Expressive Reasoning with Horn Rules and Fuzzy Description Logics

Theofilos Mailis, Giorgos Stoilos, and Giorgos Stamou

Department of Electrical and Computer Engineering, National Technical University of Athens, Zographou 15780, Greece

Abstract. This essay describes fuzzy CARIN, a knowledge representation language combining fuzzy description logics with Horn rules. Fuzzy CARIN integrates the management of fuzzy logic into the non-recursive CARIN system. It provides a sound and complete algorithm for representing and reasoning about fuzzy $\mathcal{ALCNR}$ extended with non-recursive Horn rules. Such an extension is most useful in realistic applications dealing with uncertainty and imprecision, such as multimedia processing and medical applications. Additionally, it provides the ability of answering to union of conjunctive queries, which is a novelty not previously addressed by fuzzy DL systems.

1 Introduction

Over the last two decades fragments of first order logic, called Description Logics (DLs) [1], have been brought into focus by the Artificial Intelligence community. DLs well formed semantics and great expressivity has enforced their utilization in numerous domains, such as multimedia [2–4] and medical [5] applications, as knowledge representation and reasoning languages. More importantly DLs provide the formal foundation for the standard web ontology language OWL [6] which is a milestone for the Semantic Web [7].

DLs main asset, their class-based knowledge representation formalism, also sets a limit to their expressive power as they are incapable of providing complex descriptions about role predicates. Expressive DLs such as $\mathcal{SHOIQ}$ are incapable of expressing even a simple composition between roles[1]. For this reason, as visualized in the Semantic Web stack diagram[2], there is a need for integrating DLs with rules. A natural choice for such integration would be classes of rule languages originating from logic programming and non-monotonic reasoning [10].

In [10], the "cream" of systems combining rules and DLs is presented. Systems such as DLP [11], SWRL [12], $\mathcal{AL}$-log [13], F-logic [14] and CARIN [15] present different approximations for intergrading DLs with rules. These are divided into the hybrid systems, where there is a distinction between the predicates in the rule and the DL part, and the homogeneous where there is no such distinction. CARIN is such an hybrid system that combines the DL $\mathcal{ALCNR}$ with

[1] Recent systems such as EL^{++} [8], $SROIQ$ [9] are such extensions

[2] `http://www.w3.org/2003/Talks/05-gartner-tbl/slide29-0.html`

Horn rules and through its existential entailment algorithm *offers a sound and complete inference procedure for non-recursive knowledge bases, can answer to arbitrary conjunctive queries and provides an algorithm for rule subsumption over* $\mathcal{ALCNR}$ [15].

Though CARIN offers great expressivity in order to represent a fragment of our universe, it is incapable of encoding knowledge with some degree of uncertainty and imprecision. Uncertainty emerges from our lack of knowledge about a certain fact e.g. we assume that the black dot in the background of a picture is a lion, while imprecision refers to the intrinsic inability to strictly classify a fact or a state of an object e.g. a half-empty glass of water can neither be characterized as full, nor as empty.

Fuzzy logic is a mean to represent knowledge containing uncertainty and imprecision. Several systems, such as fuzzy $\mathcal{ALC}$ [16], fuzzy $f_{KD} - \mathcal{SI}$ [4], f_{KD}-$\mathcal{SHIN}$ [17], have been proposed for combining fuzzy logic with description logics. Based on these systems we propose fuzzy CARIN, which is an extension of non-recursive CARIN, in order to represent uncertainty and imprecision. Related work combining DLs with Rules has been presented in [18, 19], providing fuzzy extensions of DL programs [20].

The need for fuzzy extensions of systems combining DLs with rules is most obvious in multimedia applications:

Example 1. Suppose that we have a, rather "optimistic", algorithm for object recognition. This algorithm is divided into an image processing and a DL extended with rules part. Assume it contains the following rules and implications:

$$leaf(x) \wedge nextTo(x, y) \wedge trunk(y) \Rightarrow tree(x, y)$$
$$\exists hascolor.green \sqcup \exists hascolor.yellow \sqsubseteq leafs \ldots$$

The algorithm implies that a tree is an object consisting of leafs and a trunk and that leafs is an object of either green or yellow color. Obviously an object described by another shade of green would never have been characterized as being leafs by a crisp system. That's where fuzzy logic fits in, allowing assertions of the form $(object : green) \geq 0.7$ that imply an object being green to a certain degree. As it will be demonstrated this degree plays an important role throughout the whole reasoning procedure.

The rest of the paper is organized as follows: section 2 provides some preliminary report on the CARIN system and fuzzy logic, section 2 provides the syntax and semantics of our system, section 4 describes the inference problems addressed by our system, section 5 presents a consistency checking algorithm for fuzzy $\mathcal{ALCNR}$ and finally section 6 presents an algorithm for answering to conjunctive queries and union of conjunctive queries.

2 Preliminaries

2.1 CARIN

The CARIN language combines the DL $\mathcal{ALCNR}$ with Horn rules. CARIN' s structural elements are concept names, role names, individuals and ordinary predicates. Individuals reflect the objects of our universe while concepts and roles correspond to unary and binary predicates. Ordinary predicates refer to predicates of any arity that are found only in the ABox and in the Horn rule component. CARIN enables us to create concept descriptions using the following constructors:

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

where A is a concept name (primitive concept), R is a role name and C, D denote concept descriptions.

A CARIN knowledge base K consists of an ABox, TBox and a Horn rule component. The ABox consists of a set of concept, role and ordinary predicate assertions of the form: $C(a)$, $R(a,b)$ and $q(a_1,\ldots,a_k)$ where q is an ordinary predicate and $a, b, a_1, \ldots, a_k$ are individuals in K. The TBox is a set of concept inclusions or definitions of the form $C \sqsubseteq D$, $C := D$ and role definitions $P_1 \sqcap \ldots \sqcap P_k \equiv R$, where $P_1, \ldots P_k$ are role names. Finally the Horn rules component consists of a set of Horn rules of the form $p_1(\overline{X}_1) \wedge \ldots \wedge p_k(\overline{X}_k) \Rightarrow q(\overline{Y})$ where $p_1, \ldots, p_k$ are either concept descriptions, role definitions or ordinary predicates of the appropriate arity.

