Knowl Inf Syst
DOI 10.1007/s10115-010-0304-1

REGULAR PAPER

Expressive reasoning with horn rules and fuzzy description logics

Theofilos Mailis · Giorgos Stoilos · Giorgos Stamou

Received: 1 November 2007 / Revised: 10 July 2009 / Accepted: 5 May 2010

Abstract In this paper, we describe 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 language. We introduce the decision problems of answering to conjunctive queries, unions of conjunctive queries and the existential entailment problem and provide a sound and complete algorithm that permits reasoning with the DL fuzzy $\mathcal{ALCNR}$ extended with non-recursive Horn rules. This extension is most useful in realistic applications that handle uncertain or imprecise data such as multimedia processing and medical applications.

Keywords Fuzzy description logics · Horn rules · Conjunctive queries · Fuzzy CARIN · Existential entailment

1 Introduction

Over the last two decades, decidable fragments of first order logic, like Description Logics (DLs) [3], have been brought into focus by the Artificial Intelligence community. A Description Logic (DL) allows us to define sets of objects referred as concepts (corresponding to unary relations), and relationships between objects called roles (corresponding to binary relations). Complex concept descriptions are built from simple concepts by the use of various constructors such as $\sqcap$, $\sqcup$, $\exists$, i.e. the complex concept description $Man \sqcap \exists hasChild.Girl$ describes all men with a female child. DLs' well-founded semantics, great expressiveness along with their sound, complete, and empirically tractable reasoning services have enforced their utilization in numerous domains such as multimedia [11,26,31] and medical [12] applications.

T. Mailis (✉) · G. Stamou
Electrical and Computer Engineering Department,
National Technical University of Athens, Athens, Greece
e-mail: theofilos@image.ntua.gr

G. Stoilos
Computing Laboratory, Oxford University, Oxford, UK

Published online: 21 May 2010

Furthermore, DLs provide the formal foundation for the standard web ontology language OWL [16] which is a milestone for the Semantic Web [15]. In particular, the languages OWL-DL and OWL-Lite are syntactic variants of the DLs $\mathcal{SHIF}(D^+)$ and $\mathcal{SHOIN}(D^+)$, respectively [19].

DLs main feature, their class-based knowledge representation formalism, sets a limit to their expressive power as they are incapable of providing complex descriptions about role predicates. Even expressive DLs such as $\mathcal{SHOIQ}$ lack the ability of expressing so much as a simple composition between roles[1]. Therefore, as a next step in the development of the Semantic Web, the need for systems providing reasoning services for languages integrating DLs with rules occurred. A natural choice for such an integration would be classes of rule languages originating from logic programming and non-monotonic reasoning [1]. In [1], the "cream" of languages combining rules and DLs is presented. Languages such as DLP [14], SWRL [18], $\mathcal{AL}$-log [8], F-logic [21] and CARIN [24] consist of different approaches for integrating DLs with rules. These languages are divided into hybrid that distinct between the predicates in the rules and the DL part, and homogeneous that have no such distinction.

CARIN is such an hybrid language that combines the DL $\mathcal{ALCNR}$ with Horn rules (two orthogonal subsets of first order logic). Horn rules are a natural representation language used in many application domains. Their main advantage is that they are a tractable subset of first-order logic for which several practical efficient inference procedures have been developed. By combining the expressive power of both formalisms and using its existential entailment algorithm the CARIN language: (i) *offers a sound and complete inference procedure for non-recursive knowledge bases*, (ii) *can solve the decision problem associated to answering (arbitrary) unions of conjunctive queries and* (iii) *provides an algorithm for rule and query subsumption over* $\mathcal{ALCNR}$.

Though CARIN offers great expressive power in order to represent a fragment of our universe, it is incapable of encoding inherently imprecise or vague information. Imprecision emerges from our lack of knowledge about a certain fact, e.g. we assume that the blurred region in the background of a picture is a lion, while vagueness 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. In order to represent vague (fuzzy) information several formalisms, such as f$_{KD}$-$\mathcal{ALC}$ [35], f$_{KD}$-$\mathcal{SI}$ [31,32], f$_{KD}$-$\mathcal{SHIN}$ [33], combining DLs with fuzzy set theory and logic have been proposed. The main difference between these fuzzy DL languages and their crisp counterparts is that concepts (and roles) correspond to fuzzy unary (binary) relations. For example if the crisp concept tall characterizes a person in our universe as tall, its fuzzy counterpart characterizes this person as tall to a certain degree.

Based on these DLs, we propose fuzzy CARIN. An extension of non-recursive CARIN that allows to represent and perform reasoning with vague information. The need for fuzzy extensions in systems combining DLs with rules is evident: in multimedia and information retrieval applications [9,13] to provide ranking degrees, in geospatial applications [27] to cope with vague concepts like "near", "far", as well as in World Wide Web applications such as business databases [41] and many more.

Example 1.1 Suppose that we have a rather "optimistic" application for object recognition. This application consists of an image processing module that extracts some information about the regions of an image and a DL extended with rules module that combines this information

[1] Recent languages such as $\mathcal{EL}^{++}$ [2], $\mathcal{SROIQ}$ [17] move toward this direction.

for the extraction of implicit knowledge:

$$(GreenColored \sqcup YellowColored) \sqcap RegularTextured \sqsubseteq Leafs$$
$$Trunk(x) \wedge isConnected(x, y) \wedge Leafs(y) \Rightarrow Tree(x, y)$$

In this case, a DL axiom implies that an object of either green or yellow color and regular texture is a leafs object whereas a rule implies that a tree is an object consisting of leafs and a trunk. Obviously, an object described by another shade of green would have never been characterized 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.

To the best of our knowledge, though there exists a great amount of work involving the integration of fuzzy logic into DLs, little work has been done toward the extension of fuzzy DLs with fuzzy Logic Programs(LPs). As stated in [10] the systems integrating DLs with LPs are based on three different approaches: the so-called axiom-based approach, the DL-log approach, and the autoepistemic approach. The fuzzy CARIN language corresponds to the first category of axiom-based systems. Other systems belonging to this category have been presented in [39] and [43]. A language extending the DL-log approach with fuzziness has been presented in [38], while [25] presents a fuzzy extension of the autoepistemic approach.

The main contribution of this paper can be briefly summarized as follows:

- We provide the syntax and semantics of a fuzzy CARIN knowledge base. A fuzzy CARIN knowledge base is constituted of an ABox, a TBox, and a Horn rules components. The semantics of the ABox and TBox components are in accordance with the semantics presented in [33] for the f_{KD}-$\mathcal{SHIN}$ language.
- We introduce the problems of conjunctive queries (CQ), unions of conjunctive queries (UCQ), and existential entailment. For these three problems, we provide the appropriate semantics based on fuzzy interpretations. Although there has been quite a few work on fuzzy SQL, such as [6], as well as on querying fuzzy DLs [29], as far as we know no such definition of conjunctive queries, unions of conjunctive queries, and existential entailment exists in fuzzy DLs and fuzzy relational databases.
- We provide an algorithm for answering the problems of conjunctive queries and unions of conjunctive queries for knowledge bases with an empty Horn rule component. This algorithm is proved to be sound, complete and terminating. More than this, we introduce a procedure for reducing the existential entailment problem to the union of conjunctive queries answering problem.
- Finally, we introduce a sound and complete algorithm for reducing the problem of answering to unions of conjunctive queries with respect to (w.r.t.) a knowledge base with a non-empty Horn rule component, to that of answering to unions of conjunctive queries w.r.t. a knowledge base with an empty Horn rules component.

The rest of the paper is organized as follows. Section 2.1 provides a preliminary report on the initial CARIN language presented by Alon Y. Levy and Marie-Christine Rousset in [24]. A short introduction on fuzzy DLs along with the most important fuzzy operators is presented in Sect. 2.2. In Sect. 3.1, we present the syntax of the fuzzy CARIN language, i.e. its constructive elements and the formalism that can be used. The meaning of these formalisms and their constructive elements is investigated through Sect. 3.2 that describes its semantics via fuzzy interpretations. In Sect. 3.3, we introduce the conjunctive query, the union of conjunctive queries and the existential entailment problems and present their extensions for fuzzy

DLs. A consistency checking algorithm over an $\mathcal{ALCNR}$ knowledge base is presented in Sect. 4, which is the milestone for the union of conjunctive queries and existential entailment algorithms, presented in Sect. 4.3.

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 (predicates of any arity). Individuals reflect the objects of our universe, while concepts and roles correspond to unary and binary predicates reflecting sets or binary relations over the objects of our universe. 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, $n \in \mathbb{N}$ 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 $C \sqsubseteq D$, $C \equiv D$ where C, D are concept descriptions. Finally, the Horn rules component consists of a set of Horn rules of the form $p_1(\overline{X}_1) \wedge \cdots \wedge p_k(\overline{X}_k) \Rightarrow q(\overline{Y})$ where $p_1, \ldots, p_k$ are either concept descriptions, roles or ordinary predicates of the appropriate arity.

The semantics of CARIN are given via interpretations. An interpretation, $\mathcal{I}$, consists of a domain and an interpretation function $\langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, 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 $q^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \cdots \times \Delta^{\mathcal{I}}$. 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 $\langle a_1^{\mathcal{I}}, \ldots, a_k^{\mathcal{I}} \rangle \in q^{\mathcal{I}}$. An interpretation, $\mathcal{I}$, satisfies the TBox axioms $C \sqsubseteq D, C \equiv D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ and $C^{\mathcal{I}} \equiv D^{\mathcal{I}}$. Finally, Horn rules of the form $p_1(\overline{X}_1) \wedge \cdots \wedge p_k(\overline{X}_k) \Rightarrow q(\overline{Y})$ imply that for any mapping ψ : varsIndivs$(\overline{X}_1 \cup \cdots \cup \overline{X}_k) \rightarrow \Delta^{\mathcal{I}}$, 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 [22]. 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 $q : \Omega_1 \times \cdots \times \Omega_n \rightarrow [0, 1]$.

The classical set theoretical operations of complement, union intersection and implication are also extended in fuzzy set theory by using fuzzy set operations [22]. Because of the difficulty of extending DLs with arbitrary fuzzy set operations, our system uses some standard norm operations like several approaches to fuzzy DLs [35]. 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 DL 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 fuzziness exists only in the ground facts component. For example, we can assert that the weather is cloudy with a degree greater or equal than 0.6, $(weather : cloudy) \geq 0.6$.

3.1 Syntax

Fuzzy CARIN's primitive elements are a set of individuals **I**, an alphabet of concept names **C**, an alphabet of role names **R**, and an alphabet of 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**. Finally, 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 one. Complex concept and role descriptions are built from concept and role names using the constructors of $\mathcal{ALCNR}$ as described in the following inductive definition:

$$\begin{aligned} C, D &\longrightarrow A \mid \top \mid \bot \mid C \sqcap D \mid C \sqcup D \mid \neg C \mid \forall R.C \mid \exists R.C \mid \geq m\ R \mid \leq m\ R \\ R &\longrightarrow P \ \mid P_1 \sqcap \cdots \sqcap P_k \end{aligned}$$

where C and D are concept descriptions, A is a concept name, R is a role description (conjunction), $P, P_1, \ldots, P_k$ are role names in **R** and m is a natural number.

The sentences in the terminological component of fuzzy CARIN are concept inclusions. A concept inclusion of the form $C \sqsubseteq D$ indicates that the degree of membership of each object in C is less or equal to its degree of membership in D.

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:

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

where $\overline{X}_1, \ldots, \overline{X}_k, \overline{Y}$ are tuples of variables and individuals, $p_1, \ldots, p_k$ are either concept descriptions, or role names, or ordinary predicates, and q is *always* an ordinary predicate. The antecedent of a Horn rule is called its body and the consequent is called its head.

In order to ensure a sound, complete, and terminating algorithm, we must impose some restrictions on the expressive power of Horn rules. First of all, fuzzy as well as the classic CARIN must be hybrid languages, 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 are allowed only in $\mathcal{H}$ and $\mathcal{A}$, and cannot be part of a concept description, even if they are unary or binary predicates. Additionally, 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. In [24] it is proved that an extension of $\mathcal{ALCNR}$ with Horn rules that do not satisfy these restrictions is undecidable. Since fuzzy DLs are generalizations of crisp DLs, it is safe to conclude that these undecidability results will also hold for the fuzzy case.

Ground fact component in fuzzy CARIN:

The ground fact component $\mathcal{A}$ of a fuzzy CARIN knowledge base contains a set of fuzzy assertions of the form:

$$\begin{aligned}(a : C) &\bowtie n\\ (\langle a, b\rangle : P) &\rhd n\\ (\overline{a} : p) &\rhd n\end{aligned}$$

where C is a concept description, P a role name, p an ordinary predicate, $\rhd \in \{\geq, >\}$, $\bowtie \in \{\geq, >, \leq, <\}$, $n \in [0, 1]$, $a, b \in \mathbf{I}$, and $\overline{a} \in \mathbf{I}^{\kappa}$ where κ is the arity of the p predicate.

Intuitively a fuzzy assertion of the form $(weather : cloudy) \geq 0.5$ means that the weather is cloudy with a degree at least equal to 0.5. We call assertions defined by $\geq, >$ *positive assertions*, denoted with $\rhd$, while those defined by $\leq, <$ *negative assertions*, denoted with $\lhd$. $\bowtie$ stands for any type of inequality. For ordinary predicates, we use only positive assertions since negation cannot be expressed in simple Horn rules.

Example 3.1 Extending the knowledge presented in Example 1.1 with a fuzzy ABox $\mathcal{A}$, referring to the regions of an image, we get the knowledge base $K = \langle \mathcal{T}, \mathcal{H}, \mathcal{A}\rangle$:

$$\begin{aligned}\mathcal{T} = \{&(GreenColored \sqcup YellowColored) \sqcap RegularTextured \sqsubseteq Leafs,\\ &BrownColored \sqcap \exists isConnected.Leafs \sqsubseteq Trunk\}\\ \mathcal{H} = \{&Trunk(x) \wedge isConnected(x, y) \wedge Leafs(y) \Rightarrow Tree(x, y)\}\\ \mathcal{A} = \{&(region_1 : GreenColored) \geq 0.3, (region_1 : RegularTextured) \geq 0.3,\\ &(region_2 : BrownColored) \geq 0.3, isConnected\,(region_2, region_1) \geq 1\}\end{aligned}$$

where we have a region in our image that is characterized as $GreenColored$ and $RegularTextured$ with a degree at least 0.3, a region that is characterized as $BrownColored$ with a degree at least 0.3 and the two regions are connected together.

Table 1 Semantics of fuzzy concept and role descriptions

Constructor	Syntax	Semantics
Top	$\top$	$\top^{\mathcal{I}}(\mathrm{a}) = 1$
Bottom	$\perp$	$\perp^{\mathcal{I}}(\mathrm{a}) = 0$
General negation	$\neg C$	$(\neg C^{\mathcal{I}})(a) = 1 - C^{\mathcal{I}}(a)$
Conjunction	$C \sqcap \mathrm{D}$	$(C \sqcap \mathrm{D}^{\mathcal{I}}(a)) = \min(\mathcal{C}^{\mathcal{I}}(a), D^{\mathcal{I}}(a))$
Disjunction	$\mathcal{C} \sqcup D$	$(\mathcal{C} \sqcup D)^{\mathcal{I}}(a) = \max(\mathcal{C}^{\mathcal{I}}(a), D^{\mathcal{I}}(a))$
Value restriction	$\forall R.\mathcal{C}$	$(\forall R, \mathcal{C}^{\mathcal{I}}(a)) = \inf_{b \in \Delta^{\mathcal{I}}} \max(1 - R^{\mathcal{I}}(a.b), \mathcal{C}^{\mathcal{I}}(b))$
Exists restriction	$\exists R.\mathcal{C}$	$(\exists R.\mathcal{C}^{\mathcal{I}}(a)) = \sup_{b \in \Delta^{\mathcal{I}}} \min(R^{\mathcal{I}}(a.b), \mathcal{C}^{\mathcal{I}}(b))$
At-most	$\geq mR$	$(\geq mR)^{\mathcal{I}}(a) = \underset{b_1,\ldots,b_m \in \Delta^{\mathcal{I}}}{\mathrm{sub}} \min_{i=1}^{m}\{R^{\mathcal{I}}(a, b_i)\}$
At-least	$\leq mR$	$(\leq mR)^{\mathcal{I}}(a) = \underset{b_1,\ldots,b_m+1 \in \Delta^{\mathcal{I}}}{\inf} \max_{i=1}^{m+1}\{1 - R^{\mathcal{I}}(a, b_i)\}$
Role conjunction	$p_1 \sqcap \cdots \sqcap p_k$	$(p_1 \sqcap \cdots \sqcap p_k)^{\mathcal{I}}(a, b) = \min(p_1^{z}(a, b), \ldots, p_k^{\mathcal{I}}(a, b))$

3.2 Semantics

The semantics of the terminological component are given via fuzzy interpretations that use membership functions ranging 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:

- An individual name $a \in \mathbf{I}$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$.
- A concept name $A \in \mathbf{C}$ to a membership function $A^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0, 1]$.
- A role name $P \in \mathbf{R}$ to a membership function $P^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0, 1]$.
- An ordinary predicate $q \in \mathbf{Q}$ of κ-arity to a membership function $q^{\mathcal{I}} : \left(\Delta^{\mathcal{I}}\right)^{\kappa} \to [0, 1]$.
- And satisfies the unique names assumption, i.e. for each tuple of different elements $a, b \in \mathbf{I}$, $a^{\mathcal{I}} \not\doteq b^{\mathcal{I}}$ holds.

