aligned}$$

Note that the relation *operand* is a function role.

A rule has an antecedent and a consequent. The antecedent is a term expression and the consequent is a term. Further, a rule has a degree to which it is fulfilled by an individual. So, we define a concept *Rule* as

$$\mathit{Rule} \sqsubseteq \exists \mathit{antecedent}.\mathit{TermExp} \sqcap \exists \mathit{consequent}.\mathit{Term} \sqcap \exists \mathit{degree}.\mathbb{R}_{[0,1]}.$$

The roles *antecedent* and *consequent* are functional roles.

4.2 Calculating the Degree of Fulfillment of a Rule

Since terms are the basic building blocks of a rule, the degree of fulfillment of a rule depends ultimately on the degrees of fulfillment of the terms occurring in the rule. Now, an individual connected to a term via the role r fulfills the term with the same degree as the corresponding value of the membership function the term is connected with via the role m. We model this by extending the definition of the concept *Term* as follows:

$$\mathit{Term} \sqsubseteq P_{=}(\mathit{degree}, r \circ m_f).$$

We can calculate the degree of fulfillment of a term expression according to the semantics suggested by Zadeh in [2, 3], which can be summarized as follows.[8] Given two membership functions μ_A and μ_B,

$$\begin{aligned}
(\mu_A \wedge \mu_B)(a) &= \min\{\mu_A(a), \mu_B(a)\} \\
(\mu_A \vee \mu_B)(a) &= \max\{\mu_A(a), \mu_B(a)\} \\
(\neg\mu_A)(a) &= 1 - \mu_A(a)
\end{aligned}$$

So, we extend the definitions of $\mathit{TermExp}_\wedge$, $\mathit{TermExp}_\vee$ and $\mathit{TermExp}_\neg$ as follows.

$$\begin{aligned}
\mathit{TermExp}_\wedge &\sqsubseteq P_{=}(\mathit{degree}, \min\{\mathit{conjunct} \circ \mathit{degree}\}) \\
\mathit{TermExp}_\vee &\sqsubseteq P_{=}(\mathit{degree}, \max\{\mathit{disjunct} \circ \mathit{degree}\}) \\
\mathit{TermExp}_\neg &\sqsubseteq P_{=1-}(\mathit{degree}, \mathit{operand} \circ \mathit{degree})
\end{aligned}$$

[8] Certainly, other T-norms and T-conorms could be used.

The predicate $P_{=1-}(a, b)$ is true iff $a = 1 - b$. min and max are aggregate functions for the concrete domain $\mathbb{R}$.

To interpret a fuzzy IF-THEN rule, we need an interpretation for the implication. In general, one can have a different interpretation of the implication for every rule, which is particularly important when the application domain requires the use of weighted rules. Here, we use a universal interpretation π of the implication in all the rules. However, we do not fix π any further. In most of the cases, it is equal to minimum. So

$$\mathit{Rule} \sqsubseteq P_\pi(\mathit{degree}, \mathit{antecedent} \circ \mathit{degree}, \mathit{consequent} \circ \mathit{degree}),$$

where P_π is a ternary predicate from the concrete domain $\mathbb{R}$ and represents the interpretation of the implication function π. That is, for given $a, b, c \in \mathbb{R}$, $P_\pi(a, b, c)$ is true iff $a = \pi(b, c)$.

5 Query Answering

So far, we have shown how a fuzzy rule can be interpreted. However, for answering queries that use fuzzy rules, we need an interpretation of fuzzy rules bases, i.e. of sets of fuzzy rules. Basically, there are two possibilities to interpret a fuzzy rule base, namely FITA (First Inferencing Then Aggregation) and FATI (First Aggregation Then Inferencing). In this paper, we will only model FITA and believe that FATI can be modeled similarly. It has been shown in [7] that the two principles are equivalent under certain conditions.