The semantics of CARIN are given via interpretations. An interpretation consists of a domain and an interpretation function $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where the domain is a non-empty set of objects and the interpretation function maps: each individual name a to an object $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each concept name C to a subset of $\Delta^{\mathcal{I}}$, $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, each role name R to a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ and each ordinary predicate q to a n-ary relation $q^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \ldots \times \Delta^{\mathcal{I}}$. An interpretation $\mathcal{I}$ satisfies $C(a)$, $R(a,b)$ and $q(a_1,\ldots,a_k)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$, $\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$ and $\langle a_1^{\mathcal{I}}, \ldots, a_k^{\mathcal{I}} \rangle \in q^{\mathcal{I}}$. TBox axioms $C \sqsubseteq D$, $C := D$ and $R := P_1 \sqcap \ldots \sqcap P_k$ imply that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, $C^{\mathcal{I}} := D^{\mathcal{I}}$ and $R^{\mathcal{I}} \equiv P_1^{\mathcal{I}} \cap \ldots \cap P_k^{\mathcal{I}}$. Finally Horn rules of the form $p_1(\overline{X}_1) \wedge \ldots \wedge p_k(\overline{X}_k) \Rightarrow q(\overline{Y})$ imply that for any mapping $\psi : VarsIndivs(\overline{X}_1 \cup \ldots \cup \overline{X}_k) \rightarrow \Delta^{\mathcal{I}}$, if $\psi(\overline{X}_i) \in p_i^{\mathcal{I}}$, then $\psi(\overline{Y}) \in q^{\mathcal{I}}$.

2.2 Fuzzy Sets

Fuzzy set theory and fuzzy logic enables to represent uncertain and imprecise knowledge [21]. In classical set theory an element x which belongs to the universe Ω, $x \in \Omega$, may or may not belong to a subset A of Ω. This can be represented by a mapping $\chi_A : \Omega \rightarrow \{0,1\}$, if $\chi_A(x) = 1$ then $x \in A$ else if $\chi_A(x) = 0$ then $x \notin A$. In fuzzy set theory, a fuzzy subset A of Ω has a mapping $\mu_A : \Omega \rightarrow [0,1]$ which means that instead of saying that $x \in A$ we can claim that x belongs to A to a certain degree. Additionally a binary fuzzy relation over two crisp sets Ω_1, Ω_2 is a mapping $R : \Omega_1 \times \Omega_2 \rightarrow [0,1]$ and a n-ary relation q over n crisp sets $\Omega_1, \ldots \Omega_n$ is a mapping $q : \Omega_1 \times \ldots \times \Omega_n \rightarrow [0,1]$.

The classical set theoretical operations of complement, union intersection and implication are also extended in fuzzy set theory by using triangular norm operations [21]. Because of the difficulty of extending DLs with arbitrary fuzzy set operations our system uses some standard norm operations [16]. These norms are: the Lukasiewicz negation $c(a) = 1 - a$, the Gödel $t - norm$ for conjunction, $t(a, b) = \min(a, b)$, the Gödel t-conorm for disjunction $u(a, b) = \max(a, b)$ and the Kleene-Dienes fuzzy implication, $J(a, b) = \max(1 - a, b)$.

3 The language of Fuzzy Carin

As stated, *non recursive fuzzy CARIN* is a language, which combines the description logic fuzzy $\mathcal{ALCNR}$ with non recursive Horn Rules. A fuzzy CARIN knowledge base K is composed of three components $K = \langle \mathcal{T}, \mathcal{H}, \mathcal{A} \rangle$, a DL terminology component $\mathcal{T}$ also called a TBox, a Horn rules component $\mathcal{H}$ and a ground facts component $\mathcal{A}$ also called an ABox. In the syntax and semantics that we propose, we consider that fuzziness exists only in the ground facts component.

3.1 Syntax

Fuzzy CARIN's structural elements are a set of individuals **I**, an alphabet of concept names **C**, role names **R** and ordinary predicate names **Q**. Elements of **I** represent the objects in our universe, while **C** and **R** correspond to unary and binary fuzzy relationships between individuals in **I**. Elements of **Q** correspond to relationships, between individuals, of any arity.

Terminological component in fuzzy CARIN: The fuzzy CARIN terminological component $\mathcal{T}$ has the same syntax as the crisp. Complex concepts are built from concept and role names using the constructors of $\mathcal{ALCNR}$ as described in Equation 1 where A is a concept name, C and D are concept descriptions and R is a role definition.

$$C, D \longrightarrow A \mid \top \mid \bot \mid C \sqcap D \mid C \sqcup D \mid \neg C \mid \forall R.C \mid \exists R.C \mid (\geq m\ R) \mid (\leq m\ R) \quad (1)$$

The TBox contains *concept definitions* $A := D$, concept inclusions $C \sqsubseteq D$ and role definitions of the form $R := P_1 \sqcap \ldots \sqcap P_k$, where P_is are role names.[3]

Horn rules in fuzzy CARIN: The Horn rule component $\mathcal{H}$ of a fuzzy CARIN knowledge base K contains a set of Horn rules that are logical sentences of the form:

$$p_1(\overline{X}_1) \wedge \ldots \wedge p_k(\overline{X}_k) \Rightarrow q(\overline{Y}) \quad (2)$$

where $\overline{X}_1, \cdots, \overline{X}_k$ and $\overline{Y}$ are tuples of variables and individuals and $p_1, \cdots p_k$ may be concept names, roles or ordinary predicates while q is *always* an ordinary predicate. The antecedents of a Horn rule are called its body and the consequents are called its head.

Fuzzy as well as the classic CARIN are, as stated before, hybrid systems, which means that there is a clear distinction between their DL and Horn rule part. For this reason ordinary predicates are defined as predicates of any arity that locate only in $\mathcal{H}$ and $\mathcal{A}$ and cannot be part of a concept description, even if they are unary or binary predicates. Additionally in order to have a sound

[3] In some bibliography role definitions may be a part of an RBox $\mathcal{R}$ instead of a TBox.

and complete algorithm, variables located in $\overline{Y}$ must also be located in one of the $\overline{X}_i$'s and only non-recursive Horn rules are adopted. A set of rules is said to be recursive if there is a cycle in the dependency relation among ordinary predicates, i.e an ordinary predicate q depends on a predicate p when p appears in the body of a rule whose head is q and dependency is a transitive relation.

Ground fact component: The ground fact component $\mathcal{A}$ of a fuzzy CARIN knowledge base contains a set of fuzzy assertions as shown in table 1:

Table 1. Fuzzy CARIN assertions

$(a : C) \bowtie n$	where $a \in \mathbf{I}$, $\bowtie \in \{\geqslant, >, \leqslant, <\}$, $n \in [0, 1]$ and C is a concept description
$(\langle a, b\rangle : R) \rhd n$	where $a, b \in \mathbf{I}$, $\rhd \in \{\geqslant, >\}$, $n \in [0, 1]$ and R is a role name
$(\overline{a} : p) \rhd n$	where $\overline{a}$ is a tuple of individuals, $\rhd \in \{\geqslant, >\}$, $n \in [0, 1]$ and p is an ordinary predicate of any arity.