The semantics of concept and role descriptions are given by the equations in Table 1 where $a, b \in \Delta^{\mathcal{I}}$, C, D are concept descriptions, A is a concept name, R is a role conjunction of the form $P_1 \sqcap \cdots \sqcap P_k$ and $P_1, \ldots, P_k$ are role names in $\mathbf{R}$.

Terminological component satisfiability:

An interpretation, $\mathcal{I}$, satisfies the terminological component $\mathcal{T}$ iff for every element $a \in \Delta^{\mathcal{I}}$ and concept inclusion axiom $C \sqsubseteq D$ in $\mathcal{T}$ it applies that

$$C^{\mathcal{I}}(a) \leq D^{\mathcal{I}}(a)$$

Horn rule satisfiability:

An interpretation, $\mathcal{I}$, satisfies a Horn rule $p_1\left(\overline{X}_1\right) \wedge \cdots \wedge p_k\left(\overline{X}_k\right) \Rightarrow q\left(\overline{Y}\right)$ iff for every mapping ψ from the variables and individuals of $\overline{X}_1, \ldots, \overline{X}_k, \overline{Y}$ to the elements of $\Delta^{\mathcal{I}}$, where each individual a is mapped to $a^{\mathcal{I}}$,

$$\min\left(p_1^{\mathcal{I}}\left(\psi\left(\overline{X_1}\right)\right), \ldots, p_k^{\mathcal{I}}\left(\psi\left(\overline{X_k}\right)\right)\right) \leq q\left(\psi\left(\overline{Y}\right)\right)$$

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 every fuzzy assertion in $\mathcal{A}$. In this case, we say $\mathcal{I}$ is *a model* of $\mathcal{A}$, denoted as $\mathcal{I} \models \mathcal{A}$. If $\mathcal{A}$ has a model we then say that $\mathcal{A}$ is *consistent*. Given a fuzzy interpretation $\mathcal{I}$ we say that

– $\mathcal{I}$ satisfies $(a : C) \bowtie n$ iff $C^{\mathcal{I}}\left(a^{\mathcal{I}}\right) \bowtie n$,
– $\mathcal{I}$ satisfies $(\langle a, b\rangle : P) \rhd n$ iff $P^{\mathcal{I}}\left(a^{\mathcal{I}}, b^{\mathcal{I}}\right) \rhd n$,
– $\mathcal{I}$ satisfies $(\langle a_1, \dots, a_k\rangle : q) \rhd n$ iff $q^{\mathcal{I}}\left(a_1^{\mathcal{I}}, \dots, a_k^{\mathcal{I}}\right) \rhd n$.

Knowledge base satisfiability:

An fuzzy 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 and role inclusion in $\mathcal{T}$, as well as each Horn rule in $\mathcal{H}$. A fuzzy 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 the knowledge base K and denoted as $\mathcal{I} \models K$.

Positive inequality normal, negation normal, normalized form:

Before applying a fuzzy CARIN reasoning algorithm, we consider that each concept assertion is in its positive inequality normal, negation normal, normalized form. Specifically, only role assertions of the form $(\langle a, b\rangle : P) \geq n$, ordinary predicate assertions of the form $(\overline{a} : p) \geq n$, and concept assertions of the form $(a : C) \geq n$ are allowed where C is in its negation normal form. A fuzzy CARIN ABox $\mathcal{A}$ can be transformed to this form in the following steps:

Step 1: Negative assertions are transformed into their Positive Inequality Normal Form (PINF) by applying the fuzzy complement in both sides of the inequality as described in [36]. For example $(a : C) \leq n$ and $(a : C) < n$ are being transformed into $(a : \neg C) \geq 1 - n$ and $(a : \neg C) > 1 - n$.

Step 2: Concepts are transformed into their Negation Normal Form. A concept can be transformed into its NNF by pushing negations inwards making use of the following concept equivalences [33,35]:

$$\begin{aligned} \neg(C \sqcup D) &\equiv (\neg C \sqcap \neg D) & \qquad \neg(C \sqcap D) &\equiv (\neg C \sqcup \neg D) \\ \neg\exists R.C &\equiv \forall R.(\neg C) & \neg\forall R.C &\equiv \exists R.(\neg C) \\ \neg \geq m_1 R &\equiv\ \leq (m_1 - 1)R & \neg \leq m_2 R &\equiv\ \geq (m_2 + 1)R \\ \neg\neg C &\equiv C & & \end{aligned}$$

where $m_1 \in \mathbb{N}^*$ and $m_2 \in \mathbb{N}$ in the above equations.

Step 3: 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 [34] 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 proved in [34] that each model $\mathcal{I}$ of K is also a model of K's normalized form and vice-versa.

Finally, following [33] we introduce a conjugated pair of fuzzy assertions. 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 $\neg\phi \equiv a : \neg C$), a pair of fuzzy assertions in their transformed PINF, negation normal, normalized form is conjugated if it is of the form $\phi \geq n, \neg\phi \geq m$ where $n + m > 1$. An ABox $\mathcal{A}$ with a conjugated pair of fuzzy assertions has no model $\mathcal{I}$.

3.3 Conjunctive queries over fuzzy dLs

The most common inference problems addressed by previous fuzzy DL systems are the satisfiability, n-satisfiability, subsumption and the entailment problem [35]. It has been proved in [33,35] that each one of the previous problems can be reduced to the problem of a knowledge base satisfiability.

Another interest family of inference problems, interwoven with relational databases, is constituted of the conjunctive query, the union of conjunctive queries, and the existential entailment problems. Although there has been quite a few work on fuzzy SQL [6] and querying fuzzy DLs [29,38], as far as we know, no such definition of conjunctive queries or unions of conjunctive queries exists for fuzzy DL knowledge bases. Following [28], we present the definition of the conjunctive query problem and extend it accordingly in order to provide a proper definition for fuzzy DLs.

Definition 3.1 (*Conjunctive Query*) A conjunctive query (CQ) over a knowledge base K is a set of atoms of the form

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

where $p_1, \ldots, p_k$ are either concept descriptions, or 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 3.2 (*Union of Conjunctive Queries*) A union of conjunctive queries (UCQ) over a knowledge base K is a set of conjunctive queries:

$$UCQ = \{CQ_1, \ldots, CQ_\kappa\}$$

where each CQ_i, for $1 \leq i \leq \kappa$, is a conjunctive query.

To say that Q is either a CQ or a 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. We may use the expression vars $(Q_1, \ldots, Q_l)$ as an abbreviation for vars(Q_1) $\cup \cdots \cup$ vars(Q_l) (the same applies for varsIndivs, indivs). In a similar way we define the sets vars $\left(\overline{Y}\right)$, varsIndivs $\left(\overline{Y}\right)$, indivs $\left(\overline{Y}\right)$ for a tuple of variables and individuals $\overline{Y}$.

Queries are interpreted in the standard way. For a CQ, we say that $\mathcal{I}$ satisfies CQ, written $\mathcal{I} \models CQ$, iff there exists a mapping σ : varsIndivs(CQ) $\rightarrow \Delta^{\mathcal{I}}$ such that:

$$\begin{array}{lll} \sigma(a) & = a^{\mathcal{I}} & \text{for each } a \in \text{indivs}(CQ) \\ p_i^{\mathcal{I}}(\sigma(\overline{Y_i})) \geq n & & \text{for each } p(\overline{Y_i}) \geq n \text{ in } CQ \end{array} \tag{1}$$

For a union of conjunctive queries $UCQ = \{CQ_1, \ldots, CQ_l\}, \mathcal{I} \models UCQ$ iff $\mathcal{I} \models CQ_i$ for some $CQ_i \in UCQ$. 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 3.3 (*Query Entailment*) Let K be a knowledge base and Q a *query*. The query entailment problem is to decide whether $K \models Q$.

Example 3.2 For the knowledge base presented in Example 3.1 and the union of conjunctive queries:

$$UCQ = \{Tree(x, y) \geq 0.3, Mountain(x) \geq 0.4\}$$

we want to know if $KB \models UCQ$ i.e. if our knowledge always implies that there exists a tree or a mountain in our image, with a degree of certainty at least 0.3, 0.4 respectively.

It is important to notice that the query entailment problem, 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. Next we are going to present a definition for the existential entailment problem for fuzzy DLs. This definition is an adaptation of the existential entailment problem presented in [24].

Definition 3.4 (*Existential Entailment*) Let $\mathcal{T}$ be a TBox in the DL fuzzy $\mathcal{ALCNR}$ and let $\beta, Q_1, \ldots, Q_m$ be sentences of the form

$$\left(\exists\overline{Y}\right) p_1\left(\overline{Y}_1\right) \geq n_1 \wedge \cdots \wedge p_k\left(\overline{Y}_k\right) \geq n_k$$

where $p_1, \ldots, p_k$ are either roles names in $\mathbf{R}$, or concept descriptions and $\overline{Y}, \overline{Y}_1, \ldots, \overline{Y}_k$ are tuples of variables and individuals in $\mathbf{I}$ such that vars $\left(\overline{Y}\right) \subseteq$ vars $\left(\overline{Y}_1, \ldots, \overline{Y}_k\right)$.

The variables that do not appear existentially quantified in Q or β are considered universally quantified. Any universally quantified variable that appears in one of the Q_i's must also appear in β. The existential entailment problem is to decide whether:

$$\langle\beta, \mathcal{T}\rangle \models \{Q_1, \ldots, Q_m\}$$

In order to give semantics for the existential entailment problem, we must first define a model $\mathcal{I}$ for β. A fuzzy interpretation for β 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: each individual name $a \in \text{indivs}(\beta)$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each variable name $x \in \text{vars}(\beta)$ to an element $x^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, each concept name $A \in \mathbf{C}$ to a membership function $A^{\mathcal{I}} : \Delta^{\mathcal{I}} \to [0, 1]$, each role name $P \in \mathbf{R}$ to a membership function $P^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to [0, 1]$. The interpretation $\mathcal{I}$ must satisfy the unique name assumption for individuals but not necessarily for variables i.e. for each tuple of elements $a, b \in \text{indivs}(\beta) a^{\mathcal{I}} \not\approx b^{\mathcal{I}}$ holds. We say that $\mathcal{I}$ is a model of β if it holds that: $C^{\mathcal{I}}(\upsilon^{\mathcal{I}}) \geq n$ for each conjunct $C(\upsilon) \geq n$ in β and $P^{\mathcal{I}}(\upsilon^{\mathcal{I}}, \omega^{\mathcal{I}}) \geq n$ for each conjunct $P(\upsilon, \omega) \geq n$ in β where C is a concept name, P is a role name, and $\upsilon, \omega \in \text{varsIndivs}(\beta)$. We see that β has semantics similar to that of an ABox $\mathcal{A}$, presented in Sect. 3.2, with the main difference that the unique name assumption does not apply for variables. $\mathcal{I}$ is a model of β w.r.t $\mathcal{T}$, i.e. $\mathcal{I} \models \langle\beta, \mathcal{T}\rangle$ iff it is a model of β and satisfies each concept inclusion in $\mathcal{T}$.

An existential entailment of the form $\langle\beta, \mathcal{T}\rangle \models \{CQ_1, \ldots, CQ_m\}$ holds iff for every interpretation $\mathcal{I}$ that satisfies β w.r.t. $\mathcal{T}$ there exists a mapping from the variables and individuals of some $Q_i \in \{CQ_1, \ldots, CQ_m\}$ (CQ_i has the form of a sentence as described in Definition 3.4) to the elements of our domain of interpretation $\tau : \text{varsIndivs}(Q_i) \to \Delta^{\mathcal{I}}$ such that it holds:

$$\begin{array}{ll} \tau(a) = a^{\mathcal{I}} & \text{for every individual } a \in \text{indivs}(\beta) \\ \tau(x) = x^{\mathcal{I}} & \text{for every universally quantified variable } x \in \text{vars}(\beta) \\ p_i^{\mathcal{I}}\left(\tau\left(\overline{Y}_i\right)\right) \geq n_i & \text{for each } i \in 1, \ldots, k \end{array} \tag{2}$$

It can be checked that the problem of answering to union of conjunctive queries free of ordinary predicates is a special case of the existential entailment problem where $\mathcal{A}$ corresponds to a β sentence with no variables. This fact indicates that the variables in each of the CQ_is are all existentially quantified.

The existential entailment problem can be used as a sound and complete algorithm for query containment over $\mathcal{ALCNR}$ and therefore for query simplification over a complex union of conjunctive queries. Detecting and dealing with redundancy is an ubiquitous problem in query optimization and a hot research topic for different research areas such as Relational Databases [5] and OWL-DL inference engines [20]. Suppose that we have a knowledge base $K = \langle \mathcal{T}, \mathcal{A} \rangle$ and we want to answer to the union of conjunctive queries $UCQ = \{CQ_1, \ldots, CQ_m\}$. If it holds that $\langle CQ_1, \mathcal{T} \rangle \models CQ_2$ we can reduce the problem to that of answering to the union of conjunctive queries $UCQ \setminus CQ_2$.

4 Reasoning in fuzzy carin

Our main goal is to provide a sound and complete algorithm for answering to the UCQ and the existential entailment problem. The algorithm presented in [24] for the crisp CARIN is based on constraint systems. A constraint system is a non-empty set of constraints of the form $s : C, sPt, \forall x.x : C$, and $s \neq t$. We follow a different approach providing an algorithm based on completion forests. A completion forest and a constraint system are both abstractions of an interpretation $\mathcal{I}$ and they are used to prove the existence of a model of a knowledge base K. It is easy to prove that there is an equivalence between algorithms based on constraint systems and completion forests. Nevertheless, we chose an algorithm based on completion forests in order to exploit the rich bibliography [32,35] of sound and complete inference procedures for fuzzy DL problems based on completion forests.

To say that a knowledge base implies a query, $K \models Q$, 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. Our algorithm for answering to the UCQ problem is performed in three steps.

- In the first step we build a set of completion forests $\text{ccf}(\mathbb{F}_K^q)$ according to the rules presented in Table 2 for a knowledge base $K = \langle \mathcal{T}, \mathcal{A} \rangle$, and by applying the q-blocking condition (see Definition 4.7) for the query UCQ. According to Theorem 4.1 each clash free completion forest implies the existence of a model of our knowledge and the existence of a model of K implies the existence of a clash free completion forest.
- In the second step, we are called to answer if each model $\mathcal{I}$ of $K = \langle \mathcal{T}, \mathcal{A} \rangle$ satisfies the UCQ. We prove that this is the case iff for each completion forest $\mathcal{F} \in \text{ccf}(\mathbb{F}_K^q)$ there exists a mapping from the variables and individuals of at least one $CQ \in UCQ$ to the nodes of $\mathcal{F}$ such that each role and concept restriction in CQ is satisfied in $\mathcal{F}$. The existential entailment problem can be easily reduced to the problem of answering to UCQ.
- In the two previous cases, we considered knowledge bases containing no Horn rule component and therefore UCQ containing no ordinary predicates. If our UCQ contains ordinary predicates, a prepossessing step is applied in order to reduce the problem of answering to a UCQ containing ordinary predicates to the problem of answering to a UCQ with no ordinary predicates.