5.1 FITA

Consider a rule base containing n rules of the form $F_1 \rightarrow G_1, \ldots F_n \rightarrow G_n$. In FITA, first each rule is interpreted. That is, for each x, y and each rule i, the value of $\pi(F_i(x), G_i(y))$ is calculated. Now, for a given x, the inference step is performed for each rule. Again, the inference operator can be different for different rules. However, we use a universal inference operator for all the rules and call it κ — in many practical cases, κ is equal to minimum. In general, the inference operator κ is some function that maps the square $[0, 1]^2$ to $[0, 1]$. Performing an inference step for a given x and a given rule i means calculating $\kappa(F(x), \pi(F_i(x), G_i(y)))$, where F is some fuzzy set describing the membership function for a given situation. Having performed the inferencing step for all the n rules, an aggregation step is performed to obtain a single value from n values. For this purpose, an aggregation operator $\alpha : [0, 1]^n \rightarrow [0, 1]$ is needed. The most common aggregation operator is maximum. However, we do not fix α any further. In the aggregation step,

$$\alpha(\kappa(F(x), \pi(F_1(x), G_1(y))), \ldots \kappa(F(x), \pi(F_n(x), G_n(y))))$$

is calculated. We define a concept *FITA* as:

$$FITA \sqsubseteq \prod_{i\in\{1,\ldots,n\}} \exists rule_i.Rule \sqcap \prod_{i\in\{1,\ldots,n\}} \exists x_i.\mathbb{R}_{[0,1]} \sqcap \exists output.\mathbb{R}_{[0,1]} \sqcap \prod_{i\in\{1,\ldots,n\}} P_\kappa(x_i, rule_i \circ degree, x \circ m_f) \sqcap P_=(output, \alpha\{x_i\})$$

where α is an aggregate function and P_κ is a ternary predicate on the concrete domain $\mathbb{R}_{[0,1]}$. $P_\kappa(a, b, c)$ is true iff $a = \kappa(b, c)$.

5.2 Defuzzification

The goal of a DL query is to determine the value of an instance. FITA delivers the membership of an arbitrary instance of the target concept to the goal fuzzy concept according to the compositional rule of inference. That is, if we have a sufficient number of instances of the target concept, we can calculate for each instance its membership to the goal fuzzy concept. This way, we obtain a set of points $(x, \mu_T(x))$, where x is an arbitrary instance of the target concept and μ_T is the target fuzzy concept.

However, the goal of query answering is to determine an instance of the target concept. This is done by interpreting the set of points $(x, \mu_T(x))$ as an area in $\mathbb{R}^2$ and defuzzifying this area. One of the most common defuzzification methods is the so called *center of gravity* method, where the geometrical center of gravity of a given area is calculated. The desired instance is then equal to the value of the x-coordinate of the center of gravity of the area. Hence, the desired instance ω can be calculated by the following formula:

$$\omega = \frac{\int x \cdot \mu(x)dx}{\int \mu(x)dx} \tag{3}$$

To model the defuzzification process, we define a concept *DefuzInfo* as follows:

$$DefuzInfo \sqsubseteq \exists fita.FITA \sqcap \exists x \sqcap \exists prod.\mathbb{R} \sqcap P_{mul}(prod, x, fita \circ output).$$

Finally, consider a concept $C \sqsubseteq \exists w.W$. For a given instance θ of C, to determine an instance ω such that $w(\theta, \omega)$ holds while considering fuzzy rules, we extend the definition of the concept C as follows:

$$C \sqsubseteq \exists di.DefuzInfo \sqcap P_=(w, x) \sqcap P_{div}(x, sum\{di \circ prod\}, sum\{di \circ fita \circ output\}).$$

We use the already existing instances of the concept W as the arbitrary instances for determining the area that is defuzzified. If no instances of W are available in the knowledge base, we can always insert some instances which do not need to be in any relation with instances of other concepts. The number and value of such instances depends on the application domain, more precisely on the width of the range (subset of $\mathbb{R}$) and on the value of dx in equation 3.

6 Related Work

Although combining fuzzy logic with description logics has gained some interest recently, not much work has been done in this field yet. We mention some of the work which we consider most important.

[8] presents a fuzzy extension to the description logic $\mathcal{ALC}$. The resulting so-called fuzzy DL enables reasoning in the presence of imprecise ALC concepts. From a semantic perspective, fuzzy concepts are interpreted as fuzzy sets i.e. given a concept C and an individual a, $C(a)$ is interpreted as the truth-value of the sentence "a is C". From a syntax perspective, specification of lower and upper bounds of the truth-value of $C(a)$ is allowed. Further, algorithms for solving the entailment problem, the subsumption problem as well as the best truth-value bound problem are presented. The main difference between [8] and our work is the focus. [8] is a fuzzy logic extension of description logics, while we aim at the encoding of and dealing with fuzzy rules systems using description logics. [8] proposes a method of modeling and reasoning about fuzzy knowledge bases, whereas we construct fuzzy knowledge base on the fly to answer crisp queries while using fuzzy rules. Similar considerations concern [9–12].