Intuitively a fuzzy assertion of the form $(weather : cloudy) \geqslant 0.5$ means that the weather is cloudy with a degree at least equal to 0.5. We call assertions defined by $\geqslant, >$ *positive assertions*, denoted with $\rhd$, while those defined by $\leqslant, <$ *negative assertions*, denoted with $\lhd$. $\bowtie$ stands for any type of inequality. In fuzzy CARIN, we consider only positive role assertion, since negative assertions would imply the existence of role negation and union of roles in $\mathcal{ALCNR}$, which would lead to undecidability. Similarly for ordinary predicates we use only positive assertions since negation cannot be expressed in simple Horn Rules.

3.2 Semantics

The semantics of the terminological component are given via fuzzy interpretations which use membership functions that range over the interval $[0, 1]$. A fuzzy interpretation is a pair $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where the domain $\Delta^{\mathcal{I}}$ is a non empty set of objects and $\cdot^{\mathcal{I}}$ is a *fuzzy interpretation function*, which maps:

1. An individual name $a \in \mathbf{I}$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$,
2. A concept name $A \in \mathbf{C}$ to a membership function $A^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0, 1]$,
3. A role name $R \in \mathbf{R}$ to a membership function $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0, 1]$,
4. An ordinary predicate $q \in \mathbf{Q}$ of l-arity to a membership function $q^{\mathcal{I}}$: $\underbrace{\Delta^{\mathcal{I}} \times \ldots \times \Delta^{\mathcal{I}}}_{l} \to [0, 1]$,
5. Finally, we make the unique names assumption, i.e. for each tuple of elements $a, b \in \mathbf{I}$, $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ holds.

The semantics of concept descriptions are given by the equations in table 2 where $a, b \in \Delta^{\mathcal{I}}$ and C,D are concept descriptions, R is a role description and A is a concept name. *Terminological component satisfiability:* An interpretation $\mathcal{I}$ satisfies the terminological component $\mathcal{T}$, iff

Table 2. Semantics

Syntax	*Semantics*
A	$A^{\mathcal{I}}(a) = n$ where $n \in [0,1]$
$\top$	$\top^{\mathcal{I}}(a) = 1$
$\bot$	$\bot^{\mathcal{I}}(a) = 0$
$\neg C$	$(\neg C)^{\mathcal{I}}(a) = 1 - C^{\mathcal{I}}(a)$
$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}}(a) = \min(C^{\mathcal{I}}(a), D^{\mathcal{I}}(a))$
$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}}(a) = \max(C^{\mathcal{I}}(a), D^{\mathcal{I}}(a))$
$\forall R.C$	$(\forall R.C)^{\mathcal{I}}(a) = \inf_{b \in \Delta^{\mathcal{I}}}\{\max(1 - R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b))\}$
$\exists R.C$	$(\exists R.C)^{\mathcal{I}}(a) = \sup_{b \in \Delta^{\mathcal{I}}}\{\min(R^{\mathcal{I}}(a,b), C^{\mathcal{I}}(b))\}$
$(\geq m\ R)$	$(\geq mR)^{\mathcal{I}}(a) = \sup_{b_1,\cdots,b_m \in \Delta^{\mathcal{I}}} \min_{i=1}^{m}\{R^{\mathcal{I}}(a,b_i)\}$
$(\leq m\ R)$	$(\leq mR)^{\mathcal{I}}(a) = \inf_{b_1,\cdots,b_{m+1} \in \Delta^{\mathcal{I}}} \max_{i=1}^{m+1}\{1 - R^{\mathcal{I}}(a,b_i)\}$

- $\forall a \in \Delta^{\mathcal{I}},\ C^{\mathcal{I}}(a) \leqslant D^{\mathcal{I}}(a)$ for each concept inclusion axiom $C \sqsubseteq D$ in $\mathcal{T}$,
- $\forall a \in \Delta^{\mathcal{I}},\ C^{\mathcal{I}}(a) = D^{\mathcal{I}}(a)$ for each concept definition axiom $C := D$ in $\mathcal{T}$,
- $\forall a, b \in \Delta^{\mathcal{I}}$, $\min(P_1^{\mathcal{I}}(a,b), \ldots, P_k^{\mathcal{I}}(a,b)) = R^{\mathcal{I}}(a,b)$ for each role definition axiom $P_1 \sqcap \ldots \sqcap P_k := R$ in $\mathcal{T}$.

Horn rule satisfiability: An interpretation $\mathcal{I}$ satisfies a Horn rule $p_1(\overline{X}_1) \wedge \ldots \wedge p_k(\overline{X}_k) \Rightarrow q(\overline{Y})$ iff for every mapping ψ from the variables and individuals of $\overline{X}_1, \ldots, \overline{X}_k, \overline{Y}$ to the objects of $\Delta^{\mathcal{I}}$, where each individual a is mapped to $a^{\mathcal{I}}$, $\min\left(p_1^{\mathcal{I}}(\psi(\overline{X_1})), \ldots, p_k^{\mathcal{I}}(\psi(\overline{X_k}))\right) \leq q(\psi(\overline{Y}))$ holds. The Horn rule component is satisfied iff all rules in it are satisfied.

Ground fact component satisfiability: A fuzzy interpretation satisfies the *ground fact component* $\mathcal{A}$ iff it satisfies all fuzzy assertions in $\mathcal{A}$ as described in table 3. In this case we say $\mathcal{I}$ is *a model* of $\mathcal{A}$ and it is denoted as $\mathcal{I} \models \mathcal{A}$. If $\mathcal{A}$ has a model we then say that it is *consistent*.

Table 3. Fuzzy assertion satisfiability

$\mathcal{I}$ satisfies	iff
$(a : C) \bowtie n$	$C^{\mathcal{I}}(a^{\mathcal{I}}) \bowtie n$
$(\langle a, b\rangle : R \rhd n)$	$R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \rhd n$
$(\langle a_1, \ldots, a_k\rangle : q \rhd n)$	$q^{\mathcal{I}}({a_1}^{\mathcal{I}}, \ldots, {a_k}^{\mathcal{I}}) \rhd n$

In fuzzy CARIN we consider that each concept assertion is in its positive inequality formal, negation normal, normalized form i.e. only concept assertions of the form $(a : C) \geq n$ are allowed, where C is in its negation normal form. The same applies for role and ordinary predicate assertions.

