Deciding Query Entailment for Fuzzy $\mathcal{SHIN}$ Ontologies

Jingwei Cheng, Z.M. Ma, Fu Zhang, and Xing Wang

Northeastern University, Shenyang, 110004, China
cjingwei@gmail.com, mazongmin@ise.neu.edu.cn

Abstract. Significant research efforts in the Semantic Web community are recently directed toward the representation and reasoning with fuzzy ontologies. As the theoretical counterpart of fuzzy ontology languages, fuzzy Description Logics (DLs) have attracted a wide range of concerns. With the emergence of a great number of large-scale domain ontologies, the basic reasoning services cannot meet the need of dealing with complex queries (mainly conjunctive queries), which are indispensable in data-intensive applications. Conjunctive queries (CQs), originated from relational databases, play an important role as an expressive reasoning service for ontologies. Since, however, the negation of a role atom in a CQ is not expressible as a part of a knowledge base, existing tableau algorithms cannot be used directly to deal with the issue. In this paper, we thus present a tableau-based algorithm for deciding query entailment of fuzzy conjunctive queries w.r.t. fuzzy $\mathcal{SHIN}$ ontologies. Moreover, the data complexity problem was still open for answering CQs in expressive fuzzy DLs. We tackle this issue by proving a tight coNP upper bound for the problem in f-$\mathcal{SHIN}$, as long as only simple roles occur in the query. Regarding combined complexity, we prove that the algorithm for query entailment is co3NExpTime in the size of the knowledge base and the query.

1 Introduction

In order to achieve reusability and a high level of interoperability of knowledge, ontologies are commonly used to express domain knowledge in the context of the Semantic Web. A key component of the Semantic Web is thus the representation and reasoning of ontologies. Description logics (DLs, for short) [1] are the logical foundation of the Semantic Web, which support knowledge representation and reasoning by means of the concepts and roles. As the logic underpinnings of Web Ontology Languages (OWLs)[1], DLs have attracted much more attentions due to their inherently built reasoning services.

In the real world, there exists a great deal of uncertainty and imprecision which is likely the rule than an exception. Based on Zadeh's fuzzy set theory[2], there have been substantial amounts of work carried out in the context of fuzzy DLs [3][4], and fuzzy ontology knowledge bases [5] are thus established.

[1] http://www.w3.org/submission/owl11-overview/

A. Gómez-Pérez, Y. Yu, and Y. Ding (Eds.): ASWC 2009, LNCS 5926, pp. 120–134, 2009.

Conjunctive queries originated from research in relational databases, and, more recently, have also been identified as a desirable form of querying DL knowledge bases. Conjunctive queries provide an expressive query language with capabilities that go beyond standard instance retrieval. There are close ties among conjunctive query answering, conjunctive query entailment and conjunctive query containment in the sense that they can be transformed into one another. The first conjunctive query algorithm [6] over DLs was actually specified for the purpose of deciding conjunctive query containment for $\mathcal{DLR}_{reg}$. Recently, query entailment and answering have been extensively studied for tractable DLs, i.e., DLs that have reasoning problems of at most polynomial complexity. For example, the constructors provided by DL-Lite family [7] are elaborately chosen such that the standard reasoning tasks are PTIME-COMPLETE and query entailment is in LOGSPACE with respect to data complexity. Moreover, in DL-Lite family, as TBox reasoning can usually be done independently of the ABox, ABox storage can be transformed into database storage, thus knowledge base users can achieve efficient queries by means of well-established DBMS query engines. Another tractable DL comes from $\mathcal{EL}$ with PTIME-COMPLETE reasoning complexity. It was shown that union of conjunctive queries (UCQs) entailment in $\mathcal{EL}$ and in its extensions with role hierarchies is NP-complete regarding the combined complexity [8]. The data complexity of UCQ entailment in $\mathcal{EL}$ is PTIME-COMPLETE [9]. Allowing, additionally, role composition in the logic as in $\mathcal{EL}^{++}$, leads to undecidability [10]. Query answering algorithms for expressive DLs are being tracked with equal intensity. CARIN system [11], the first framework for combining a description logic knowledge base with rules, provided a decision procedure for conjunctive query entailment in the description logic $\mathcal{ALCNR}$, where $\mathcal{R}$ stands for role conjunction. The conjunctive query entailment algorithms for more expressive DLs, e.g., $\mathcal{SHIQ}$ and $\mathcal{SHOQ}$, are presented in [12][13].