[13] proposes f-SWRL, a fuzzy extension of SWRL. However, the syntax and semantics of f-SWRL allow to use weights between 0 and 1 for the atoms in a rule. So, there is actually no fuzziness in f-SWRL rules.

Finally, we believe that our work is different from the existing work dealing with combination of fuzzy logic with description logics at a very fundamental level. The existing approaches maintain a fuzzy knowledge base and membership information of fuzzy A-Box assertions must be provided explicitly. In our approach, we model fuzzy membership functions and calculate the membership of an individual to membership functions inside the reasoner. Hence, we do not require extra syntax for supporting fuzzy reasoning.

7 Conclusion and Outlook

In this paper, we have shown how fuzzy membership functions and fuzzy IF-THEN rules can be modeled with description logics that support the concrete domain $\mathbb{R}$ and simple aggregate functions like min, max, *sum* etc. Thus, we have presented a technique that enables easier modeling of complex dependencies by using fuzzy IF-THEN rules than by using only crisp DL rules. Our modeling technique allows to calculate a good approximate answer to a query inside a crisp DL reasoner. Hence, we have integrated and used fuzzy knowledge in a crisp knowledge base.

In the future, we intend to implement and evaluate our approach in a concrete scenario. Our aim is to compute and provide fuzzy membership values as modeled in section 3.2 at runtime in a lazy fashion instead of precomputing the corresponding ABox entries a priori. A good candidate application could be an extension by auctions of the semantic matchmaking portal presented in [14].

Acknowledgments

The authors acknowledge support by the German Federal Ministry of Education and Research (BMBF) under the SmartWeb and Internetökonomie projects, and by the European Commission under contract IST-2003-506826 SEKT and under the KnowledgeWeb Network of Excellence.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory Implemenation and Applications. Cambridge University Press (2003)
2. Zadeh, L.A.: Fuzzy Sets. Information and Control **8** (1965) 338–353
3. Yager, R.R., Ovchinnikov, S., Tong, R.M., Nguyen, H.T., eds.: Fuzzy sets and applications - Selected Papers by L. A. Zadeh. John Wiley & Sons, New York, NY, USA (1987)
4. Baader, F., Hanschke, P.: A Schema for Integrating Concrete Domains into Concept Languages. In: Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91), Sydney (1991) 452–457
5. Baader, F., Sattler, U.: Description Logics with Concrete Domains and Aggregation. In Prade, H., ed.: Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98), John Wiley & Sons Ltd (1998) 336–340
6. Baader, F., Sattler, U.: Description Logics with Aggregates and Concrete Domains. Information Systems **28** (2003) 979–1004
7. Temme, K.H., Thiele, H.: On the correctness of the principles of FATI and FITA and their equivalence. In: IFSA 95 - Sixth International Fuzzy Systems Association World Congress. Volume II. (1995) 475–478
8. Straccia, U.: Reasoning within fuzzy description logics. J. Artif. Intell. Res. **14** (2001) 137–166
9. Straccia, U.: Transforming Fuzzy Description Logics into Classical Description Logics. In: Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA-04). Number 3229 in Lecture Notes in Computer Science, Lisbon, Portugal, Springer Verlag (2004) 385–399
10. Straccia, U.: Towards a Fuzzy Description Logic for the Semantic Web (Preliminary Report). In: 2nd European Semantic Web Conference (ESWC-05). Number 3532 in Lecture Notes in Computer Science, Crete, Springer Verlag (2005) 167–181
11. Straccia, U.: Description Logics with Fuzzy Concrete Domains. In Bachus, F., Jaakkola, T., eds.: 21st Conference on Uncertainty in Artificial Intelligence (UAI-05), Edinburgh, Scotland, AUAI Press (2005) 559–567
12. Straccia, U.: A fuzzy description logic for the semantic web. In Sanchez, E., ed.: Capturing Intelligence: Fuzzy Logic and the Semantic Web. Elsevier (2005)
13. Pan, J.Z., Stamou, G., Tzouvaras, V., Horrocks, I.: f-SWRL: A Fuzzy Extension of SWRL. In: Proc. of the International Conference on Artificial Neural Networks (ICANN 2005), Special section on "Intelligent multimedia and semantics". (2005)
14. Agarwal, S., Lamparter, S.: sMart - A Semantic Matchmaking Portal for Electronic Markets. In: Proceedings of the 7th International IEEE Conference on E-Commerce Technology 2005, Munich, Germany, IEEE Computer Society (2005)