Negative assertions can be converted to their Positive Inequality Normal Form (PINF) by applying the fuzzy complement in both sides of the inequality as described in [22]. For example $(a : C) \leq n$ and $(a : C) < n$ are being transformed into $(a : \neg C) \geq 1 - n$ and $(a : \neg C) > 1 - n$.

We also assume that all concepts are in their Negation Normal Form. A concept can be transformed into its NNF by pushing negation inwards making use of the following concept equivalences [16, 17]:

$$\begin{array}{llll} \neg(C \sqcup D) & \equiv (\neg C \sqcap \neg D) & \neg(C \sqcap D) & \equiv (\neg C \sqcup \neg D) \\ \neg\exists R.C & \equiv \forall R.(\neg C) & \neg\forall R.C & \equiv \exists R.(\neg C) \\ \neg \geq p_1 R & \equiv \leq (p_1 - 1)R & \neg \leq p_2 R & \equiv \geq (p_2 + 1)R \\ \neg\neg C = C & & & \end{array}$$

where $p_1 \in \mathbb{N}^*$ and $p_2 \in \mathbb{N}$ in the above equations.

Normalized assertions, are assertions where $>$ is eliminated with $\geq$. This can be achieved by introducing a positive, infinitely small value ϵ which, from an analysis point of view, would be equal to 0^+. Following [23] each concept assertion $a : C > n$ is normalized to $a : C \geq n + \epsilon$. The same kind of normalization holds for role and ordinary predicate assertions. It has been proven in [23] that each model $\mathcal{I}$ of K is also a model of K's normalized form and vice versa.

Finally following [17] a conjugated pair of fuzzy assertions is a pair of assertions whose semantics are contradicted. If ϕ represents a crisp concept assertion and $\neg\phi$ its negation (e.g. if $\phi \equiv a : C$ then $\neg\phi \equiv a : \neg C$) the instances of conjugated pairs are seen in table 4. An ABox $\mathcal{A}$ with a conjugated pair of fuzzy

Table 4. Conjugated pairs of fuzzy assertions

	$\neg\phi > m$	$\neg\phi \geq m$
$\phi \geq n$	$n + m \geq 1$	$n + m > 1$
$\phi > n$	$n + m \geq 1$	$n + m \geq 1$

assertions has no model $\mathcal{I}$.

Knowledge base satisfiability: An ABox $\mathcal{A}$ is consistent w.r.t. a TBox $\mathcal{T}$ and a Horn rules component $\mathcal{H}$ if it has a model, $\mathcal{I} \models \mathcal{A}$, that satisfies every concept, role inclusion and definition in $\mathcal{T}$ as well as each Horn rule in $\mathcal{H}$. A knowledge base $K = \langle \mathcal{A}, \mathcal{T}, \mathcal{H} \rangle$ is satisfiable when there exists such a model $\mathcal{I}$ which is called a model of a knowledge base K and denoted as $\mathcal{I} \models K$.

4 Reasoning

The most common inference problems addressed by previous fuzzy DL systems are the satisfiability, n-satisfiability, subsumption and the entailment problem [16]. It has been proven in [16, 17] that each one of the previous problems can be reduced to the problem of a knowledge base satisfiability.

Another kind of inference problem interwoven with relational databases is the conjunctive query answering problem. Following [24] we present the definition of the conjunctive query problem for fuzzy DLs.

Definition 1 (Conjunctive Query). *A conjunctive query* (CQ) *over a knowledge base* K *is a set of atoms of the form*

$$CQ = \{p_1(\overline{Y}_1) \rhd n_1 \wedge \ldots \wedge p_k(\overline{Y}_k) \rhd n_k\}$$

where $p_1, \ldots, p_k$ *are either concept names in* $\mathbf{C}$*, role names in* $\mathbf{R}$ *or ordinary predicates in* $\mathbf{Q}$ *and* $\overline{Y}_1, \ldots, \overline{Y}_k$ *are tuples of variables and individuals in* $\mathbf{I}$ *matching each* p_i*'s arity.*

Similarly to assertions, conjunctive queries are also transformed to their normalized form by substituting each $p_i(\overline{Y}_i) > n_i$ in CQ with $p_i(\overline{Y}_i) \geq n_i + \epsilon$.

Definition 2 (Union of Conjunctive Queries). *A union of conjunctive queries* (UCQ) *over a knowledge base* K *is a set of conjunctive queries:*

$$UCQ = \{Q_1, \ldots, Q_l\}$$

where Q_i *is a* CQ *for each* $1 \leq i \leq l$*.*

To say that Q is either a CQ or an UCQ, we simply say that Q is a *query*. We denote by *varsIndivs*(Q) the set of variables and individuals in a query Q, by *vars*(Q) the set of variables in Q and by *Indivs*(Q) the set of individuals in Q.

Queries are interpreted in the standard way. For a CQ, we say that $\mathcal{I}$ models CQ, $\mathcal{I} \models CQ$, iff there exists a mapping $\sigma : varsIndivs(CQ) \to \Delta^{\mathcal{I}}$ such that $\sigma(a) = a^{\mathcal{I}}$ for each $a \in Indivs(CQ)$ and $p_i^{\mathcal{I}}(\sigma(\overline{Y_i})) \geq n$ for each $p(\overline{Y_i}) \geq n$ in CQ. For a union of conjunctive queries $UCQ = \{Q_1, \ldots, Q_l\}$, $\mathcal{I} \models UCQ$ iff $\mathcal{I} \models Q_i$ for some $Q_i \in UCQ$. For a knowledge base K and a query Q, we say that K entails Q, denoted $K \models Q$, iff $\mathcal{I} \models Q$ for each model $\mathcal{I}$ of K.

Definition 3 (Query Entailment). *Let* K *be a knowledge base and* Q *a* query*. The query entailment problem is to decide whether* $K \models Q$*.*

It is important to notice that the query entailment, contrary to the entailment problem, cannot be reduced to consistency checking, since the negation of a query cannot be expressed as part of a knowledge base. For this reason consistency checking does not suffice for answering to conjunctive queries.

5 Consistency checking for Fuzzy CARIN

To say that $K \models Q$ it has to hold that $\mathcal{I} \models Q$ for each model $\mathcal{I}$ of K. Instead of checking an infinite number of interpretations $\mathcal{I}$ satisfying K, our algorithm checks a finite number of completion forests. A completion forest $\mathcal{F}$ is an abstraction of an interpretation $\mathcal{I}$ and in most tableaux algorithms a complete and clash free $\mathcal{F}$ is the proof of the existence of a model of K. In 5.1 we provide an algorithm for consistency checking in $\mathcal{ALCNR}$ and based on this algorithm the conjunctive query answering problem is solved as described in 6.