In Sect. 4.1, we present a fuzzy tableau for fuzzy $\mathcal{ALCNR}$ which is an intermediate form of representation between a fuzzy interpretation $\mathcal{I}$ and a completion forest $\mathcal{F}$. In Sect. 4.2, we present an algorithm for consistency checking in $\mathcal{ALCNR}$ based on completion forests. On these foundations an algorithm for the inference problems defined in Sect. 3.3 will be presented in Sect. 4.3.

Table 2 Tableaux expansion rules for fuzzy $\mathcal{ALCNR}$

Rule	Description		
$\sqcap_{\geq}$	if	1.	$\langle C_1 \sqcap C_2, \geq, n\rangle \in \mathcal{L}(x)$,
		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.	$\langle C_1 \sqcup C_2, \geq, n\rangle \in \mathcal{L}(x)$,
		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 $\mathcal{L}(\langle x, y\rangle) = \{\langle P_i, \geq, n\rangle\}$, $\mathcal{L}(y) = \{\langle C, \geq, n\rangle\}$ for all $i \in \{1, \ldots, k\}$ for the role conjunction $R \rightarrow P_1 \sqcap \ldots \sqcap P_k$
$\forall_{\geq}$	if	1.	$\langle \forall R.C, \geq, n\rangle \in \mathcal{L}(x)$,
		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 $y_1, \ldots, y_m$ 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 $\mathcal{L}(\langle x, y_i\rangle) = \{\langle P_{i'}, \geq, n\rangle\}$ and $y_i \neq y_j$ for $1 \leq i < j \leq m$ and for all $i' \in \{1, \ldots, k\}$ for the role conjunction $R \rightarrow P_1 \sqcap \ldots \sqcap P_k$
$\leq_{\geq}$	if	1.	$\langle \leq mR, \geq, n\rangle \in \mathcal{L}(x)$,
		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	1.	$\mathcal{L}(z) \rightarrow \mathcal{L}(z) \cup \mathcal{L}(y)$
		2.	$\mathcal{L}((\langle x, z\rangle) \rightarrow \mathcal{L}(\langle x, z\rangle) \cup \mathcal{L}(\langle x, y\rangle)$
		3.	$\mathcal{L}(\langle x, y\rangle) \rightarrow \emptyset$,
		4.	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 $n \in N^{\mathcal{A}}$ [1]
		then	$\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{E\}$ for some $E \in \{\langle \neg C, \geq, 1 - n + \epsilon\rangle, \langle D, \geq, n\rangle\}$

4.1 A tableau for fuzzy $\mathcal{ALCNR}$

Tableaux algorithms check for consistency by trying to build a fuzzy tableau for $\mathcal{A}$ w.r.t. $\mathcal{T}$, that is, an abstraction of a model of our knowledge K. In this Section we provide the tableau for fuzzy $\mathcal{ALCNR}$. Without loss of generality we consider that concept assertions are in their positive inequality normal, negation normal, normalized form as described in Sect. 3.2.

Definition 4.1 If $\mathcal{A}$ is a fuzzy $\mathcal{ALCNR}$ ABox, $\mathbf{R}$ is the set of role names occurring in K, $\mathbf{I}$ is the set of individuals in K and $\mathcal{T}$ is a TBox, a fuzzy tableau T for $\mathcal{A}$ w.r.t. $\mathcal{T}$ is defined to be a quadruple $(\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{V})$ such that: $\mathbf{S}$ is a set of elements, $\mathcal{L} : \mathbf{S} \times \mathrm{sub}(K) \rightarrow [0, 1]$ maps each pair of an element and a concept in sub(K) to the membership degree of that element to the concept, $\mathcal{E} : \mathbf{R} \times \mathbf{S} \times \mathbf{S} \rightarrow [0, 1]$ maps each role in $\mathbf{R}$ and pair of elements to the membership degree of the pair to the role, and $\mathcal{V} : \mathbf{I} \rightarrow \mathbf{S}$ maps individuals occurring in $\mathcal{A}$ to elements of $\mathbf{S}$.

For all $s, t \in \mathbf{S}$, $C, E \in \text{sub}(K)$, $n \in [0, 1]$, $P, P_1, \ldots, P_k \in \mathbf{R}$ and R a role conjunction of the form $P_1 \sqcap \cdots \sqcap P_k$, T satisfies:

1. $\mathcal{L}(s, \bot) = 0$ and $\mathcal{L}(s, \top) = 1$ for all $s \in \mathbf{S}$,
2. If $\mathcal{L}(s, \neg A) \geq n$, then $\mathcal{L}(s, A) \leq 1 - n$,
3. If $\mathcal{L}(s, C \sqcap E) \geq n$, then $\mathcal{L}(s, C) \geq n$ and $\mathcal{L}(s, E) \geq n$,
4. If $\mathcal{L}(s, C \sqcup E) \geq n$, then $\mathcal{L}(s, C) \geq n$ or $\mathcal{L}(s, E) \geq n$,
5. If $\mathcal{L}(s, \forall R.C) \geq n$ then for all $t \in \mathbf{S}$ it holds either that $\mathcal{E}(P_i, \langle s, t\rangle) \leq 1 - n$ for some $i \in \{1, \ldots, k\}$, or $\mathcal{L}(t, C) \geq n$,
6. If $\mathcal{L}(s, \exists R.C) \geq n$, then there exists some $t \in \mathbf{S}$ such that $\mathcal{E}(P_i, \langle s, t\rangle) \geq n$ for all $i \in \{1, \ldots, k\}$ and $\mathcal{L}(t, C) \geq n$,
7. If $\mathcal{L}(s, \geq p\ R) \geq n$, then $\sharp R^T(s, \geq, n) \geq p$,
8. If $\mathcal{L}(s, \leq p\ R) \geq n$, then $\sharp R^T(s, \geq, 1 - n + \epsilon) \leq p$,
9. If $C \sqsubseteq D \in \mathcal{T}$, then either $\mathcal{L}(s, \neg C) \geq 1 - n + \epsilon$, or $\mathcal{L}(s, D) \geq n$ for all $s \in \mathbf{S}$ and $n \in N_{\mathcal{A}}$,
10. If $(a : C) \geq n \in \mathcal{A}$, then $\mathcal{L}(\mathcal{V}(a), C) \geq n$,
11. If $(\langle a, b\rangle : P) \geq n \in \mathcal{A}$, then $\mathcal{E}(P, \langle \mathcal{V}(a), \mathcal{V}(b)\rangle) \geq n$,
12. For each $a, b \in \mathbf{I}$, $\mathcal{V}(a) \neq \mathcal{V}(b)$ holds,

Where $\sharp$ denotes the cardinality of a set. For a role description $R \rightarrow P_1 \sqcap \cdots \sqcap P_k$, $R^T(s, \geq, n) = \{t \in \mathbf{S} \mid \mathcal{E}(P_1, \langle s, t\rangle) \geq n, \ldots, \mathcal{E}(P_k, \langle s, t\rangle) \geq n\}$ is the subset of $\mathbf{S}$ containing all the elements connected from s through all P_is with a degree greater or equal than n. Moreover sub(C) denotes the set of sub-concepts of a concept C and sub(K) denotes the set of sub-concepts that appear in a knowledge base K. Accordingly sub(Q) denotes the set of sub-concepts appearing in a query Q. Finally $N_{\mathcal{A}}$ is the set of degrees appearing in $\mathcal{A}$.

Lemma 4.1 *A fuzzy $\mathcal{ALCNR}$ ABox $\mathcal{A}$ is consistent w.r.t. $\mathcal{T}$ iff there exists a fuzzy tableau for $\mathcal{A}$ w.r.t. $\mathcal{T}$.*

4.2 $\mathcal{ALCNR}$ completion forests

The completion forest introduced is based on the completion forest presented in [24]. As in [24] 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 $\mathcal{ALCNR}$ [4] our algorithm adopts the q-blocking condition, introduced in [24], 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 4.2 (*Completion Tree*) A completion tree for fuzzy $\mathcal{ALCNR}$ is a tree all nodes of which are variable nodes, except from the root node which corresponds to an individual. Each node x is labeled with a set of triples:

$$\mathcal{L}(x) = \{\langle C, \geq, n\rangle \mid C \in \text{sub}(K), n \in [0, 1]\}$$

Each edge is labeled with a set of triples:

$$\mathcal{L}(x, y) = \{\langle P, \geq, n\rangle \mid P \in \mathbf{R}, n \in [0, 1]\}$$

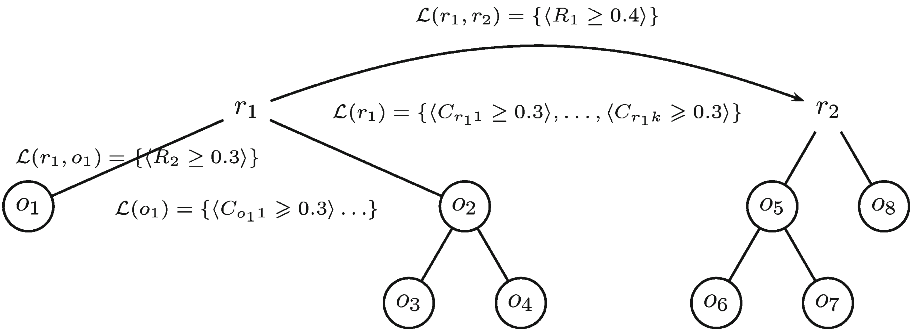

Fig. 1 A fuzzy $\mathcal{ALCNR}$ completion forest

Definition 4.3 (*Completion Forest*) A completion forest $\mathcal{F}$ is a collection of trees whose roots, corresponding to individuals, are arbitrarily connected by arcs. As before, edges between root nodes are labeled with the set

$$\mathcal{L}(x, y) = \{\langle P, \geq, n\rangle \mid P \in \mathbf{R}, n \in [0, 1]\}$$

Intuitively each triple $\langle C, \geq, n\rangle$ (or $\langle P, \geq, n\rangle$), called membership triple, represents the membership degree and the type of assertion of each node (or pair of nodes) to a concept $C \in sub(K)$ (or role $P \in \mathbf{R}$).

Example 4.1 In Fig. 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 labeled with a set of concepts with degrees and each edge must be labeled 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 labeled due to space limitations.

Definition 4.4 (*nodes, vars, $R_{\geq n}$-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) for the role conjunction $R \to P_1 \sqcap \cdots \sqcap P_k$, w is an *$R_{\geq n}$-successor* of υ when nodes υ and w are connected by an edge $\langle \upsilon, w\rangle$ such that $\langle P_i, \geq, n_i\rangle \in \mathcal{L}(x, y)$ with $n_i \geq n$ for all $i \in \{1, \ldots, k\}$, (iv) υ is a *successor* of w when υ is an *$R_{\geq n}$-successor* of w with $n > 0$, (v) descendant is the transitive closure of successor.

Note that our definition of $R_{\geq n}$-successor indicates that if υ is an $R_{\geq n}$-successor of w then it is also an $R_{\geq n'}$-successor of w for $n' < n$.

Example 4.2 In Fig. 1, o_1 is a $R_{2_{\geq 0.3}}$ successor of r_1.

Definition 4.5 (*q-tree equivalence*) The *q-tree* of a variable υ is the tree that includes the node υ and its successors whose distance from υ is at most q direct-successors arcs. We denote the set of nodes in the q-tree of υ by $V_q(\upsilon)$. Two nodes $\upsilon, w \in \mathcal{F}$ are said to be q-tree equivalent in $\mathcal{F}$ if there exists an isomorphism $\psi : V_q(\upsilon) \to V_q(w)$ such that (i) $\psi(\upsilon) = w$, (ii) for every $s \in V_q(\upsilon)$, $\langle C, \geq, n\rangle \in \mathcal{L}(s)$ iff $\langle C, \geq, n\rangle \in \mathcal{L}(\psi(s))$ (iii) for every $s, t \in V_q(\upsilon)$, $\langle P, \geq, n\rangle \in \mathcal{L}(\langle s, t\rangle)$ iff $\langle P, \geq, n\rangle \in \mathcal{L}(\langle \psi(s), \psi(t)\rangle)$. Intuitively, two variables are q-tree equivalent if the trees of depth q of which they are roots are isomorphic.

Definition 4.6 (*q-Witness*) A node υ is the *q-witness* of a node w when (i) υ is an ancestor of w, (ii) υ and w are q-tree equivalent, (iii) $w \notin V_q(\upsilon)$.

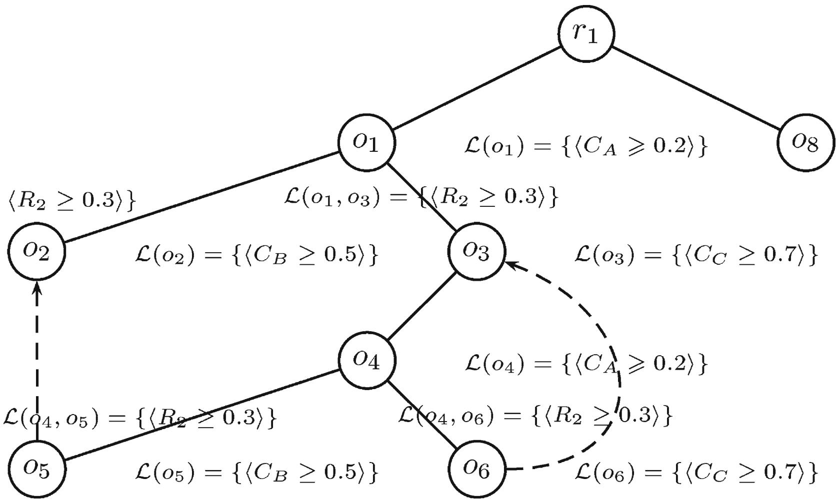

Fig. 2 Blocking example

Definition 4.7 (*q-blocking*) A node x is q-blocked either when it is the leaf of a q-tree in $\mathcal{F}$ whose root w has a q-witness v and $w \in \text{vars}(\mathcal{F})$, or when $\mathcal{L}(x) = \emptyset$. From now on when referring to blocking we indicate q-blocking.

Example 4.3 In Fig. 2 o_1 is a 1-witness of o_4 since the 1-tree of o_1 is equivalent to 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.

Definition 4.8 (*Clash free completion forest*) For a node x, $\mathcal{L}(x)$ contains a *clash* if it contains: (i) A conjugated pair of triples (a conjugated pair of triples can be defined straightforwardly by the definition of a conjugated pair of fuzzy assertions described in Sect. 3.2), (ii) one of the triples $\langle \perp, \geq, n\rangle$ with $n > 0$, $\langle C, \geq, n\rangle$ with $n > 1$, (iii) some triple $\langle \leq pR, \geq, n\rangle$ and x has $p + 1 R_{\geq n}$-successors $y_0, \ldots, y_p$ such that $y_i \neq y_j$ 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 $\mathcal{A}$, labeled 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 $(\langle a_i, a_j\rangle : P) \geq n \in \mathcal{A}$ labeled with $\mathcal{L}(\langle x_0^i, x_0^j\rangle)$ such that $\{\langle P, \geq n\rangle\} \subseteq \mathcal{L}(\langle x_0^i, x_0^j\rangle)$, (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. $\mathcal{F}_K$ is expanded by repeatedly applying the completion rules from Table 2.

In Table 2 rules $\sqcap_{\geq}, \sqcup_{\geq}, \exists_{\geq}, \forall_{\geq}$ are first introduced in [35] and then modified for completion forests in [31], rules $\geq_{\geq}$ and $\leq_{\geq}$ are presented in [33], while rule $\sqsubseteq$ is first introduced in [34]. The $\leq_{r\geq}$ presented in [33] cannot be applied since $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ holds for every pair of individuals $a, b \in \mathbf{I}$.