When querying over fuzzy DL KBs, as in the crisp case, same difficulties emerged in that existing fuzzy DL reasoners, such as fuzzyDL[2] and FiRE[3], are not capable of dealing with CQs either. Some work has been done in a relative narrow range, mainly focused on lightweight fuzzy ontology languages, e.g. [14] and [15] on f-DL-Lite, [16] on f-$\mathcal{ALC}$ and [17] on f-$\mathcal{ALCN}$. In [18], A fuzzy extension of CARIN is provided, along with an fuzzy version of existential entailment algorithm for answering conjunctive queries.

In this paper, we extend the results obtained in [18] for fuzzy $\mathcal{ALCNR}$ and in [19] for crisp $\mathcal{SHIN}$ to fuzzy $\mathcal{SHIN}$. This paper makes the following major contributions:

- It presents a tableau-based algorithm for deciding query entailment over f-$\mathcal{SHIN}$ KBs.
- It provides complexity upper bounds w.r.t both combined complexity and data complexity.

The remainder of this paper is organized as follows. Section 2 briefly reviews the necessary background knowledge of fuzzy sets, fuzzy logics, the syntax and

[2] http://gaia.isti.cnr.it/$\sim$ straccia/software/fuzzyDL/fuzzyDL.html
[3] http://www.image.ece.ntua.gr/$\sim$nsimou

semantics of f-$\mathcal{SHIN}$, and the formal definition of a fuzzy query language. Section 3 presents the conjunctive query entailment algorithm for f-$\mathcal{SHIN}$ along with complexity analysis. Section 4 concludes this paper.

For the lack of the space, we omit most of the proofs, which can be found in a accompanying technical report [20].

2 Preliminaries

2.1 Fuzzy Set and Fuzzy Logic

In classical set theory, an element in a set either belongs or0 does not belong to the set. By contrast, in fuzzy set theory, an element belongs to a set with certain degree, which is described with the aid of a membership function valued in the real unit interval [0,1].

A fuzzy set A with regard to a universe U is characterized by a membership function $\mu_A : U \rightarrow [0, 1]$ (or simply $A(x) \in [0, 1]$), which assigns a membership degree to each element u in U, denoted by $\mu_A(u)$. $\mu_A(u)$ gives us an assessment of the degree that u belongs to A. Typically, if $\mu_A(u) = 1$ then u definitely belongs to A, whereas $\mu_A(u) \geq 0.8$ means that u is "likely" to be an element of A. In addition, when using Gödel T-norm, T-conorm and Lukasiewicz negation for interpreting conjunctions, disjunctions and complements respectively, we have: for all $u \in U$ and for all fuzzy sets A_1, A_2 with respect to U, $\mu_{A_1 \cap A_2}(u) = \min\{\mu_{A_1}(u), \mu_{A_2}(u)\}$, $\mu_{A_1 \cup A_2}(u) = \max\{\mu_{A_1}(u), \mu_{A_2}(u)\}$, and $\mu_{\bar{A}}(u) = 1 - \mu_A(u)$, where $\bar{A}$ is the complement of A in U.

As for fuzzy logics, the degree of membership $\mu_A(u)$ of an element $u \in U$ w.r.t. the fuzzy set A over U is regarded as the truth-value of the statement "u is in A". Accordingly, in fuzzy DL, (i) a concept C, rather than being interpreted as a classical set, will be interpreted as a fuzzy set and, thus, concepts become imprecise; and, consequently, (ii) the statement "o is in C", i.e. $o : C$, will have a truth-value in [0,1] given by the degree of membership of being the individual o a member of the fuzzy set C.

2.2 Fuzzy $\mathcal{SHIN}$

We introduce the basic terms and notations used throughout this paper. In particular, we introduce the syntax and semantics for fuzzy DL f-$\mathcal{SHIN}$ [4].