5.1 $\mathcal{ALCNR}$ Completion Forests

The completion forest introduced is based on the completion forest presented in [15]. As in [15] the application of the expansion rules for the completion forest could lead to an arbitrary number of nodes due to the existence of cyclic concept inclusions. In order to ensure the termination of the expansion rules a blocking condition should be adopted. Contrary to the simple blocking condition embraced by $\mathcal{ALCNR}$ [25] our algorithm adopts the q-blocking condition, introduced in [15], in order to cope with union of conjunctive queries. In the next paragraphs the notions of completion forest, q-blocking and the expansion rules are explained in detail.

Definition 4 (Completion Tree). *A completion tree for fuzzy $\mathcal{ALCNR}$ is a tree, all nodes of which are variables, except from the root node which might be an individual. Each node x is labelled with a set $\mathcal{L}(x) = \{\langle C, \geq, n\rangle\}$, where $C \in sub(K)$ and $n \in [0,1]$. Each edge is labelled with a set $\mathcal{L}(x, y) = \{\langle R, \geq, n\rangle\}$, where $R \in \mathbf{R}$ are roles occurring in K.*

(Completion Forest). A completion forest $\mathcal{F}$ is a collection of trees whose roots, which correspond to individuals, are arbitrarily connected by arcs. As before, edges between root nodes are labelled with the set $\mathcal{L}(x, y) = \{\langle R, \geq, n\rangle\}$, where $R \in \mathbf{R}$.

In the previous definition $sub(K)$ denotes the set of concepts occurring in K along with their sub-concepts.

Example 2. In Figure 1 we see a completion forest for fuzzy $\mathcal{ALCNR}$ where r_1, r_2 correspond to root nodes while $o_1, \ldots, o_8$ are variable nodes created by node generating rules. Each node must be labelled with a set of concepts with degrees and each edge must be labelled with a set of roles with degrees. In this example only nodes r_1, o_1 and edges $\langle r_1, o_1\rangle, \langle r_1, r_2\rangle$ are labelled due to space limitations.

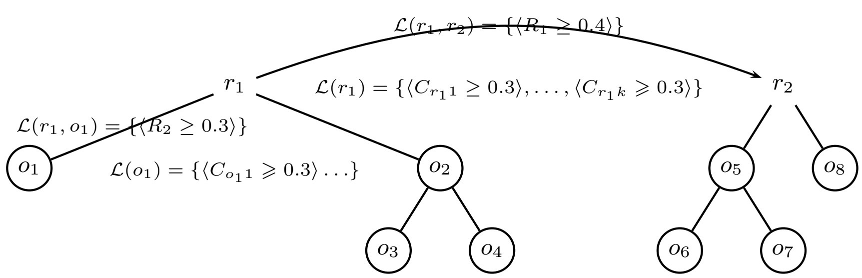


Fig. 1. A fuzzy $\mathcal{ALCNR}$ completion forest

Definition 5 (nodes, vars, R-successor,successor,descendant). *For a completion forest $\mathcal{F}$: (i) $nodes(\mathcal{F})$ denotes the set of nodes in $\mathcal{F}$, (ii) $vars(\mathcal{F})$ denotes* *the set of variable nodes in $\mathcal{F}$, (iii) v is an $R_{\geq n}$*-successor *of w when nodes v and w are connected by an edge $\langle v, w\rangle$ with $\{\langle P_1, \geq, n_1\rangle, \ldots, \langle P_k, \geq, n_k\rangle\} \subseteq \mathcal{L}(\langle x, y\rangle)$, $R := P_1 \sqcap \ldots \sqcap P_k$ and $\min(n_1, \ldots, n_k) \geq n$, (iv) v is a* successor *of w, when v is an $R_{\geq n}$*-successor *of w with $n > 0$, (v) descendent is the transitive closure of successor.*

Example 3. In figure 1, o_1 is a $R_{2 \geq 0.3}$ successor of r_1.

Definition 6 (q-tree equivalence). *The q*-tree *of a variable v is the tree that includes the node v and its successors, whose distance from v is at most q direct-successors arcs. We denote the set of nodes in the q-tree of v by $V_q(v)$. Two nodes $v, w \in \mathcal{F}$ are said to be q-tree equivalent in $\mathcal{F}$ if there exists an isomorphism $\psi : V_q(v) \to V_q(w)$ such that (i) $\psi(v) = w$, (ii) for every $s \in V_q(v)$, $\langle C, \geq, n\rangle \in \mathcal{L}(s)$ iff $\langle C, \geq, n\rangle \in \mathcal{L}(\psi(s))$ (iii) for every $s, t \in V_q(v)$, $\langle R, \geq, n\rangle \in \mathcal{L}(\langle s, t\rangle)$ iff $\langle R, \geq, n\rangle \in \mathcal{L}(\langle \psi(s), \psi(t)\rangle)$. Intuitively, two variables are q-tree equivalent if the trees of depth q of which they are roots are isomorphic.*

Definition 7 (q-Witness). *A node v is the q*-witness *of a node w when (i) v is an ancestor of w, (ii) v and w are q-tree equivalent, (iii) $w \notin V_n(v)$.*

Definition 8 (q-blocking). *A node x is q-blocked when it is the leaf of a q-tree in $\mathcal{F}$ whose root w has a q-witness v and $w \in vars(\mathcal{F})$ or when $\mathcal{L}(x) = \emptyset$.*

Example 4. In Figure 2 o_1 is a 1-witness of o_4, since the 1-tree of o_1 is equivalent of the 1-tree of o_4 because $\mathcal{L}(o_1) = \mathcal{L}(o_4)$, $\mathcal{L}(o_2) = \mathcal{L}(o_5)$, $\mathcal{L}(o_3) = \mathcal{L}(o_6)$ and $\mathcal{L}(o_1, o_2) = \mathcal{L}(o_4, o_5)$, $\mathcal{L}(o_1, o_3) = \mathcal{L}(o_4, o_6)$. For this reason o_5 is blocked by o_2 and o_3 is blocked by o_6.