Definition 4.9 (*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 2 to $\mathcal{F}_K$. A completion forest $\mathcal{F}$ is

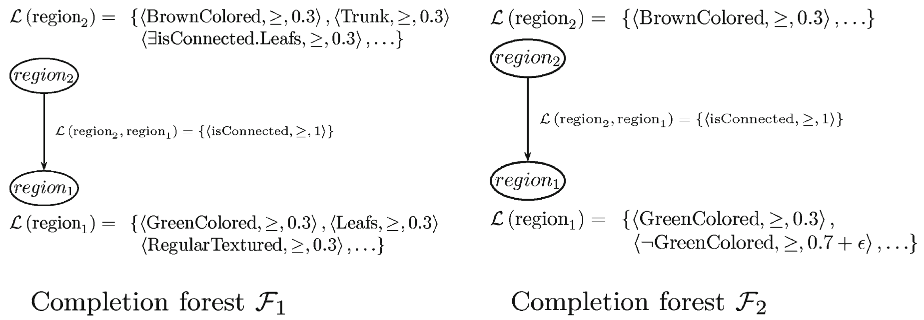

Fig. 3 Completion forests for the knowledge base K presented in Example 3.1. The above completion forest $\mathcal{F}_1$ is complete and clash free while $\mathcal{F}_2$ contains a clash

q-complete when none of the rules in Table 2 can be applied to it. We denote by $\text{ccf}(\mathbb{F}_K^q)$ the set of completion forests in $\mathbb{F}_K$ that are q-complete and clash free.

Lemma 4.2 (Termination) *For each fuzzy $\mathcal{ALCNR}$ ABox $\mathcal{A}$ and TBox $\mathcal{T}$, the tableaux algorithm terminates when started for $\mathcal{A}$ and $\mathcal{T}$.*

Lemma 4.3 (Soundness) *If the expansion rules can be applied to a fuzzy $\mathcal{ALCNR}$ ABox $\mathcal{A}$ and TBox $\mathcal{T}$ such that they yield a complete and clash-free completion forest, then $\mathcal{A}$ has a fuzzy tableau w.r.t $\mathcal{T}$.*

Lemma 4.4 (Completeness) *Let $\mathcal{A}$ be a fuzzy $\mathcal{ALCNR}$ ABox and $\mathcal{T}$ a TBox. If $\mathcal{A}$ has a fuzzy tableau w.r.t. $\mathcal{T}$, then the expansion rules can be applied to $\mathcal{A}$ and $\mathcal{T}$ in such a way that the tableau algorithm yields a complete and clash-free completion forest.*

According to Lemmas 4.1, 4.3 and 4.4, we have that:

Theorem 4.1 *Each class free completion forest $\mathcal{F} \in \text{ccf}(\mathbb{F}_K^q)$ corresponds to a model $\mathcal{I}$ of K and vice-versa.*

Example 4.4 For the DL part of the knowledge base presented in Example 3.1 we have a set of completion forests $\mathbb{F}_K^q$ after the application of the expansion rules of Table 2. Two of them are presented in Fig. 3. As we can see $\mathcal{F}_2$ contains a clash since it contains two conjugated triples (related to the *GreenColored* concept). Therefore only $\mathcal{F}_1 \in \text{ccf}(\mathbb{F}_K^q)$ while $\mathcal{F}_2 \notin \text{ccf}(\mathbb{F}_K^q)$.

In Sect. 4.3 we show how the set $\text{ccf}(\mathbb{F}_K^q)$ can be exploited in order to answer to unions of conjunctive queries.

4.3 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, that examines the finite set of clash free completion forests $\text{ccf}\left(\mathbb{F}_K^q\right)$. Our algorithm is first presented for unions of conjunctive queries free of ordinary predicates and a Horn rule component $K = \langle \mathcal{T}, \mathcal{A} \rangle$ and then extended for query answering with ordinary predicates.

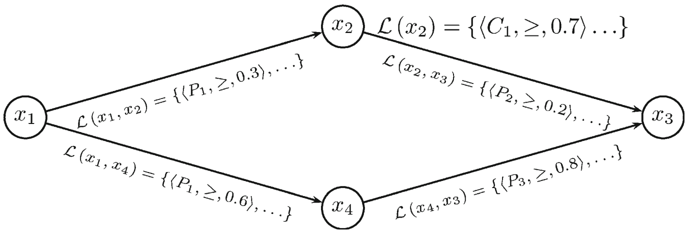

Fig. 4 Conjunctive query mapped to a graph

4.3.1 Answering to union of conjunctive queries free of ordinary predicates

Following [28], in order to have a complete algorithm for answering to conjunctive queries we must add to our $TBox$ the rule:

$$C \sqsubseteq C \tag{3}$$

for each concept name C appearing in a conjunctive query. This ensures that in each completion forest either $(x : C) \geq n$ or $(x : C) < n$[2] holds and consequently it can be checked if a node can be mapped to a variable of our conjunctive query.

Additionally we have to show why q-blocking is adopted instead of simple blocking. A conjunctive query CQ as presented in Definition 3.1 can be mapped to a graph G_{CQ} whose nodes correspond to variables and individuals, each node x is labeled with a set $\mathcal{L}(x) = \{\langle C, \geq, n\rangle \mid C \in \text{sub}(CQ), n \in [0, 1]\}$ and each edge $\langle x, y\rangle$ is labeled with a set $\mathcal{L}(x, y) = \{\langle P, \geq, n\rangle \mid P \in \mathbf{R}, n \in [0, 1]\}$. Suppose that d_{xy} is the length of the *lengthiest acyclic,directed* 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 sub-tree 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 $|UCQ|$ coincidences with the value of the maximum $|CQ|$ i.e. $|UCQ| = \max\{|CQ| \mid CQ \in UCQ\}$.

Example 4.5 The conjunctive query:

$$\begin{aligned} CQ = \{&P_1(x_1, x_2) \geq 0.3, \quad C_1(x_2) \geq 0.7,\\ &P_2(x_2, x_3) \geq 0.2, \ P_1(x_1, x_4) \geq 0.6,\\ &P_3(x_4, x_3) \geq 0.8\} \end{aligned}$$

is represented by the graph in Fig. 4 and has $|CQ| = 2$.

Definition 4.10 Suppose that we have a conjunctive query:

$$\begin{aligned} CQ = \ &C_1(x_1) \geq n_1 \wedge \cdots \wedge C_k(x_k) \geq n_k\\ &\wedge P_{k+1}(y_{k+1}, z_{k+1}) \geq n_{k+1} \wedge \cdots \wedge P_\kappa(y_\kappa, z_\kappa) \geq n_\kappa \end{aligned}$$

For a completion forest $\mathcal{F}$ we say that $CQ \hookrightarrow \mathcal{F}$ iff there exists a mapping σ : varsIndivs$(CQ) \to$ nodes$(\mathcal{F})$ such that:

1. σ maps each individual $a \in \mathbf{I}$ to its corresponding root node,
2. $\langle C_i, \geq, n_i'\rangle \in \mathcal{L}(\sigma(x_i))$ for some $n_i' \geq n_i$, and

---

[2] $(x : \neg C) \geq 1 - n + \epsilon$ is its PINF and normalized form.

3. $\sigma(z_j)$ is an $(P_j)_{\geq n_j}$-successor of $\sigma(y_j)$

for each $1 \leq i \leq k$ and $k+1 \leq j \leq \kappa$. For a union of conjunctive queries $UCQ = \{CQ_1, \ldots, CQ_l\}$ we say that $UCQ \hookrightarrow \mathcal{F}$ iff $CQ_i \hookrightarrow \mathcal{F}$ for some $CQ_i \in UCQ$.

Definition 4.11 Suppose we have a conjunctive query:

$$\begin{aligned} CQ = \ & C_1(x_1) \geq n_1 \wedge \cdots \wedge C_k(x_k) \geq n_k \wedge \\ & P_{k+1}(y_{k+1}, z_{k+1}) \geq n_{k+1} \wedge \cdots \wedge P_\kappa(y_\kappa, z_\kappa) \geq n_\kappa \end{aligned}$$

For a fuzzy tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{V})$ we say that $CQ \hookrightarrow T$ iff there exists a mapping σ : varsIndivs$(CQ) \to \mathbf{S}$ such that

1. σ maps each individual $a \in \mathbf{I}$ to $\mathcal{V}(a)$,
2. $\mathcal{L}(\sigma(x_i), C_i) \geq n_i$, and
3. $\mathcal{E}\left(P_j, \langle\sigma(y_j), \sigma(z_j)\rangle\right) \geq n_j$

for each $1 \leq i \leq k$ and $k+1 \leq j \leq \kappa$. For a union of conjunctive queries $UCQ = \{CQ_1, \ldots, CQ_l\}$ we say that $UCQ \hookrightarrow T$ iff $CQ_i \hookrightarrow T$ for some $CQ_i \in UCQ$.

Lemma 4.5 *$UCQ \hookrightarrow T$ for every consistent tableau T w.r.t. $\mathcal{A}$ and $\mathcal{T}$, iff $\mathcal{I} \models UCQ$ for every model $\mathcal{I}$ of $K = \langle\mathcal{T}, \mathcal{A}\rangle$.*

Lemma 4.6 *If $UCQ \hookrightarrow T$ for every consistent tableau T w.r.t. $\mathcal{A}$ and $\mathcal{T}$, then $UCQ \hookrightarrow \mathcal{F}$ for every completion forest $\mathcal{F} \in ccf(\mathbb{F}_K^q)$.*

Lemma 4.7 *If $\mathcal{F} \in ccf(\mathbb{F}_K^q)$ and $Q \hookrightarrow \mathcal{F}$, then $Q \hookrightarrow T$ for every tableau with respect to $\mathcal{A}$ and $\mathcal{T}$.*

Theorem 4.2 *According to Lemmas 4.5–4.7 we have that a knowledge base K entails a union of conjunctive queries UCQ ($K \models UCQ$) iff $UCQ \hookrightarrow \mathcal{F}$ for every $\mathcal{F} \in$ ccf$(\mathbb{F}_K^q)$.*

Example 4.6 For example for the conjunctive query

$$CQ = Trunk(x) \geq 0.3 \wedge isConnected(x, y) \geq 0.3 \wedge Leafs(y) \geq 0.3.$$

and the completion forest $\mathcal{F}_1$ from Example 4.4 we have that the mapping σ, such that $\sigma(x) = region_2$ and $\sigma(y) = region_1$ satisfies $CQ \hookrightarrow \mathcal{F}_1$ where $\mathcal{F}_1 \in$ ccf $\left(\mathbb{F}_K^q\right)$. In order to prove that $K \models CQ$ it has to prove that $CQ \hookrightarrow \mathcal{F}$ for every $\mathcal{F} \in$ ccf $\left(\mathbb{F}_K^q\right)$—if we examine all completions forest we will see that this is the case.

4.3.2 Answering to conjunctive queries with ordinary predicates

Up till now we have presented an algorithm for the query entailment problem $K \models UCQ$ w.r.t a knowledge base $K = \langle\mathcal{T}, \mathcal{A}\rangle$ that has no Horn rule component (and subsequently UCQ does not contain any ordinary predicates). In this section, we will describe a procedure that reduces the problem of query entailment w.r.t. a knowledge base with a Horn rule component $K = \langle\mathcal{T}, \mathcal{H}, \mathcal{A}\rangle$, to a problem of query entailment w.r.t. a knowledge base with no Horn rule component $K = \langle\mathcal{T}, \mathcal{A}\rangle$. This procedure is performed in two steps.

In the first step, we get rid of ordinary predicate assertions. For each ordinary predicate q of arity $m > 1$ we introduce a set of pseudo roles $P_{q_1}, \ldots, P_{q_{m-1}}$ (for ordinary predicates of arity 1 it suffices to introduce a pseudo concept C_q). Then we create a new Horn rule component $\mathcal{H}'$ by adding to the original $\mathcal{H}$ a new Horn rule of the form

Algorithm 1 Reduction of a UCQ containing ordinary Predicates to a UCQ free of ordinary predicates.

procedure reduction(UCQ)

- if UCQ contains a conjunctive query CQ of the form

$$CQ = p_1\left(\overline{Y}_1\right) \geq n_1 \wedge \cdots \wedge q\left(\overline{Y}_q\right) \geq n_q \wedge \cdots \wedge p_k\left(\overline{Y}_k\right) \geq n_k$$

  where q is an ordinary predicate then:

  - create a new $UCQ' := UCQ \setminus \{CQ\}$
  - for each Horn rule of the form $r_1(\overline{X}_1) \wedge \cdots \wedge r_k(\overline{X}_k) \Rightarrow q(\overline{Y})$
    - create a mapping ψ such that:
      - $\psi\left(\overline{Y}\right) = \overline{Y}_q$,
      - $\psi(x) = x'$ where $x \in \text{vars}\left(\overline{X}_1, \ldots, \overline{X}_k\right) \setminus \text{vars}\left(\overline{Y}\right)$ and x' is a new variable name not existing in CQ,
      - $\psi(a) = a$ for each individual $a \in \mathbf{I}$.
    - create a conjunctive query CQ' from CQ where the conjunct $q\left(\overline{Y}_q\right) \geq n_q$ is replaced by $r_1\left(\psi\left(\overline{X}_1\right)\right) \geq n_q \wedge \cdots \wedge r_k\left(\psi\left(\overline{X}_k\right)\right) \geq n_q$.
    - $UCQ' := UCQ' \cup \{CQ'\}$.
  - return **reduction**(UCQ')
- else return UCQ

$P_{q_1}(x_1, x_2) \wedge \cdots \wedge P_{q_{m-1}}(x_{m-1}, x_m) \Rightarrow q(x_1, \ldots, x_m)$ where $x_1, \ldots, x_m$ are variable names. The initial ABox $\mathcal{A}$ is substituted by a new ABox $\mathcal{A}'$ where each assertion about q is substituted by a set of assertions about $P_{q_1}, \ldots, P_{q_{m-1}}$. For example an assertion about an ordinary predicate $q(a_1, \ldots, a_m) \geq n$ can be substituted by a set of pseudo role assertions $\mathcal{A}_q = \left\{P_{q_1}(a_1, a_2) \geq n, \ldots, P_{q_{m-1}}(a_{m-1}, a_m) \geq n\right\}$. It can be easily checked that for $K = \langle \mathcal{T}, \mathcal{H}, \mathcal{A} \rangle$ and $K' = \left\langle \mathcal{T}, \mathcal{H}', \mathcal{A}' \right\rangle$ it holds that $K \models UCQ$ iff $K' \models UCQ$.

In the second step, we iteratively replace a conjunctive query that contains an ordinary predicate q with a union of conjunctive queries where the ordinary predicate q is substituted by concepts, roles, and ordinary predicates that appear in the body of in a Horn rule whose head is q, according to Algorithm 1.

Example 4.7 For example the conjunctive query:

$$CQ = Tree(x, y) \geq 0.3$$

presented in Example 3.2, according to the Horn rule component:

$$\mathcal{H} = \{Trunk(x) \wedge isConnected(x, y) \wedge Leafs(y) \Rightarrow Tree(x, y)\}$$

presented in Example 3.1, will be reduced to the conjunctive query:

$$CQ = Trunk(x) \geq 0.3 \wedge isConnected(x, y) \geq 0.3 \wedge Leafs(y) \geq 0.3.$$

and according to Example 4.6 we have that $CQ \hookrightarrow \mathcal{F}_1$.

Lemma 4.8 *The procedure described in Algorithm 1 terminates in a finite number of steps.*

Since the Horn rule component contains only acyclic Horn rules the termination of the algorithm can be easily verified.

Lemma 4.9 *For each cycle of the Algorithm 1 the replacement of UCQ with UCQ' is sound and complete, i.e. $K \models UCQ$ iff $K \models UCQ'$.*

4.3.3 Reasoning for the existential entailment problem

According to [24] an ABox $\mathcal{A}$ corresponds to a β sentence that does not contain any variables. When β contains variables, β may have models in which two or more variables (or one or more variable and an individual) are mapped to the same object in the domain. To check entailment in this case we need to apply the algorithm to any homomorphism h on β.