Let N_C, N_R, and N_I be countable infinite and pairwise disjoint sets of concept, role and individual names, respectively. We assume that the set of role names N_R can be divided into two disjoint subsets, N_{tR} and N_{nR}, where the former stands for the subset of *transitive role names*, and the latter stands for the subset of *non-transitive role names*. This assumption can be written as $N_R = N_{tR} \cup N_{nR}$ and $N_{tR} \cap N_{nR} = \emptyset$.

f-$\mathcal{SHIN}$ roles (or roles for short) are defined as $R ::= R_N | R^-$, where $R_N \in N_R$, R^- is called the *inverse role* of R.

A role inclusion axiom is of the form $R \sqsubseteq S$, with R, S roles. A role hierarchy (also called a RBox) $\mathcal{R}$ is a finite set of role inclusion axioms.

For the sake of brevity and clarity, we use following notations:

1. To avoid using verbose role expressions of the form R^{--} or even R^{---}, we use an abbreviation $\mathsf{Inv}(R)$ to denote inverse role of R, i.e. $\mathsf{Inv}(R) = R^-$ if $R \in N_R$, and $\mathsf{Inv}(R) = S$ if $R = S^-$ with $S \in N_R$.
2. For a RBox $\mathcal{R}$, we define $\sqsubseteq^*_{\mathcal{R}}$ as the reflexive transitive closure of $\sqsubseteq$ over $\mathcal{R} \cup \{\mathsf{Inv}(R) \sqsubseteq \mathsf{Inv}(S) | R \sqsubseteq S \in \mathcal{R}\}$. We use $R \equiv^*_{\mathcal{R}} S$ as an abbreviation for $R \sqsubseteq^*_{\mathcal{R}} S$ and $S \sqsubseteq^*_{\mathcal{R}} R$.
3. For a RBox $\mathcal{R}$ and a role S, we define the set $\mathsf{Trans}_{\mathcal{R}}$ of transitive roles as $\{S|$ there is a role R with $R \equiv^*_{\mathcal{R}} S$ and $R \in N_{tR}$ or $\mathsf{Inv}(R) \in N_{tR}\}$.
4. A role S is called simple w.r.t. a RBox $\mathcal{R}$ if, for each role R such that $R \sqsubseteq^*_{\mathcal{R}} S$, $R \notin \mathsf{Trans}_{\mathcal{R}}$.

The subscript $\mathcal{R}$ of $\sqsubseteq^*_{\mathcal{R}}$ and $\mathsf{Trans}_{\mathcal{R}}$ is dropped if clear from the context.

f-$\mathcal{SHIN}$ concepts (or concepts for short) are formed out of concept names according to the following abstract syntax, where $A \in N_C, R \in N_R, p \in \mathbb{N}$, and S a simple role:

$$C, D ::= \top | \bot | A | \neg A | C \sqcap D | C \sqcup D | \forall R.C | \exists R.C | \leq pS | \geq pS$$

A TBox is a finite set of *concept definition axioms* of the form $A \equiv D$ and *general concept inclusion* axioms (GCIs) of the form $C \sqsubseteq D$.

An ABox consists of fuzzy assertions of the form $C(o) \bowtie n$, $R(o, o') \rhd n$, or $o \not\approx o'$, where $o, o' \in N_I$, $\bowtie$ stands for any type of inequality, i.e., $\bowtie \in \{\geq, >, \leq, <\}$. We use $\rhd$ to denote $\geq$ or $>$, and $\lhd$ to denote $\leq$ or $<$. We call ABox assertions defined by $\rhd$ *positive assertions*, while those defined by $\lhd$ *negative assertions*. Note that, we consider only positive fuzzy role assertions, since negative role assertions would imply the existence of role negation, which would lead to undecidability [18].

An *f*-$\mathcal{SHIN}$ knowledge base (KB) $\mathcal{K}$ is a triple $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ with $\mathcal{T}$ a TBox, $\mathcal{R}$ a RBox and $\mathcal{A}$ an ABox.

Let C be a fuzzy concept, $\mathcal{R}$ a RBox, we denote by $sub(C, \mathcal{R})$ the set of subconcepts of C. We define $sub(\mathcal{A}, \mathcal{R}) = \bigcup_{C(o) \bowtie n \in \mathcal{A}} sub(C, \mathcal{R})$. $\mathcal{R}$ is dropped if clear from the context.