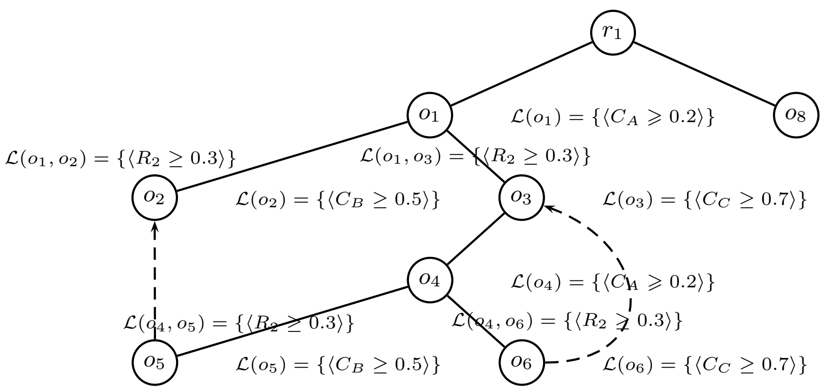


Fig. 2. Blocking Example

Definition 9 (Clash free completion forest). *For a node x, $\mathcal{L}(x)$ contains a* clash *if it contains: (i) A conjugated pair of triples. Conjugated pairs of triples* *are identical to conjugated pairs of fuzzy assertions described in table 4, (ii) one of the triples* $\langle \bot, \geqslant, n\rangle$*, with* $n > 0$*, or* $\langle C, \geq, n\rangle$ *with* $n > 1$*, or (iii) some triple* $\langle \leqslant pR, \geq, n\rangle$*,* x *has* $p+1$ $R_{\geq n'}$*-successors* $y_0, \ldots, y_p$*, with* $n' = 1 - n + \epsilon$ *and* $y_i \neq y_j$ *for all* $0 \leq i < j \leq p$*. A completion forest* $\mathcal{F}$ *is clash free if none of its nodes contains a clash.*

For an $\mathcal{ALCNR}$ ABox $\mathcal{A}$, the algorithm initializes a completion forest $\mathcal{F}_K$ to contain (i) a root node x_0^i, for each individual $a_i \in \mathbf{I}$ in $\mathcal{A}$, labelled with $\mathcal{L}(x_0^i)$ such that $\{\langle C_i, \geq, n\rangle\} \subseteq \mathcal{L}(x_0^i)$ for each assertion of the form $(a_i : C_i) \geq n \in \mathcal{A}$, (ii) an edge $\langle x_0^i, x_0^j\rangle$, for each assertion $(\langle a_i, a_j\rangle : R_i) \geq n \in \mathcal{A}$, labelled with $\mathcal{L}(\langle x_0^i, x_0^j\rangle)$ such that $\{\langle R_i, \geq n\rangle\} \subseteq \mathcal{L}(\langle x_0^i, x_0^j\rangle)$, (iii) the relation $\not\doteq$ as $x_0^i \not\doteq x_0^j$ for each two different individuals $a_i, a_j \in \mathbf{I}$ and the relation $\doteq$ to be empty. $\mathcal{F}$ is expanded by repeatedly applying the completion rules from table 5.

In table 5 rules $\sqcap_{\geq}$, $\sqcup_{\geq}$, $\exists_{\geq}$, $\forall_{\geq}$ are first introduced in [16] and then modified for completion forests in [4], rules $\geq_{\geq}$ and $\leq_{\geq}$ are presented in [17], while rule $\sqsubseteq$ is first introduced in [23]. The $\leq_{r\geq}$ presented in [17] cannot be applied, since $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ holds for every pair of individuals $a, b \in \mathbf{I}$.

Definition 10 (q-complete completion forest). *We denote by* $\mathbb{F}_K$ *the set of completion forests* $\mathcal{F}$ *obtained by applying the expansion rules in table 5 to* $\mathcal{F}_K$*. A completion forest* $\mathcal{F}$ *is q-complete when none of the rules in table 5 can be applied to it. We denote by* $\mathrm{ccf}(\mathbb{F}_K^q)$ *the set of completion forests in* $\mathbb{F}_K$ *that are q-complete and clash free.*

It can be proven that each $\mathcal{F} \in \mathrm{ccf}(\mathbb{F}_K^q)$ can be mapped to a model $\mathcal{I}$ of K and vice versa (detailed proofs can be found in [17]). In section 6 we show how the set $\mathrm{ccf}(\mathbb{F}_K^q)$ can be exploited in order to answer to unions of conjunctive queries.

6 Union of Conjunctive queries

In this section we will introduce an algorithm for answering to union of conjunctive queries over an $\mathcal{ALCNR}$ knowledge base K, where we exam the finite set of clash free completion forests $ccf(\mathcal{F}_{\mathbb{K}}^{|Q|})$. Our algorithm is first presented for union of conjunctive queries free of ordinary predicates (6.1) and then extended for query answering with ordinary predicates (6.2).

6.1 Answering to conjunctive queries without ordinary predicates

In order to have a complete algorithm for answering to conjunctive queries we must add to our $TBox$ the rule $C \sqsubseteq C$ for each concept name C appearing in a conjunctive query. This ensures that in each completion forest either $(x : C) \geq n$ or $(x : C) < n$ [4] holds and consequently it can be checked if a node can be mapped to a variable of our conjunctive query.