The homomorphism h is a mapping h : varsIndivs$(\beta) \to \mathcal{P}$ (varsIndivs(β)), where $\mathcal{P}$ corresponds to the powerset function. Obviously each variable or individual is mapped to the set of its homomorphic variables and individuals. From the mapping h and the set varsIndivs(β) we can define a new set of individuals $\mathbf{I}'$. Intuitively, if we consider an homomorphism between the variables x, y of β, i.e. $h(x) = h(y) = \{x, y\}$, then we define a new individual in $\mathbf{I}'$ that is identified by $\{x, y\}$. In order to be a valid homomorphism, the function h must satisfy the following properties:

- $x \in h(x)$,
- if $y \in h(x)$ then it also holds that $h(y) = h(x)$,
- for every pair of different individuals $a, b \in \mathbf{I}$ it applies that $h(a) \neq h(b)$.

Now from β and h we build an ABox $\mathcal{A}'$ as follows: for each conjunct in β about a concept $C(x) \geq n$ (or role $R(x, y) \geq n$) we get a concept (or role) assertion in $\mathcal{A}$ such that $C(h(x)) \geq n$ (or $R(h(x), h(y)) \geq n$). Similarly each individual and universally quantified variable x in one of the conjuncts of Q_i for $1 \leq i \leq m$ is replaced by $h(x)$ and we get a query Q_i' that contains only existentially quantified variables.

Lemma 4.10 $\langle \beta, \mathcal{T} \rangle \models \{Q_1, \ldots, Q_k\}$ *iff for every valid homomorphism h on the set of variables and individuals of* β*, it holds that* $K \models \left\{Q_1', \ldots, Q_k'\right\}$ *where* $K = \langle \mathcal{T}, \mathcal{A} \rangle$, $\mathcal{A} = h(\beta)$, *and* $Q_i' = h(Q_i)$ *for every* $1 \leq i \leq k$.

5 Conclusions and future work

This paper presents a quite general fuzzy extension of the CARIN language. Based on CARIN we have described a language allowing for the integration of fuzzy DLs and Horn rules; thus offering more expressive power due to its ability to represent imprecise and vague information. The extension we have proposed is based on a combination of the DL fuzzy $\mathcal{ALCNR}$ with acyclic Horn rules. The syntax and semantics we suggest are in accordance with those of other fuzzy DL languages such as f_{KD}-$\mathcal{ALC}$ [35], f_{KD}-$\mathcal{SI}$ [31,32] and f_{KD}-$\mathcal{SHIN}$ [33]. Furthermore, we have introduced for fuzzy DLs the key problems of conjunctive queries, union of conjunctive queries and existential entailment, providing proper semantics together with a sound and complete inference procedure for each of these problems. Similar to the classic CARIN language we restrict our expressiveness to acyclic Horn rules. In [24] it is proved that the reasoning problem w.r.t. cyclic Horn rules is undecidable (the proof is based on reduction from the halting problem to a CARIN decision problem). It can be proved that the existence of a sound, complete and terminating reasoning procedure for fuzzy CARIN with cyclic Horn rules would imply the same for the classic CARIN language[3] which is absurd. Therefore we conclude that the reasoning problem w.r.t. cyclic Horn rules is undecidable for fuzzy CARIN.

As far as future directions are concerned, these will include the study of possible extensions of the fuzzy CARIN algorithm using more expressive DLs. Toward this direction in

[3] In order to prove this claim we can exploit the existing work on Reduction from fuzzy to crisp DLs [40].

[28] a sound and complete algorithm for performing unions of conjunctive queries over the DL $\mathcal{SHIQ}$ is presented and we should examine if the same problem could be addressed over f_{KD}-$\mathcal{SHIQ}$ knowledge bases. Another topic of interest is the extension of the CARIN language with fuzzy general concept inclusions and weighted fuzzy rule systems. A great amount of existing work involving the weighted fuzzy rule systems has been carried out (e.g. [7]), while in [37] fuzzy general concept inclusions were introduced. Intuitively a fuzzy GCI of the form $\langle C \sqsubseteq D, n\rangle$ implies that if someone belongs to the set C to some degree $n_1 \in [0, 1]$ then he will also belong to the set D to some other degree $n_2 \in [0, 1]$ where n_2 increases w.r.t. the values of n_1, n. The same applies for weighted fuzzy Horn rules. Another interesting extension is related to the greatest lower bound. The greatest lower bound of some crisp concept assertion w.r.t. a fuzzy knowledge base K is defined as $glb(K, (a : C)) = \sup\{n \mid K \models (a : C) \geq n\}$. For example the fact that $glb(K, (John : Tall)) = 0.8$ indicates that $John$ is $Tall$ for every model of K with a degree greater or equal than 0.8 and there exists at least one model where $John$ is $Tall$ with a degree equal to 0.8. Determining the glb is called the *Best Truth Value Bound* (BTVB) problem [34]. Extending the BTVB problem for conjunctive queries and subsequently for unions of conjunctive queries, is a very interesting problem. A BTVB conjunctive query should have the form:

$$CQ_{BTVB} = p_1\left(\overline{Y_1}\right) \wedge \cdots \wedge p_k\left(\overline{Y_k}\right)$$

where $p_1, \ldots, p_k$ are either concepts, roles, or ordinary predicates. Here we could say that $glb\left(K, CQ_{BTVB}\right) = n$ iff for every model $\mathcal{I}$ of K there exists some mapping σ : varsIndivs $\left(\overline{Y_i}\right) \rightarrow \Delta^{\mathcal{I}}$ such that $p_i^{\mathcal{I}}\left(\sigma\left(\overline{Y_i}\right)\right) \geq n$ for every integer $1 \leq i \leq k$. We believe that our algorithm needs some minor changes in order to answer to this problem. The UCQ_{BTVB} can be defined accordingly.

One of the main drawbacks for further extending the fuzzy CARIN language is its high computational complexity. It has been proved, in [24], for the crisp CARIN system that the time complexity for $\beta \cup \mathcal{T} \not\models Q$ is non-deterministic doubly exponential in the size of $\beta \cup \mathcal{T}$ and triply exponential in the size of $\beta \cup \mathcal{T} \cup Q$. What remains to investigate is if such is the case for its fuzzy extension, or if it leads to a higher worst case complexity. Therefore the use of the fuzzy CARIN language in realistic applications presumes for a more efficient reasoning system. One direction toward a more efficient reasoning system is to extend with Horn rules more tractable DLs then $\mathcal{ALCNR}$. The $\mathcal{ELP}$ language presented in [23] is one such extension combining the polynomial time complexity language $\mathcal{EL}^{++}$ with rules. Another approach in order to provide a more efficient algorithm is to investigate practically efficient methods for reasoning in CARIN. One of the main sources of complexity of our algorithm is that in order to have a sound and complete inference procedure for answering to $UCQs$ each node contains either $\langle \neg C, \geq, 1 - n + \epsilon\rangle$ or $\langle C, \geq, n\rangle$ for all concept descriptions and all degrees in the KB. A more conservative application of this rule, which would bound its application only for the concept names and degrees contained in a conjunctive query, would considerably improve the performance of our algorithm. Another source of complexity of the algorithm is related to the q-blocking condition adopted. A possible solution in this case is to introduce a dynamic version of q-blocking in which an initial completion forest with simple blocking is created and only if none of the conjunctive queries in UCQ has an answer, the completion forest is further expanded. Finally one of the main sources of complexity stems from the fact that fuzzy CARIN is constructed based on non-optimized tableau methods. In [30] optimization techniques that can improve the performance of fuzzy-DL systems' reasoning are presented, while existing optimizations of tableau algorithms for classic DL reasoners [42] can also be adopted.

A Proofs

Proof of Lemma 4.1 For the "if" direction, if $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{V})$ is a fuzzy tableau for $\mathcal{A}$ w.r.t. $\mathcal{T}$, we can construct a model $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ of $\mathcal{A}$ w.r.t $\mathcal{T}$ in the following way:

$$
\begin{array}{lll}
\Delta^{\mathcal{I}} = \mathbf{S} & & \\
a^{\mathcal{I}} = \mathcal{V}(a), a \in \mathbf{I} & & \\
\top^{\mathcal{I}}(s) = \mathcal{L}(s, \top), & \text{for all } s \in \mathbf{S} & \\
\bot^{\mathcal{I}}(s) = \mathcal{L}(s, \bot), & \text{for all } s \in \mathbf{S} & \\
A^{\mathcal{I}}(s) = \mathcal{L}(s, A), & \text{for all } s \in \mathbf{S} & \text{and concept names } A \\
P^{\mathcal{I}}(s, t) = \mathcal{E}(P, \langle s, t \rangle) & \text{for each role } P \in \mathbf{R} &
\end{array}
$$

The proof that $\mathcal{I}$ is a model of $\mathcal{A}$ is based on the proof described in [32] for the language f_{KD}-$\mathcal{SI}$. In [32] it is shown, by induction on the structure of concepts that

$$\mathcal{L}(s, C) \geq n \text{ implies } C^{\mathcal{I}}(s) \geq n \quad \text{for any } s \in \mathbf{S} \tag{4}$$

Since our language does not contain transitive roles, the proof for $\forall$ is a little bit differentiated, not taking into account the part of the proof that refers to transitive roles. The proof for number restrictions is identical to the one described in [32] for the more expressive language $f_{KD} - \mathcal{SHIN}$. Some minor changes to the proofs must be made in order to handle role conjunctions. We give a detailed part of the proof for existential restrictions:

– If $\mathcal{L}(s, \exists R.C) \geq n$ for some role conjunction $R \to P_1 \sqcap \cdots \sqcap P_k$ then there exists some $t \in \mathbf{S}$ such that $\mathcal{E}(P_i, \langle s, t \rangle) \geq n$ for all $i \in \{1, \ldots, k\}$ and $\mathcal{L}(t, C) \geq n$ holds. By definition $P_i^{\mathcal{I}}(s, t) \geq n$ and by the semantics of role conjunction

$$R^{\mathcal{I}}(s, t) = \min\left(P_1^{\mathcal{I}}(s, t), \ldots, P_k^{\mathcal{I}}(s, t)\right) \geq n$$

Since by induction $C^{\mathcal{I}}(t) \geq n$ holds we have that

$$(\exists R.C)^{\mathcal{I}}(s) = \sup_{t \in \Delta^{\mathcal{I}}} \min\left(R^{\mathcal{I}}(s, t), C^{\mathcal{I}}(t)\right) \geq n$$

Property 9 in Definition 4.1 indicates that each concept inclusion $C \sqsubseteq D \in \mathcal{T}$ is satisfied in every model $\mathcal{I}$ of $\mathcal{T}$ (an extensive proof can be found in [34]), properties 10, 11 indicate the satisfaction of each concept and role assertion in $\mathcal{A}$ while property 12 indicates the satisfaction of the unique name assumption.

For the "only if" direction a model $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$ of $\mathcal{A}$ w.r.t. $\mathcal{T}$ can define a tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{V})$ for $\mathcal{A}$ w.r.t. $\mathcal{T}$ as follows:

$$
\begin{aligned}
\mathbf{S} &= \Delta^{\mathcal{I}} \\
\mathcal{E}(P, \langle s, t \rangle) &= P^{\mathcal{I}}(s, t) \\
\mathcal{L}(s, C) &= C^{\mathcal{I}}(s) \\
\mathcal{V}(a) &= a^{\mathcal{I}}
\end{aligned}
$$

The proofs for properties 1, 2, 3, 4, 5, 6, 10, 11 follow the proofs for $f_{KD} - \mathcal{SI}$ in [32] while the proofs for properties 7, 8 are identical to the proofs for $f_{KD} - \mathcal{SHIN}$ in [32] and the proof for property 9 is described in [34]. Property 12 holds due to the unique name assumption. Some minor changes to the proofs must be made in order to handle role conjunctions for example the proof for existential restriction is changed as follows:

– Let $\mathcal{L}(s, \exists R.C) \geq n$ for some role conjunction $R \rightarrow P_1 \sqcap \cdots \sqcap P_k$. The definition of T implies that $(\exists R.C)^{\mathcal{I}}(s) \geq n \Rightarrow \sup_{y \in \Delta^{\mathcal{I}}} \min\left(R^{\mathcal{I}}(s, y), C^{\mathcal{I}}(y)\right) \geq n$. This means that there exists some $t \in \Delta^{\mathcal{I}}$ with $R^{\mathcal{I}}(s, t) \geq n$ and $C^{\mathcal{I}}(t) \geq n$. Since $R^{\mathcal{I}}(s, t) \geq n$ holds, $P_i^{\mathcal{I}}(s, t) \geq n$ also holds for $i \in \{1, \ldots, k\}$. By definition $t \in \mathbf{S}$ and T satisfies Property 6. □

Proof of Lemma 4.2 According to [28], in order to prove the termination of the algorithm it suffices to prove that there exists a maximal number T_q of non-isomorphic q-trees in a completion forest for K. This condition ensures that the expansion of the completion forest won't keep infinitely, since each q-tree equivalence will cause blocking at some point. Suppose that $\mathrm{sub}(K) \cup \mathrm{sub}(Q)$ is the set of subconcepts of a conjunctive query Q on a knowledge base K and $c = |\mathrm{sub}(K) \cup \mathrm{sub}(Q)|$ its cardinality, $r = |\mathbf{R}|$ the number of role names, $m_{\max}$ the maximum m occurring in a number restriction of the form $\geq m\ R$ and $N_{\mathcal{A},Q}$ the number of degrees in $ABox$ assertions as well as conjunctive queries (along with the degrees augmented with ϵ).

There can be at most $2^{c \cdot N_{\mathcal{A},Q}}$ node labels in a completion forest since each node is characterized by tuples of concepts and degrees. Each successor of a node can be the root of a tree of depth $(n-1)$. Considering a single role R, if a node υ has x R-successors, then there is a maximum number of $(T_{n-1})^x$ trees of depth $(n-1)$ rooted at υ.

We consider that in the worst case, the $\geq_{\geq}$ generating rule can be contained in each concept $C \in \mathrm{sub}(K) \cup \mathrm{sub}(Q)$ for each number of degrees and for the highest degree $m_{\max}$. This gives a bound of $c \cdot m_{\max} \cdot N_{\mathcal{A},Q}$ R-successors for each role.

The number of R-successors of a node might range from 0 to $c \cdot m_{\max} \cdot N_{\mathcal{A},Q}$, and for each number of R-successors, we have at most $(T_{n-1})^{(c \cdot m_{\max} \cdot N_{\mathcal{A},Q})}$ trees of depth $(n-1)$. So, each node can be the root of at most $(c \cdot m_{\max} \cdot N_{\mathcal{A},Q})(T_{q-1})^{(c \cdot m_{\max} \cdot N_{\mathcal{A},Q})}$ trees of depth $n-1$ if we consider one single role.

Since at most the same number of trees can be generated for every role in $\mathbf{R}$, there is a bound of $\left(\left(c \cdot m_{\max} \cdot N_{\mathcal{A},Q}\right)\left(T_{q-1}\right)^{(c \cdot m_{\max} \cdot N_{\mathcal{A},Q})}\right)^r$ trees of depth $(n-1)$ rooted at each node. The number of different roots of a n-tree is bounded by 2^c. We now give an upper bound on the number of non-isomorphic n-trees as

$$T_q = \mathcal{O}\left(2^c \left((c \cdot m_{\max} \cdot N_{\mathcal{A},Q})\left(T_{q-1}\right)^{(c \cdot m_{\max} \cdot N_{\mathcal{A},Q})}\right)^r\right)$$

To simplify the notation, let's consider $x = 2^c(c + m_{\max})^r$ and $a = c \cdot m_{\max} r$. Then we have

$$T_q = \mathcal{O}(x \cdot (T_{q-1})^a) = \mathcal{O}(x^{1+a+\cdots+a^{n-1}} \cdot (T_0)^{a^n}) = \mathcal{O}((x \cdot T_0)^{a^n})$$

The maximal number of trees of depth 0 is also bounded by 2^c. Returning to the original notation we get

$$T_q = \mathcal{O}\left((2^{2 \cdot c}(c \cdot m_{\max} \cdot N_{\mathcal{A},Q})^r)^{(c \cdot m_{\max} \cdot N_{\mathcal{A},Q} \cdot r)^n}\right)$$

□

Proof of Lemma 4.3 Let $\mathcal{F}_K$ be a complete and clash-free completion-forest constructed by the tableaux algorithm for $\mathcal{A}$ w.r.t. $\mathcal{T}$. The construction of a fuzzy tableau $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{V})$ will be performed based on the construction of a fuzzy model presented in [32,35].

For a set of triples of the form $\langle A, \geq, n_i \rangle \in \mathcal{L}(s)$, the maximum value of n_i's is chosen as a membership degree of s to the fuzzy set $A^{\mathcal{I}}$, i.e. the degree $\mathcal{L}(s, A)$ in our case. Please note that the labellings $\mathcal{L}(s, C)$ refer to nodes of the fuzzy tableau, while those of $\mathcal{L}(s)$ to nodes of the completion-forest. Given the existence of a clash-free completion forest $\mathcal{F}_K$, a fuzzy tableau can be constructed as follows:

$$
\begin{aligned}
\mathbf{S} &= \{s \mid s \text{ is a node in } \mathcal{F}_{\mathcal{A}} \text{ and } s \text{ is not blocked}\},\\
\mathcal{L}(s, \bot) &= 0, \text{ for all } s \in \mathbf{S},\\
\mathcal{L}(s, \top) &= 1, \text{ for all } s \in \mathbf{S},\\
\mathcal{L}(s, C) &= \sup \{n_i \mid \langle C, \geq, n_i\rangle \in \mathcal{L}(s)\}, \text{ for all } s \in \mathbf{S},\\
\mathcal{E}(P, \langle s, t\rangle) &= \sup\,(\{n_i \mid \langle P, \geq, n_i\rangle \in \mathcal{L}(s, t)\} \cup\\
&\qquad \{n_i \mid \langle P, \geq, n_i\rangle \in \mathcal{L}(s, z) \text{ for each node } z \text{ blocked by } t\})\\
&\quad \text{for all } s, t \in \mathbf{S},\\
\mathcal{V}(a_i) &= s_{a_i}, \text{ where } s_{a_i} \text{ is a root node.}
\end{aligned}
\tag{5}
$$

It can be shown that T is a fuzzy tableau for $\mathcal{A}$ w.r.t. $\mathcal{T}$:

1. Property 1 of Definition 4.1 is satisfied due to the construction of T and because $\mathcal{F}_{\mathcal{A}}$ is class-free.
2. Properties 2,3 and 4 are satisfied and the proof is identical to the proof for $f_{KD} - \mathcal{SI}$.
3. Property 5 of Definition 4.1 is satisfied. Let $s \in \mathbf{S}$ with $\mathcal{L}(s, \forall R.C) = n_0 \geq n$ for some role conjunction $R \to P_1 \sqcap \cdots \sqcap P_k$. We consider two possibilities, the first is that there exists some P_i for some integer $1 \leq i \leq k$, such that $\mathcal{E}(P_i, \langle s, t\rangle) = n_i$ with $n_i \leq 1 - n_0$ and the second is that $\mathcal{E}(P_i, \langle s, t\rangle) = n_i$ with $n_i \geq 1 - n_0 + \epsilon$ for all $1 \leq i \leq k$. For the first case we have that $\mathcal{E}(P_i, \langle s, t\rangle) = n_i \leq 1 - n$, therefore property 5 is satisfied. For the second case we have, by construction of T, that $\langle \forall R.C, \geq, n_1\rangle \in \mathcal{L}(s)$ and since $\mathcal{E}(P_i, \langle s, t\rangle) = n_i \geq 1 - n_0 + \epsilon$ for all $1 \leq i \leq k$ we have either that t is an $R_{\geq 1-n+\epsilon}$-successor of s, or that z is an $R_{\geq 1-n+\epsilon}$-successor of s and t blocks z. If t is an $R_{\geq 1-n+\epsilon}$-successor of s the $\forall_{\geq}$ rule ensures that $\langle C, \geq, n_0\rangle \in \mathcal{L}(t)$, therefore $\mathcal{L}(s, t) \geq n_0 \geq n$. Same applies for the second case since $\langle C, \geq, n_0\rangle \in \mathcal{L}(z)$ and $\mathcal{L}(z) = \mathcal{L}(t)$ because of the blocking condition that indicates an isomorphism ψ between z and t.
4. Property 6 of Definition 4.1 is satisfied. Let $s \in \mathbf{S}$ with $\mathcal{L}(s, \exists R.C) = n_0 \geq n$ for a role conjunction $R \to P_1 \sqcap \cdots \sqcap P_k$. The construction of T implies that $\langle \exists R.C, \geq, n_0\rangle \in \mathcal{L}(s)$. The $\exists_{\geq}$ rule ensures that s has an $R_{\geq n_0}$-successor t (and therefore $\langle P_i, \geq, n_i\rangle \in \mathcal{L}(s, t)$ holds for all integers $1 \leq i \leq k$ with $n_i \geq n_0$) such that $\langle C, \geq, n_0\rangle \in \mathcal{L}(t)$. If t is not blocked Property 6 holds since, by construction of T, $\mathcal{E}(P_i, \langle s, t\rangle) \geq n_0 \geq n$ for all integers $1 \leq i \leq k$ and $\mathcal{L}(t, C) \geq n_0 \geq n$. Same applies for the case that t is blocked by z, since $\mathcal{L}(t) = \mathcal{L}(z)$ holds.
5. Property 7 of Definition 4.1 is satisfied. Suppose that $\mathcal{L}(s, \geq mR) = n_0 \geq n$ for a role conjunction $R \to P_1 \sqcap \cdots \sqcap P_k$. The construction of T indicates that $\langle \geq mR, \geq, n_0\rangle \in \mathcal{L}(s)$. Since $\mathcal{F}_{\mathcal{A}}$ is complete, there will be at least m nodes $t_1, \ldots, t_m$ that are $R_{\geq n_0}$ successors of s such that $t_i \neq t_j$ for all $1 \leq i < j \leq m$ (and therefore $\left\langle P_{i'}, \geq, n_{i,i'}\right\rangle \in \mathcal{L}(s, t_i)$ and $n_{i,i'} \geq n_0$ hold for all $1 \leq i \leq m$, $1 \leq i \leq k$). If t_i's are not blocked, due to the construction of T, we have that $\mathcal{E}(P_{i'}, \langle s, t_i\rangle) = n_{i,i'} \geq n_0 \geq n$ for all $1 \leq i \leq m$, $1 \leq i' \leq k$ and $t_i \neq t_j$ for all $1 \leq i < j \leq m$, so property 7 holds. If t_i's are blocked, due to our blocking condition that indicates isomorphism between completion forests, and the construction of T that indicates $z_i \neq z_j$, if z_i blocks t_i and z_j blocks t_j we have that $\mathcal{E}(P_{i'}, \langle s, z_i\rangle) = n_{i,i'} \geq n_0 \geq n$ for all $1 \leq i \leq m$, $1 \leq i' \leq k$ and $z_i \neq z_j$ for all $1 \leq i < j \leq m$ so property 7 holds.
6. Property 8 of Definition 4.1 is satisfied. Suppose that $\mathcal{L}(s, \leq mR) = n_0 \geq n$. The construction of T indicates that $\langle \leq mR, \geq, n_0\rangle \in \mathcal{L}(s)$. Since $\mathcal{F}_{\mathcal{A}}$ is complete and class free, there will be at most m $R_{\geq 1-n_0+\epsilon}$ successors of s. Following the previous reasoning it is easy to show that property 8 holds.

Table 3 Tableaux expansion rules for fuzzy $\mathcal{ALCNR}$

Rule	Description		
$\sqcup_{\geq}$	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 $C \in \{\langle C_1, \geq, n\rangle, \langle C_2, \geq, n\rangle\}$ not conjugated with $\langle \neg C_1, \geq, 1 - \mathcal{L}(\pi(x), C_1)\rangle$ or $\langle \neg C_2, \geq, 1 - \mathcal{L}(\pi(x), C_2)\rangle$
$\leq_{\geq}$	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 and $\pi(y) = \pi(z)$
	then	1.	$\mathcal{L}(z) \rightarrow \mathcal{L}(z) \cup \mathcal{L}(y)$
		2.	$\mathcal{L}(\langle x, z\rangle) \rightarrow \mathcal{L}(\langle x, z\rangle) \cup \mathcal{L}(\langle x, y\rangle)$
		3.	$\mathcal{L}(\langle x, y\rangle) \rightarrow \emptyset$,
		4.	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 $n \in N^{\mathcal{A}}$
	then		$\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{E\}$ for some $E \in \{\langle \neg C, \geq, 1 - n + \epsilon\rangle, \langle D, \geq, n\rangle\}$ not conjugated with $\langle C, \geq, \mathcal{L}(\pi(x), C)\rangle$ or $\langle \neg D, \geq, 1 - \mathcal{L}(\pi(x), D)\rangle$

7. The proof for property 9 of Definition 4.1 is presented in [34].
8. Property 10, 11 and 12 hold due to the initialization of the completion forest $\mathcal{F}_K$. □

Proof of Lemma 4.4 The proof of completeness is based on [32] with some differences emerging from the none existence of transitive and inverse roles in our knowledge.

Let $T = (\mathbf{S}, \mathcal{L}, \mathcal{E}, \mathcal{V})$ be a fuzzy tableau for $\mathcal{A}$ w.r.t. $\mathcal{T}$. Using T we trigger the application of the expansion rules such that they yield a completion-forest $\mathcal{F}$ that is both complete and clash-free. Following [17] a mapping π, which maps nodes of $\mathcal{F}$ to elements of $\mathbf{S}$ and guides the application of the non-deterministic rules $\sqcup_{\geq}$, $\leq_{\geq}$ and $\sqsubseteq$, is defined such that the following properties hold:

$$\langle C, \geq, n\rangle \in \mathcal{L}(x) \text{ in } \mathcal{F} \Rightarrow \mathcal{L}(\pi(x), C) \geq n \text{ in } T \tag{6}$$

$$x \text{ is a } P_{\geq n} \text{ successor of } y \text{ in } \mathcal{F} \Rightarrow \mathcal{E}(P, \langle \pi(x), \pi(y)\rangle) \geq n \text{ in } T \tag{7}$$

$$x \neq y \text{ in } \mathcal{F} \Rightarrow \pi(x) \neq \pi(y) \text{ in } T \tag{8}$$

According to [32], the proposed method differs from the one used in crisp DLs in the following way. Using the membership degree of a node to a concept, found in the fuzzy tableau, we create artificial triples which are tested against conjugation with the candidate triples that the non-deterministic rules can insert in the completion forest. The triples that don't cause the conjugation can be added. The modified rules that are used to guide such an expansion are presented in Table 3. □

Proof of Lemma 4.5 For the if direction we make the hypothesis that $UCQ \hookrightarrow T$ for every consistent tableau T w.r.t. $\mathcal{A}$ and $\mathcal{T}$ and we want to show that $\mathcal{I} \models UCQ$ for every model $\mathcal{I}$ of $K = \langle \mathcal{T}, \mathcal{A}\rangle$. Suppose that $\mathcal{I}$ is a model of K. Following the construction in the proof of Lemma 4.1 we can build a tableau T for $\mathcal{A}$ w.r.t. $\mathcal{T}$.

From our hypothesis that $UCQ \hookrightarrow T$ for every consistent tableau T w.r.t. $\mathcal{A}$ and $\mathcal{T}$ according to Definition 4.11, there exists a mapping σ : varsIndivs$(CQ) \rightarrow \mathbf{S}$ such that: (i) σ maps each $a \in$ indivs(CQ) to $\mathcal{V}(a)$, (ii) $\mathcal{L}(\sigma(x_i), C_i) \geq n_i$ and (iii) $\mathcal{E}\left(P_j, \langle\sigma(y_j), \sigma(z_j)\rangle\right) \geq n_j$, where C_i, P_j are the concepts and roles in CQ, n_i, n_j their degrees and $x_i, y_i, z_i \in$ varsIndivs(CQ). By construction of T (see proof of Lemma 4.1) we have that (i) σ maps each $a \in$ indivs(CQ) to $a^{\mathcal{I}}$, (ii) $C^{\mathcal{I}}(\sigma(x_i)) = \mathcal{L}(\sigma(x_i), C_i) \geq n_i$ and (iii) $P^{\mathcal{I}}\left(\sigma(y_j), \sigma(z_j)\right) = \mathcal{E}\left(P_j, \langle\sigma(y_j), \sigma(z_j)\rangle\right) \geq n_j$. Therefore, $\mathcal{I} \models UCQ$.

For the only if direction we make the hypothesis that $\mathcal{I} \models UCQ$ for every model $\mathcal{I}$ of $K = \langle\mathcal{T}, \mathcal{A}\rangle$ and we want to show that $UCQ \hookrightarrow T$ for every consistent tableau T w.r.t. $\mathcal{A}$ and $\mathcal{T}$. Suppose that T is a consistent tableau w.r.t. $\mathcal{A}$ and $\mathcal{T}$. Then we can build a model $\mathcal{I}$ of K in a similar way as in the proof of Lemma 4.1.

Since $\mathcal{I} \models UCQ$ we have that there exists a mapping σ : varsIndivs$(CQ) \rightarrow \Delta^{\mathcal{I}}$ such that (i) σ maps each $a \in$ indivs(CQ) to $a^{\mathcal{I}}$, (ii) $C^{\mathcal{I}}(\sigma(x_i)) \geq n_i$ and (iii) $P^{\mathcal{I}}\left(\sigma(y_j), \sigma(z_j)\right) \geq n_j$, where C_i, P_j are the concepts and roles in CQ, n_i, n_j their degrees and $x_i, y_i, z_i \in$ varsIndivs(CQ). By construction of $\mathcal{I}$ from T we also have that (i) σ maps each $a \in$ indivs(CQ) to $\mathcal{V}(a)$, and (ii) $P^{\mathcal{I}}\left(\sigma(y_j), \sigma(z_j)\right) = \mathcal{E}\left(P_j, \langle\sigma(y_j), \sigma(z_j)\rangle\right) \geq n_j$.

In order to finish our proof that σ is a mapping such that $CQ \hookrightarrow T$ it remains to show that $\mathcal{L}(\sigma(x_i), C_i) \geq n_i$. Due to Eq. 3, we have for $\sigma(x_i)$ that either $\mathcal{L}(\sigma(x_i), \neg C) > 1 - n$ or $\mathcal{L}(\sigma(x_i), C) \geq n$ holds. If $\mathcal{L}(\sigma(x_i), \neg C) > 1 - n$ holds then according to Eq. 4 $(\neg C)^{\mathcal{I}}(\sigma(x_i)) > 1 - n$ and according to the semantics of $\neg$ we have $C^{\mathcal{I}}(\sigma(x_i)) < n$. Since $C^{\mathcal{I}}(\sigma(x_i)) < n$ contradicts the fact that $C^{\mathcal{I}}(\sigma(x_i)) \geq n_i$ (by definition of the σ mapping) we must have that $\mathcal{L}(\sigma(x_i), C) \geq n$ which finishes our proof. $\square$

Proof of Lemma 4.6 We assume that $UCQ \hookrightarrow T$ for every consistent tableau T w.r.t. $\mathcal{A}$ and $\mathcal{T}$ and we want to prove that $UCQ \hookrightarrow \mathcal{F}$ for each $\mathcal{F} \in ccf\left(\mathbb{F}_K^q\right)$ where $K = \langle\mathcal{T}, \mathcal{A}\rangle$ and $q = |UCQ|$. Suppose that $\mathcal{F} \in ccf(\mathbb{F}_K^q)$, from $\mathcal{F}$ we built a tableau T according to Eq. 5. According to the proof of Lemma 4.3 this tableau T is consistent w.r.t. $\mathcal{A}$ and $\mathcal{T}$, and from our hypothesis this implies that $UCQ \hookrightarrow T$. We call a pair $\langle s, t\rangle \in \mathbf{S} \times \mathbf{S}$ in T as after-blocked, if it emerges from the second branch of the construction of $\mathcal{E}$ in Eq. 5 (it represents an edge between a non-blocked and a blocked node).