[4] $(a : \neg C) > 1 - n + \epsilon$ is its PINF, normalized form

Table 5. Tableaux expansion rules for fuzzy $\mathcal{ALCNR}$

Rule	Description
$\sqcap_{\geq}$	if 1. $\langle C_1 \sqcap C_2, \geq, n\rangle \in \mathcal{L}(x)$, x is not blocked, 2. $\{\langle C_1, \geq, n\rangle, \langle C_2, \geq, n\rangle\} \not\subseteq \mathcal{L}(x)$ then $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{\langle C_1, \geq, n\rangle, \langle C_2, \geq, n\rangle\}$
$\sqcup_{\geq}$	if 1. $\langle C_1 \sqcup C_2, \geq, n\rangle \in \mathcal{L}(x)$, x is not blocked, 2. $\{\langle C_1, \geq, n\rangle, \langle C_2, \geq, n\rangle\} \cap \mathcal{L}(x) = \emptyset$ then $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C\}$ for some $C \in \{\langle C_1, \geq, n\rangle, \langle C_2, \geq, n\rangle\}$
$\exists_{\geq}$	if 1. $\langle \exists R.C, \geq, n\rangle \in \mathcal{L}(x)$, x is not blocked, 2. x has no $R_{\geq n}$-successor y with $\langle C, \geq, n\rangle \in \mathcal{L}(y)$ then create a new node y with $\mathcal{L}(\langle x, y\rangle) = \{\langle R, \geq, n\rangle\}$, $\mathcal{L}(y) = \{\langle C, \geq, n\rangle\}$
$\forall_{\geq}$	if 1. $\langle \forall R.C, \geq, n\rangle \in \mathcal{L}(x)$, x is not blocked, 2. x has an $R_{\geq n'}$-successor y with $n' = 1 - n + \epsilon$ then $\mathcal{L}(y) \rightarrow \mathcal{L}(y) \cup \{\langle C, \geq, n\rangle\}$
$\geq_{\geq}$	if 1. $\langle \geq mR, \geq, n\rangle \in \mathcal{L}(x)$, x is not blocked, 2. there are no m $R_{\geq n}$-successors $y_1, \ldots, y_p$ of x 3. with $y_i \neq y_j$ for $1 \leq i < j \leq m$ then create m new nodes $y_1, \ldots, y_m$, with $\mathcal{L}(\langle x, y_i\rangle) = \{\langle R, \geq, n\}$ and $y_i \neq y_j$ for $1 \leq i < j \leq m$
$\leq_{\geq}$	if 1. $\langle \leq mR, \geq, n\rangle \in \mathcal{L}(x)$, x is not blocked, 2. there are more then m $R_{\geq n'}$-successors of x with $n' = 1 - n + \epsilon$ and there are two of them y, z, with no $y \neq z$, 3. y is not a root node then (a) $\mathcal{L}(z) \rightarrow \mathcal{L}(z) \cup \mathcal{L}(y)$ (b) $\mathcal{L}(\langle x, z\rangle) \rightarrow \mathcal{L}(\langle x, z\rangle) \cup \mathcal{L}(\langle x, y\rangle)$ (c) $\mathcal{L}(\langle x, y\rangle) \rightarrow \emptyset$, $\mathcal{L}(y) \rightarrow \emptyset$ (d) Set $u \neq z$ for all u with $u \neq y$
$\sqsubseteq$	if 1. $C \sqsubseteq D \in \mathcal{T}$ and 2. $\{\langle \neg C, \geq, 1 - n + \epsilon\rangle, \langle D, \geq, n\rangle\} \cap \mathcal{L}(x) = \emptyset$ for $n \in N^{\mathcal{A}}$ [a] then $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{E\}$ for some $E \in \{\langle \neg C, \geq, 1 - n + \epsilon\rangle, \langle D, \rhd, n\rangle\}$

[a] $N^{\mathcal{A}}$ denotes the set of degrees in ABox assertions as well as the set of degrees in conjunctive queries.

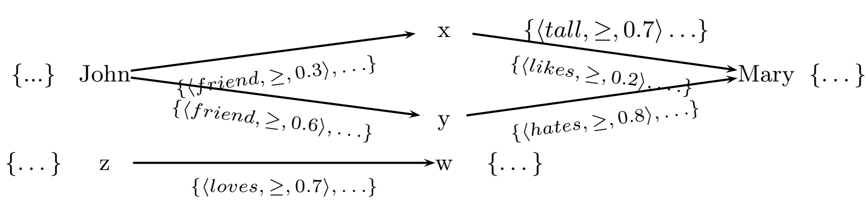
Additionally we have to show why q-blocking is adopted instead of simple blocking. A conjunctive query CQ as presented in definition 1 can be mapped to a graph G_{CQ} whose nodes correspond to variables and individuals, each node is labelled with a set $\mathcal{L}(x) = \{\langle C, \geq, n\rangle\}$ and each edge is labelled with a set $\mathcal{L}(x, y) = \{\langle R, \geq, n\rangle\}$ where C and R are concepts and roles in CQ. Suppose that d_{xy} is the length of the lengthiest acyclic path between nodes x and y, we define $|CQ|$ to be the maximum d_{xy} between the set of pairs of connected nodes in CQ. Naturally we deduce that a conjunctive query CQ cannot be mapped to a subtree of a completion forest $\mathcal{F}$ that has more than $|CQ|$ arcs height. The $|CQ|$-blocking condition ensures that a possible mapping from CQ to $\mathcal{F}$ wont be blocked. In case of a union of conjunctive queries UCQ we will consider that $|UCQ|$ coincidents with the value of the maximum $|CQ|$.

Example 5. The conjunctive query:

$$CQ = \left\{ \begin{array}{l} friend(John, x) \geq 0.3,\ tall(x) \geq 0.7, \\ likes(x, Mary) \geq 0.2,\ friend(John, y) \geq 0.6, \\ hates(y, Mary) \geq 0.8,\ loves(z, w) \geq 0.4 \end{array} \right\}$$ [5]

is represented by the graph in figure 3. For this conjunctive query $|CQ| = 2$.

Fig. 3. Conjunctive query mapped to a graph



Definition 11. *Suppose we have a query* $Q = C_1(x_1) \geq n_1 \wedge \ldots \wedge C_k(x_k) \geq n_k \wedge R_1(y_1, z_1) \geq n_{k+1} \wedge \ldots \wedge R_l(y_l, z_l) \geq n_{k+l}$*. For a completion forest* $\mathcal{F}$ *we say that* $Q \hookrightarrow \mathcal{F}$ *iff there exists a mapping* $\sigma : varsIndivs(Q) \to \mathrm{nodes}(\mathcal{F})$ *such that* $\{\langle C_i, \geq, n_i \rangle\} \in \mathcal{L}(\sigma(x_i))$ *and* $\sigma(y_j)$ *is an* $R_{\geq n_j}$*-successor of* $\sigma(z_j)$ *for each* $1 \leq i \leq k$ *and* $k+1 \leq j \leq l$*. For a union of conjunctive queries* $UCQ = \{Q_1, \ldots, Q_l\}$ *we say that* $UCQ \hookrightarrow \mathcal{F}$ *iff* $Q_i \hookrightarrow \mathcal{F}$ *for some* $Q_i \in UCQ$*.*

It can be proven that if a mapping $Q \hookrightarrow \mathcal{F}$ exists for each $\mathcal{F} \in ccf(\mathcal{F}_{\mathbb{K}}^{|Q|})$, then $K \models Q$.