We will first prove that if there exists a mapping in T, containing no after-blocked pairs, such that $UCQ \hookrightarrow T$, then it also holds that $UCQ \hookrightarrow \mathcal{F}$. Since $UCQ \hookrightarrow T$ holds, there exists a conjunctive query $CQ \in UCQ$ such that $CQ \hookrightarrow T$. Since $CQ \hookrightarrow T$, according to Definition 4.11, there exists a mapping σ : varsIndivs$(CQ) \rightarrow \mathbf{S}$ such that: (i) σ maps each $a \in$ indivs(CQ) to $\mathcal{V}(a)$, (ii) $\mathcal{L}(\sigma(x_i), C_i) \geq n_i$ and (iii) $\mathcal{E}\left(P_j, \langle\sigma(y_j), \sigma(z_j)\rangle\right) \geq n_j$, where C_i, P_j are the concepts and roles in CQ, n_i, n_j their degrees and $x_i, y_i, z_i \in$ varsIndivs(CQ). Since $\mathcal{L}(\sigma(x_i), C_i) = n_i' \geq n_i$ holds in T then—by construction of T according to Eq. 5—$\langle C_i, \geq, n_i'\rangle \in \mathcal{L}(\sigma(x_i))$ in $\mathcal{F}$ and $n_i' \geq n_i$. Similarly since $\mathcal{E}\left(P_j, \langle\sigma(y_j), \sigma(z_j)\rangle\right) = n_j' \geq n_j$ holds in T, we conclude—according to Eq. 5—that $\langle P_j, \geq, n_j'\rangle \in \mathcal{L}(\langle\sigma(y_j), \sigma(t_j)\rangle)$ and $n_j' \geq n_j$. Finally, according to Eq. 5 we have that $\mathcal{V}(a_i) = s_{a_i}$ where s_{a_i} is a root node. Therefore the mapping σ satisfies all the conditions described in Definition 4.10 for the completion forest $\mathcal{F}$ and so $CQ \hookrightarrow \mathcal{F}$ and consequently $UCQ \hookrightarrow \mathcal{F}$ both hold.

In order to finish our proof it remains to show that a mapping $\sigma : CQ \rightarrow \mathbf{S}$ containing after-blocked pairs can be reduced to a mapping $\sigma' : CQ \rightarrow \mathbf{S}$ containing no after-blocked pairs. The blocking condition in Definition 4.7 implies that in the initial completion forest $\mathcal{F}$, that was used in order to create T, there exists an isomorphism ψ between the nodes of a q-tree A and a q-tree B where each node in B is a descendant of the root node in A and the set of nodes in A is disjoint with the set of nodes in B. Suppose that ψ is the isomorphism from nodes in B to nodes in A. We inductively define the mapping σ' as follows:

$$\sigma'(x) = \begin{cases} \psi(\sigma(x)) & \text{if a conjunct } P_j(x, y) \geq n_j \text{ in } CQ \text{ is satisfied by} \\ & \text{a pair } \langle\sigma(x), \sigma(y)\rangle \text{ that corresponds to an} \\ & \text{after-blocked edge} \\ \psi(\sigma(x)) & \text{if a conjunct } P_j(x, y) \geq n_j \text{ in } CQ \text{ is satisfied by} \\ & \text{a pair } \langle\sigma(x), \sigma(y)\rangle \text{ and } \sigma'(y) = \psi(\sigma(y)) \\ \sigma(x) & \text{otherwise.} \end{cases} \tag{9}$$

The q-blocking condition ensures that the mapping ψ is defined for every $\sigma(x) \in nodes(B)$ since the depth of the q-tree B is at least equal to the depth of CQ.

By construction, the mapping σ' does not contain any after-blocked pairs. Now it remains to prove that if σ is a mapping implying that $CQ \hookrightarrow T$ the same applies for σ'. According to the definition of q-tree equivalence (Definition 4.5) for an isomorphism ψ, we have that $\mathcal{L}(s) = \mathcal{L}(\psi(s))$ in the completion forest $\mathcal{F}$ and therefore $\mathcal{L}(s, C) = \mathcal{L}(\psi(s), C)$ for each concept description $C \in sub(K)$ in the tableau T and therefore:

$$\mathcal{L}(\sigma(x), C) = \mathcal{L}(\psi(\sigma(x)), C)$$

Now it remains to show that $\mathcal{E}\left(P, \left\langle\sigma'(x), \sigma'(y)\right\rangle\right) = \mathcal{E}\left(P, \langle\sigma(x), \sigma(x)\rangle\right)$:

- If the mapping σ' was altered by the first rule of Eq. 9 this indicates that $\langle\sigma(x), \sigma(y)\rangle$ corresponds to an after-blocked edge, meaning that in the completion forest $\mathcal{F}$ there is a node z that is a successor of $\sigma(x)$ and $\sigma(y)$ blocks z. Since $\sigma(y)$ blocks z according to the conditions of q-blocking and q-tree equivalence (Definitions 4.7, 4.5) there exists an isomorphism ψ such that $\psi(z) = \sigma(y)$ and if $\langle P, \geq, n\rangle \in \mathcal{L}\left(\langle\sigma(x), z\rangle\right)$ then $\langle P, \geq, n\rangle \in \mathcal{L}\left(\langle\psi(\sigma(x)), \psi(z)\rangle\right)$ holds for each role name in $\mathbf{R}$. Since $\mathcal{L}\left(\langle\sigma(x), z\rangle\right) = \mathcal{L}\left(\langle\psi(\sigma(x)), \psi(z)\rangle\right)$ and $\psi(z) = \sigma(y)$ we have by construction of $\mathcal{E}$ that $\mathcal{E}\left(P_j, \left\langle\sigma'(x), \sigma'(y)\right\rangle\right) = \mathcal{E}\left(P_j, \langle\psi(\sigma(x)), \sigma(y)\rangle\right) = \mathcal{E}\left(P_j, \langle\sigma(x), \sigma(y)\rangle\right)$.
- If the mapping was altered by the second rule of Eq. 9 we have due to the blocking condition that $\mathcal{L}\left(\langle\sigma(x), \sigma(y)\rangle\right) = \mathcal{L}\left(\langle\psi(\sigma(x)), \psi(\sigma(y))\rangle\right)$ and therefore $\mathcal{E}\left(P_j, \left\langle\sigma'(x), \sigma'(y)\right\rangle\right) = \mathcal{E}\left(P_j, \langle\sigma(x), \sigma(y)\rangle\right)$.
- Finally for edges $\left\langle\sigma'(x), \sigma'(y)\right\rangle$ such that $\sigma'(x) = \sigma(x)$ and $\sigma'(y) = \sigma(y)$ it obviously holds that $\mathcal{E}\left(P_j, \left\langle\sigma'(x), \sigma'(y)\right\rangle\right) = \mathcal{E}\left(P_j, \langle\sigma(x), \sigma(y)\rangle\right)$.

Therefore σ' is a mapping with no after-blocked edges that satisfies $CQ \hookrightarrow T$. □

Proof of Lemma 4.7 Suppose that $UCQ \hookrightarrow \mathcal{F}$ for every $\mathcal{F} \in ccf\left(\mathbb{F}_K^q\right)$ where $K = \langle\mathcal{T}, \mathcal{A}\rangle$ and $q = |UCQ|$ and we want to prove that $UCQ \hookrightarrow T$ for every consistent tableau T w.r.t. $\mathcal{A}$ and $\mathcal{T}$. We will prove it by contradiction. We make the assumption that there exists a tableau T such that $CQ \hookrightarrow T$ does not hold.

We construct from T a completion forest $\mathcal{F}$ as in Lemma 4.4 by a mapping π which maps nodes of $\mathcal{F}$ to elements of $\mathbf{S}$ and steers the application of the non-deterministic rules such that the knowledge in T won't be conjugated with the corresponding knowledge in $\mathcal{F}$. According to Lemma 4.4 $\mathcal{F} \in ccf\left(\mathbb{F}_K^q\right)$. Since $UCQ \hookrightarrow \mathcal{F}$ for every $\mathcal{F} \in ccf\left(\mathbb{F}_K^q\right)$, there exists a mapping σ : varsIndivs(CQ) $\rightarrow$ *nodes* ($\mathcal{F}$) such that: (i) σ maps each individual in indivs(CQ) to its corresponding root node, (ii) $\langle C_i, \geq, n_i\rangle \in \mathcal{L}\left(\sigma(x_i)\right)$ for each conjunct $C_i(x_i) \geq n_i$ in CQ, and (iii) $\sigma(y_i)$ is an $P_{j \geq n_j}$ successor of $\sigma(z_j)$ for each conjunct $P_j\left(y_j, z_j\right) \geq n_j$ in CQ (where CQ is some mapping in UCQ such that $CQ \hookrightarrow \mathcal{F}$).

Based on the mappings π, σ we build a new mapping σ' : varsIndivs(CQ) $\rightarrow \mathbf{S}$ as follows: $\sigma'(x) = \pi(\sigma(x))$. From the properties of the mapping π presented in Eqs. 6, 7, 8 we have that if $\langle C, \geq, n\rangle \in \mathcal{L}(\sigma(x))$ in $\mathcal{F}$ then $\mathcal{L}(\pi(\sigma(x)), C) \geq n$ in T. We also have that if $\sigma(x)$ is a $P_{\geq n}$ successor of $\sigma(y)$ in the completion forest $\mathcal{F}$ then $\mathcal{E}(P, \langle\pi(\sigma(x)), \pi(\sigma(y))\rangle) \geq n$ holds in T. Finally since the mapping π maps root nodes to elements of $\mathcal{V}$ we can conclude that σ' implies that $CQ \hookrightarrow T$ which contradicts our assumption that $CQ \hookrightarrow T$ does not hold. □

Proof of Lemma 4.9 Only if direction: For the only if direction it suffices to prove that $K \models UCQ' \Rightarrow K \models UCQ$. Suppose that $\mathcal{I}$ is a model of K if $\mathcal{I} \models CQ$ for some $CQ \in UCQ' \cap UCQ$, then obviously $\mathcal{I} \models UCQ$. If $\mathcal{I} \models CQ$ for some $CQ \in UCQ' \setminus UCQ$ then according to Algorithm 1 it satisfies a conjunctive query of the form:

$$CQ' = p_1\left(\overline{Y}_1\right) \geq n_1 \wedge \cdots \wedge r_1\left(\psi\left(\overline{X}_1\right)\right) \geq n_q \wedge \cdots$$
$$\wedge r_k\left(\psi\left(\overline{X}_k\right)\right) \geq n_q \wedge \cdots \wedge p_k\left(\overline{Y}_k\right) \geq n_k$$

where the mapping ψ is defined according to Algorithm 1 due to the existence of a Horn rule $r_1(\overline{X}_1) \wedge \cdots \wedge r_k(\overline{X}_k) \Rightarrow q(\overline{Y})$. Since $\mathcal{I}$ satisfies CQ', then there exists a mapping $\sigma : \text{varsIndivs}(CQ') \rightarrow \Delta^{\mathcal{I}}$ such that $r_i^{\mathcal{I}}\left(\sigma\left(\psi\left(\overline{X}_i\right)\right)\right) \geq n_q$ for every integer $1 \leq i \leq k$. According to the Horn rules semantics we have that:

$$q^{\mathcal{I}}\left(\sigma\left(\psi\left(\overline{Y}\right)\right)\right) \geq \min\left(r_1^{\mathcal{I}}\left(\sigma\left(\psi\left(\overline{X}_1\right)\right)\right), \ldots, r_k^{\mathcal{I}}\left(\sigma\left(\psi\left(\overline{X}_k\right)\right)\right)\right) \geq n_q$$

and since $\psi\left(\overline{Y}\right) = \overline{Y}_q$ we also have that $q^{\mathcal{I}}\left(\sigma\left(\overline{Y}_q\right)\right) \geq n_q$. So there also exists a solution for UCQ.

If direction: We want to prove that $K \models UCQ \Rightarrow K \models UCQ'$. It suffices to prove that if there exists a model $\mathcal{I}'$ of K that does not satisfy UCQ' then there also exists a model $\mathcal{I}$ of K that does not satisfy UCQ. If the interpretation $\mathcal{I}'$ does not satisfy UCQ then $\mathcal{I}'$ is the mapping we were searching for. If $\mathcal{I}'$ satisfies UCQ this means that the query $CQ \in UCQ \setminus UCQ'$ is satisfied by $\mathcal{I}'$ ($UCQ \setminus UCQ'$ contains exactly one element since in each cycle of an execution of the reduction algorithm only one conjunctive query is removed). This means that there exists a mapping $\sigma : \text{varsIndivs}(CQ') \rightarrow \Delta^{\mathcal{I}'}$ such that:

$$p_1^{\mathcal{I}'}\left(\sigma\left(\overline{Y}_1\right)\right) \geq n_1 \wedge \cdots \wedge q^{\mathcal{I}'}\left(\sigma\left(\overline{Y}_q\right)\right) \geq n_q \wedge \cdots \wedge p_k^{\mathcal{I}'}\left(\sigma\left(\overline{Y}_k\right)\right) \geq n_k$$

From the interpretation $\mathcal{I}'$ we build another interpretation $\mathcal{I}$ such that $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$ and $\cdot^{\mathcal{I}}$ is identical to $\cdot^{\mathcal{I}'}$ with the only exception that $q^{\mathcal{I}}\left(\sigma\left(\overline{Y}_q\right)\right) = n'_q$ such that n'_q satisfies the following two properties:

– for every Horn rule of the form $p_1(\overline{X}_1) \wedge \cdots \wedge p_k(\overline{X}_k) \Rightarrow q(\overline{Y})$ that has q in its head it holds that $n'_q \geq \min\left(p_1^{\mathcal{I}}(\psi(\overline{X_1})), \ldots, p_k^{\mathcal{I}}(\psi(\overline{X_k}))\right)$ for every mapping $\psi : \text{varsIndivs}\left(\overline{X}_1, \ldots, \overline{X}_k, \overline{Y}\right) \rightarrow \Delta^{\mathcal{I}}$ such that $\psi\left(\overline{Y}\right) = \sigma\left(\overline{Y}_q\right)$.
– for some Horn rule of the form $p_1(\overline{X}_1) \wedge \cdots \wedge p_k(\overline{X}_k) \Rightarrow q(\overline{Y})$ that has q in its head it holds that $n'_q = \min\left(p_1^{\mathcal{I}}(\psi(\overline{X_1})), \ldots, p_k^{\mathcal{I}}(\psi(\overline{X_k}))\right)$ for some mapping $\psi : \text{varsIndivs}\left(\overline{X}_1, \ldots, \overline{X}_k, \overline{Y}\right) \rightarrow \Delta^{\mathcal{I}}$ such that $\psi\left(\overline{Y}\right) = \sigma\left(\overline{Y}_q\right)$.

In order to prove that $\mathcal{I}$ is also a model of K it suffices to show that it satisfies each Horn rule in $\mathcal{H}$ that has q in its head. This is obvious by construction of n'_q and according to the semantics of Horn rules presented in Sect. 3.2. So in the constructed model $\mathcal{I}$ we have that $q^{\mathcal{I}}\left(\sigma\left(\overline{Y}_q\right)\right) = n'_q < n_q$ (otherwise there would be a model of $\mathcal{I}'$ that also satisfied UCQ'). If there is some other mapping in $\mathcal{I}$ σ' such that $q^{\mathcal{I}}\left(\sigma'\left(\overline{Y}_q\right)\right) \geq n_q$ we can similarly create another interpretation such that $q^{\mathcal{I}}\left(\sigma'\left(\overline{Y}_q\right)\right) < n_q$. Therefore we can create a model $\mathcal{I}$ of K such that it does not satisfy CQ and therefore UCQ as we wanted to show. □

Proof of Lemma 4.10 We want to show that $\langle \beta, \mathcal{T} \rangle \models \{Q_1, \ldots, Q_m\} \Leftrightarrow K \models \{Q'_1, \ldots, Q'_m\}$ for $K = \langle \mathcal{A}', \mathcal{T} \rangle$. It suffices to prove that there exists some $\mathcal{I}' \models K$ such that $\mathcal{I}' \not\models \{Q'_1, \ldots, Q'_m\}$ iff there exists some $\mathcal{I} \models \langle \beta, \mathcal{T} \rangle$ such that $\mathcal{I} \not\models \{Q_1, \ldots, Q_m\}$.

For the if direction: Suppose that $\mathcal{I}' \models K$ and $\mathcal{I}' \not\models \{Q'_1, \ldots, Q'_m\}$. From the mapping $\mathcal{I}'$ we create a mapping $\mathcal{I}$ such that $\Delta^{\mathcal{I}} = \Delta^{\mathcal{I}'}$, $x^{\mathcal{I}} = h(x)^{\mathcal{I}'}$ for each $x \in \text{varsIndivs}(\beta)$, $A^{\mathcal{I}}(\upsilon) = A^{\mathcal{I}'}(\upsilon)$ and $R^{\mathcal{I}}(\upsilon, \omega) = R^{\mathcal{I}'}(\upsilon, \omega)$ for each $\upsilon, \omega \in \Delta^{\mathcal{I}'}$. It is obvious that $\mathcal{I} \models \langle \beta, \mathcal{T} \rangle$. In order to finish our proof it remains to show that $\mathcal{I} \not\models \{Q_1, \ldots, Q_m\}$.

We make the assumption that $\mathcal{I} \models Q_i$ for some integer $1 \leq i \leq m$. Then there exists a mapping $\tau : \text{varsIndivs}(Q_i) \to \Delta^{\mathcal{I}}$ that satisfies the conditions described in Eq. 2. We define a mapping $\sigma : \text{varsIndivs}(Q'_i) \to \Delta^{\mathcal{I}'}$ such that:

$$\sigma(x') = \begin{cases} \tau(x') & \text{if} \quad x' \in \text{vars}(Q'_i) \\ x'^{\mathcal{I}'} & \text{if} \quad x' \in \text{indivs}(Q'_i) \end{cases} \tag{10}$$

In order to prove that σ is a solution for Q'_i it suffices to show that $\tau(x) = \sigma(x')$ for every element $x \in \text{varsIndivs}(Q_i)$, $x' \in \text{varsIndivs}(Q'_i)$ such that if x is located in the jth position in Q_i, x' is located in the jth position in Q'_i. If x is an existentially quantified variable in Q_i then it remains unchanged in Q'_i and therefore $\tau(x) = \sigma(x')$ according to Eq. 10. If x is a universally quantified variable, or some individual in Q_i, then x' corresponds to $h(x)$ in Q'_i. From Eq. 2 we have that $\tau(x) = x^{\mathcal{I}}$ and from Eq. 10 that $\sigma(x') = h(x)^{\mathcal{I}'}$. By construction of $\mathcal{I}$ we have that $x^{\mathcal{I}} = h(x)^{\mathcal{I}'}$ as we wanted to show. Therefore we have that $\mathcal{I}' \models \{Q'_1, \ldots, Q'_m\}$ which is impossible and therefore the assumption that $\mathcal{I} \models Q_i$ is wrong and therefore $\mathcal{I} \not\models \{Q_1, \ldots, Q_m\}$.

For the only if direction: Suppose that $\mathcal{I} \models \langle \beta, \mathcal{T} \rangle$ and $\mathcal{I} \not\models \{Q_1, \ldots, Q_m\}$. For every $x \in \text{varsIndivs}(\beta)$ we create an homomorphism h as follows:

$$h(x) = \left\{ y \mid x^{\mathcal{I}} = y^{\mathcal{I}} \text{ for every } y \in \text{varsIndivs}(\beta) \right\}$$

From the homomorphism h we build a new ABox $\mathcal{A}'$ and union of conjunctive queries $\{Q'_1, \ldots, Q'_m\}$.

From $\mathcal{I}$ we create a mapping $\mathcal{I}'$ such that $\Delta^{\mathcal{I}'} = \Delta^{\mathcal{I}}, h(x)^{\mathcal{I}'} = x^{\mathcal{I}}$ for every $x \in \text{varsIndivs}(\beta)$, $A^{\mathcal{I}'}(\upsilon) = A^{\mathcal{I}}(\upsilon)$ and $R^{\mathcal{I}'}(\upsilon, \omega) = R^{\mathcal{I}}(\upsilon, \omega)$ for every $\upsilon, \omega \in \Delta^{\mathcal{I}}$. Obviously $\mathcal{I}' \models \mathcal{A}'$. If $\mathcal{I}' \models Q'_i$ for some Q'_i then there exists a mapping $\sigma : \text{varsIndivs}(Q'_i) \to \Delta^{\mathcal{I}'}$ that satisfies the conditions described in Eq. 1. From this mapping, we build a mapping $\tau : \text{varsIndivs}(Q_i) \to \Delta^{\mathcal{I}}$ such that:

$$\tau(x) = \begin{cases} \sigma(h(x)) & \text{if } x \in \text{varsIndivs}(\beta) \\ \sigma(x) & \text{otherwise} \end{cases}$$

It can be easily checked that the mapping τ satisfies the conditions described in Eq. 2 and therefore it implies that $\mathcal{I} \models \{Q_1, \ldots, Q_m\}$. Since $\mathcal{I} \models \{Q_1, \ldots, Q_m\}$ contradicts the fact that $\mathcal{I} \not\models \{Q_1, \ldots, Q_m\}$, the hypothesis we have made that $\mathcal{I}' \models Q'_i$ is wrong. □

References

1. Antoniou G, Viegas DC, Grosof B, Horrocks I, Kifer M, Maluszynski J, Patel-Schneider PF (2006) Combining rules and ontologies, a survey
2. Baader F, Brandt S, Lutz C (2005) Pushing the el envelope. In: Kaelbling LP, Saffiotti A (eds) IJCAI-05, Proceedings of the nineteenth international joint conference on artificial intelligence. Professional Book Center, Edinburgh, pp 364–369

3. Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF (eds) (2003) The description logic handbook: theory, implementation, and applications. Cambridge University Press, Cambridge
4. Baader F, Hollunder B (1991) A terminological knowledge representation system with complete inference algorithms. In: Boley H, Richter MM (eds) Processing declarative knowledge, international workshop PDK 91. Springer, pp 67–86
5. Cao B, Badia A (2009) Exploiting maximal redundancy to optimize sql queries. Knowl Inf Syst 20:187–220
6. Carrasco RA, Vila MA, Galindo J (2003) Fsql: a flexible query language for data mining. Enterp Inf Syst IV:68–74
7. Chortaras A, Stamou G, Stafylopatis A (2007) Integrated query answering with weighted fuzzy rules. In: Mellouli K (ed) Symbolic and quantitative approaches to reasoning with uncertainty, 9th European Conference, ECSQARU. Springer, Tunisia, pp 767–778
8. Donini FM, Lenzerini M, Nardi D, Schaerf A (1998) Al-log: integrating datalog and description logics. J Intell Inf Syst 10(3):227–252
9. Fagin R (1998) Fuzzy queries in multimedia database systems. In: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on principles of database systems. ACM, Seattle, United States, pp 1–10
10. Franconi E, Tessaris S (2004) Rules and queries with ontologies: unified logical framework. In: Hans JO, Schaffert S (eds) Principles and practice of semantic web reasoning, second international workshop, PPSWR. Springer, France, pp 50–60
11. Goble CA, Haul C, Bechhofer S (1996) Describing and classifying multimedia using the description logic grail. In: Sethi IK, Jain R (eds) Storage and retrieval for image and video databases (SPIE). San Diego, USA pp 132–143
12. Golbreich C, Bierlaire O, Dameron O, Gibaud B (2005) Use case: ontology with rules for identifying brain anatomical structures. In: W3C workshop on rule languages for interoperability. W3C, Washington
13. Gong Z, Liu Q (2009) Improving keyword based web image search with visual feature distribution and term expansion. Knowl Inf Syst 21:113–132
14. Grosof BN, Horrocks I, Volz R, Decker S (2003) Description logic programs: combining logic programs with description logic. In: Proceedings of the 12th international conference on world wide web. ACM New York, Budapest, Hungary, pp 48–57
15. Herman I (2007) Semantic web technical report. http://www.w3.org/2001/sw/
16. Herman I, Hendler J (2004) Web ontology language (OWL). technical report. http://www.w3.org/2004/OWL/
17. Horrocks I, Kutz O, Sattler U (2006) The even more irresistible sroiq. In: Doherty P, Mylopoulos J, Welty CA (eds) Proceedings, tenth international conference on principles of knowledge representation and reasoning. AAAI Press, Lake District of the United Kingdom, pp 57–67
18. Horrocks I, Patel-Schneider PF (2004) A proposal for an owl rules language. In: Feldman SI, Uretsky M, Najork M, Wills CE (eds) Proceedings of the 13th international conference on world wide web, WWW. ACM, New York, pp 723–731
19. Horrocks I, Patel-Schneider PF, Van Harmelen F (2003) From shiq and rdf to owl: the making of a web ontology language. J Web Semant 1:7–26
20. Jing Y, Jeong D, Baik D-K (2008) Sparql graph pattern rewriting for owl-dl inference queries. Knowl Inf Syst 20:243–262
21. Kifer M, Lausen G, Wu J (1995) Logical foundations of object-oriented and frame-based languages. J ACM 42(4):741–843
22. Klir GJ, Yuan B (1995) Fuzzy sets and fuzzy logic: theory and applications. Prentice-Hall, Upper Saddle River
23. Krötzsch M, Rudolph S, Hitzler P (2008) Elp: tractable rules for owl 2. In: Sheth AP, Staab S, Dean M, Paolucci M, Maynard D, Finin TW, Thirunarayan K (eds) Proceedings of 7th international semanticWeb conference, ISWC. Springer, Germany, pp 649–664
24. Levy AY, Rousset M-C (1998) Combining horn rules and description logics in carin. Artif Intell 104(1–2): 165–209
25. Lukasiewicz T (2006) Fuzzy description logic programs under the answer set semantics for the semantic web. In: Eiter T, Franconi E, Hodgson R, Stephens S (eds) Rules and rule markup languages for the semantic web, second international conference, RuleML. IEEE Computer Society, Athens, pp 89–96
26. Meghini C, Sebastiani F, Straccia U (2001) A model of multimedia information retrieval. J ACM 48(5):909–970
27. O'Dea D, Geoghegan S, Ekins C (2005) Dealing with geospatial information in the semantic web. In: Meyer T, Orgun MA (eds) AOW '05: proceedings of the 2005 Australasian ontology workshop. pp 69–73

28. Ortiz M, Calvanese D, Eiter T (2006) Characterizing data complexity for conjunctive query answering in expressive description logics. In: Proceedings, the twenty-first national conference on artificial intelligence and the eighteenth innovative applications of artificial intelligence conference. Boston, Massachusetts, USA, pp 275–280
29. Pan JZ, Stamou G, Stoilos G, Thomas E (2007) Expressive querying over fuzzy dl-lite ontologies. In: Calvanese D, Franconi E, Haarslev V, Lembo D, Motik B, Turhan A-Y, Tessaris S (eds) Proceedings of the 2007 international workshop on description logics (DL2007). Bozen-Bolzano, Italy, CEUR-WS.org
30. Simou N, Mailis T, Stoilos G, Stamou G (2010) Optimization techniques for fuzzy description logics. In: Proceedings of the international workshop on description logics (DL-2010), pp 244–254
31. Stoilos G, Stamou G, Tzouvaras V, Pan JZ, Horrocks I (2005) A fuzzy description logic for multimedia knowledge representation. In: Proceedings of the international workshop on multimedia and the semantic web
32. Stoilos G, Stamou GB, Pan JZ, Tzouvaras V, Horrocks Ian (2007) Reasoning with very expressive fuzzy description logics. J Artif Intell Res (JAIR) 30:273–320
33. Stoilos G, Stamou GB, Tzouvaras V, Pan JZ, Horrocks I (2005) The fuzzy description logic f-shin. In: da Costa PCG, Laskey KB, Laskey KJ, Pool M (eds) International semantic web conference, ISWC 2005, workshop 3: uncertainty reasoning for the semantic web. Galway, Ireland, pp 67–76
34. Stoilos G, Straccia U, Stamou GB, Pan JZ (2006) General concept inclusions in fuzzy description logics. In: Brewka G, Coradeschi S, Perini A, Traverso P (eds) ECAI 2006, 17th european conference on artificial intelligence. IOS Press, Riva del Garda, Italy, pp 457–461
35. Straccia U (2001) Reasoning within fuzzy description logics. J Artif Intell Res (JAIR) 14:137–166
36. Straccia U (2005) Description logics with fuzzy concrete domains. In: Fahiem B, Tommi J (eds) 21st conference on uncertainty in artificial intelligence (UAI-05). AUAI Press, Edinburgh, pp 559–567
37. Straccia U (2006) A fuzzy description logic for the semantic web. In: Sanchez E (ed) Fuzzy logic and the semantic web capturing intelligence, chap. 4. Elsevier, Amsterdam pp 73–90
38. Straccia U (2006) Fuzzy description logic programs. In: Proceedings of the 11th international conference on information processing and management of uncertainty in knowledge-based systems, (IPMU-06), pp 1818–1825
39. Straccia U (2006) Uncertainty and description logic programs over lattices. In: Sanchez E (ed) Fuzzy logic and the semantic web, capturing intelligence, chap. 7. Elsevier, Amsterdam pp 115–133
40. Straccia U, Isti cnr, Moruzzi GV(2004) Transforming fuzzy description logics into classical description logics. In: In Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA-04), number 3229 in Lecture Notes in Computer Science. Springer Verlag, pp. 385–399
41. Taha K, Elmasri R (2008) Bussengine: a business search engine. Knowl Inf Syst 23:153–197
42. Tsarkov D, Horrocks I, Patel-Schneider PF (2007) Optimizing terminological reasoning for expressive description logics. J Autom Reason 39(3):277–316
43. Venetis T, Stoilos G, Stamou GB, Kollias SD (2007) f-dlps: extending description logic programs with fuzzy sets and fuzzy logic. In: FUZZ-IEEE 2007, IEEE international conference on fuzzy systems. IEEE, London, pp 1–6

Author Biographies

Theofilos Mailis was born in Athens in January 1981. He obtained a Diploma Degree in Electrical and Computer Engineering in 2004 from National Technical University of Athens. He is currently a Ph.D. student at the Image, Video and Multimedia Laboratory (IVML), of the School of Electrical and Computer Engineering, National and Technical University of Athens. His research interests include knowledge representation and reasoning (description logics, logic programming, fuzzy logic, ontology and rule languages) and theory of uncertainty (fuzzy sets and systems).

Giorgos Stoilos was born in Athens in July 1981. In 2003 he received his Certificate in Computer Science from the Department of Computer Science and Telecommunication, National and Kapodistrian University of Athens. Then, at the same year he joined the Image, Video and Multimedia Laboratory (IVML), of the School of Electrical and Computer Engineering, National and Technical University of Athens to study toward obtaining a Ph.D. degree, which he successfully defended in February 2008. He worked for 1 year in the IVML which he left in May 2009 and since then he is occupied in the Oxford university computing laboratory. His research interests include all aspects of knowledge representation and reasoning, especially focused on Description Logics and ontology languages, such as OWL, as well as their application in areas such as the Semantic Web and Multimedia.

Giorgos Stamou was born in Athens, Greece in 1971. He obtained a Diploma Degree in Electrical and Computer Engineering in 1994 and a Ph.D. in Computational Intelligence in 1998, both from National Technical University of Athens. Now, he is a lecturer in School of Electrical and Computer Engineering, Division of Computer Science and a member of the Image, Video and Multimedia Laboratory, heading the Knowledge Technologies Group. Before, he was a senior researcher in the Institute of Computer and Communication Systems. His main research interest include knowledge representation and reasoning (description logics, logic programming, fuzzy logic, ontology and rule languages), theory of uncertainty (fuzzy sets and systems, Markov models), machine learning (neural networks, support vector machines), semantic interoperability (ontology alignment, Semantic Web applications) and digital archives (multimedia content annotation, multimedia databases). He has published numerous papers in several areas of computer science and contributed in several International Conferences and Workshops.