6.2 Answering to Conjunctive Queries with ordinary predicates

Initially, we will consider conjunctive queries containing no assertions about ordinary predicates, in such a case it holds:

Proposition 1. *Suppose that we have a conjunctive query of the form* $Q = \{p_1 \geq n_1 \wedge \ldots \wedge p_k \geq n_k \wedge \ldots \wedge p_m \geq n_m\}$ *and a set of Horn Rules related to* p_k*:*

$$\mathcal{H}_{p_k} = \left\{ \begin{array}{c} p_k \leftarrow p_{11} \wedge \ldots \wedge p_{1l_1}, \\ \vdots \\ p_k \leftarrow p_{m1} \wedge \ldots \wedge p_{ml_m} \end{array} \right\}$$

Q can be replaced with a union of conjunctive queries $UCQ = \{Q_1, \ldots, Q_m\}$ *where in each* Q_j*,* $p_k \geq n_k$ *is replaced with* $p_{k1} \geq m_k \wedge \ldots \wedge p_{kl_k} \geq m_k$*.*

[5] Here we claim that someone may like, hate, love or be a friend of someone else at certain degree

Fuzzy assertions about ordinary predicates can be introduced by the use of pseudo-roles and pseudo-concepts. For example an assertion about an ordinary predicate $q(a_1, \ldots, a_m) \geq n$ can be substituted by a set of role assertions $\mathcal{A}_q = \{R_{q1}(a_1, a_2) \geq n, \ldots, R_{q(m-1)}(a_{m-1}, a_m) \geq n\}$ and a Horn rule $q \leftarrow R_{q1} \wedge \ldots \wedge R_{q(m-1)}$. In such a case conjunctive queries and union of conjunctive queries can be recurrently stretched to union of conjunctive queries containing only concepts and roles (pseudo or not) since only acyclic Horn Rules are allowed in $\mathcal{H}$. So the problem is reduced to the problem described in section 6.1.

7 Conclusions

Till now we have presented the integration of fuzzy logic, description logics and Horn rules, into fuzzy CARIN. The acquired system matches the benefits of its ancestors, providing a very expressive language for handling uncertainty and imprecision, with the counterweight of its high complexity, resulting from the high complexity of its structural elements. Future directions concern the study of fuzzy CARIN's complexity and its extension with more expressive DLs (a guide towards that direction is provided in [24]). It should also be extended to answer to other kind of inference problems, originating from the fuzzy DL domain, such as *glb* queries [16].

Acknowledgement.

This work was partially supported by the European Commission under projects X-Media (FP6-26978) and K-space (IST-2005-027026). The work of Giorgos Stoilos was partially supported by the Greek Secretariat of Research and Technology (PENED Ontomedia 03 ED 475).

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: The Description Logic Handbook: Theory, Implementation, and Applications. In Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., eds.: Description Logic Handbook, Cambridge University Press (2003)
2. Meghini, C., Sebastiani, F., Straccia, U.: A model of multimedia information retrieval. J. ACM **48** (2001) 909–970
3. Goble, C.A., Haul, C., Bechhofer, S.: Describing and classifying multimedia using the description logic grail. In: Storage and Retrieval for Image and Video Databases (SPIE). (1996) 132–143
4. Stoilos, G., Stamou, G., Tzouvaras, V., Pan, J., Horrocks, I.: A fuzzy description logic for multimedia knowledge representation, Proc. of the International Workshop on Multimedia and the Semantic Web (2005)
5. Golbreich, C., Bierlaire, O., Dameron, O., Gibaud, B.: Use case: Ontology with rules for identifying brain anatomical structures. In: Rule Languages for Interoperability, W3C (2005)

6. I. Herman, J.H.: Web ontology language (OWL). technical report (2004) http://www.w3.org/2004/OWL/.
7. Herman, I.: Semantic web. technical report (2007) http://www.w3.org/2001/sw/.
8. Baader, F., Brandt, S., Lutz, C.: Pushing the el envelope. In Kaelbling, L.P., Saffiotti, A., eds.: IJCAI, Professional Book Center (2005) 364–369
9. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In Doherty, P., Mylopoulos, J., Welty, C.A., eds.: KR, AAAI Press (2006) 57–67
10. Antoniou, G., Damasio, C.V., Grosof, B., Horrocks, I., Kifer, M., Maluszynski, J., Patel-Schneider, P.F.: Combining rules and ontologies. a survey (2006)
11. Grosof, B.N., Horrocks, I., Volz, R., Decker, S.: Description logic programs: combining logic programs with description logic. In: WWW. (2003) 48–57
12. Horrocks, I., Patel-Schneider, P.F.: A proposal for an OWL rules language. In Feldman, S.I., Uretsky, M., Najork, M., Wills, C.E., eds.: WWW, ACM (2004) 723–731
13. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A.: Al-log: Integrating datalog and description logics. J. Intell. Inf. Syst. **10** (1998) 227–252
14. Kifer, M., Lausen, G., Wu, J.: Logical foundations of object-oriented and frame-based languages. J. ACM **42** (1995) 741–843
15. Levy, A.Y., Rousset, M.C.: Combining horn rules and description logics in carin. Artif. Intell. **104** (1998) 165–209
16. Straccia, U.: Reasoning within fuzzy description logics. J. Artif. Intell. Res. (JAIR) **14** (2001) 137–166
17. Stoilos, G., Stamou, G.B., Tzouvaras, V., Pan, J.Z., Horrocks, I.: The fuzzy description logic $f-\mathcal{SHIN}$. In da Costa, P.C.G., Laskey, K.B., Laskey, K.J., Pool, M., eds.: ISWC-URSW. (2005) 67–76
18. Lukasiewicz, T.: Fuzzy description logic programs under the answer set semantics for the semantic web. In Eiter, T., Franconi, E., Hodgson, R., Stephens, S., eds.: RuleML, IEEE Computer Society (2006) 89–96
19. Straccia, U.: Fuzzy description logic programs. In: Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, (IPMU-06). (2006) 1818–1825
20. Eiter, T., Lukasiewicz, T., Schindlauer, R., Tompits, H.: Well-founded semantics for description logic programs in the semantic web. In Antoniou, G., Boley, H., eds.: RuleML. Volume 3323 of Lecture Notes in Computer Science., Springer (2004) 81–97
21. Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Volume 567 of [26]
22. 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) 559–567
23. Stoilos, G., Straccia, U., Stamou, G.B., Pan, J.Z.: General concept inclusions influzzy description logics. In Brewka, G., Coradeschi, S., Perini, A., Traverso, P., eds.: ECAI, IOS Press (2006) 457–461
24. Ortiz, M., Calvanese, D., Eiter, T.: Characterizing data complexity for conjunctive query answering in expressive description logics. In: AAAI, AAAI Press (2006)
25. Baader, F., Hollunder, B.: A terminological knowledge representation system with complete inference algorithms. [26] 67–86
26. Boley, H., Richter, M.M., eds.: Processing Declarative Knowledge, International Workshop PDK'91, Kaiserslautern, Germany, July 1-3, 1991, Proceedings. In Boley, H., Richter, M.M., eds.: PDK. Volume 567 of Lecture Notes in Computer Science., Springer (1991)