The semantics of *f*-$\mathcal{SHIN}$ are provided by a fuzzy interpretation, which is a pair $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. Here $\Delta^{\mathcal{I}}$ is a non-empty set of objects, called the domain of interpretation, and $\cdot^{\mathcal{I}}$ is an interpretation function which maps different individual names into different elements in $\Delta^{\mathcal{I}}$, concept name A into membership function $A^{\mathcal{I}}$: $\Delta^{\mathcal{I}} \to$ [0,1], role R into membership function $R^{\mathcal{I}}$:$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \to$ [0,1]. The semantics of *f*-$\mathcal{SHIN}$ concepts and roles are depicted as follows.

- $\top^{\mathcal{I}}(o) = 1$
- $\bot^{\mathcal{I}}(o) = 0$
- $(C \sqcap D)^{\mathcal{I}}(o) = \min\{C^{\mathcal{I}}(o), D^{\mathcal{I}}(o)\}$
- $(C \sqcup D)^{\mathcal{I}}(o) = \max\{C^{\mathcal{I}}(o), D^{\mathcal{I}}(o)\}$
- $(\neg C)^{\mathcal{I}}(o) = 1 - C^{\mathcal{I}}(o)$
- $(\forall R.C)^{\mathcal{I}}(o) = \inf_{o' \in \Delta^{\mathcal{I}}}\{\max\{1 - R^{\mathcal{I}}(o, o'), C^{\mathcal{I}}(o')\}\}$
- $(\exists R.C)^{\mathcal{I}}(o) = \sup_{o' \in \Delta^{\mathcal{I}}}\{\min\{R^{\mathcal{I}}(o, o'), C^{\mathcal{I}}(o')\}\}$

– $(\geq pS)^{\mathcal{I}}(o) = \sup_{o_1,\ldots,o_p \in \Delta^{\mathcal{I}}}\{\min_{i=1}^{p}\{R^{\mathcal{I}}(o,o_i)\}\}$
– $(\leq pS)^{\mathcal{I}}(o) = \inf_{o_1,\ldots,o_{p+1} \in \Delta^{\mathcal{I}}} \max_{i=1}^{p+1}\{1 - R^{\mathcal{I}}(o,o_i)\}$
– $\mathsf{Inv}(R)^{\mathcal{I}}(o,o') = R^{\mathcal{I}}(o',o)$

Given an interpretation $\mathcal{I}$ and an inclusion axiom $C \sqsubseteq D$, $\mathcal{I}$ is *a model of* $C \sqsubseteq D$, if $C^{\mathcal{I}}(o) \leq D^{\mathcal{I}}(o)$ for any $o \in \Delta^{\mathcal{I}}$, written as $\mathcal{I} \models C \sqsubseteq D$. Similarly, for ABox assertions, $\mathcal{I} \models B(o) \bowtie n$ (resp. $\mathcal{I} \models R(o,o') \bowtie n$), iff $B^{\mathcal{I}}(o^{\mathcal{I}}) \bowtie n$ (resp. $R^{\mathcal{I}}(o^{\mathcal{I}}, o'^{\mathcal{I}}) \bowtie n$), and $\mathcal{I} \models o \not\approx o'$ iff $o^{\mathcal{I}} \neq o'^{\mathcal{I}}$. As for RBox, $\mathcal{I} \models R \sqsubseteq S$ iff $\forall\langle o,o'\rangle \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, $R^{\mathcal{I}}\langle o,o'\rangle \leq S^{\mathcal{I}}\langle o,o'\rangle$, and $\mathcal{I} \models \mathsf{Trans}(R)$, iff $\forall o,o',o'' \in \Delta^{\mathcal{I}}$, $R^{\mathcal{I}}(o,o'') \geq \sup_{o' \in \Delta^{\mathcal{I}}}\{\min(R^{\mathcal{I}}(o,o'), R^{\mathcal{I}}(o',o''))\}$. Note that the semantics of transitive roles result from the definition of *sup-min transitive* relation in fuzzy set theory.

If an interpretation $\mathcal{I}$ is a model of all the axioms and assertions in a KB $\mathcal{K}$, we call it *a model of* $\mathcal{K}$. A KB is *satisfiable* iff it has at least one model. A KB $\mathcal{K}$ *entails* (logically implies) a fuzzy assertion φ, written as $\mathcal{K} \models \varphi$, iff all the models of $\mathcal{K}$ are also models of φ.

Given a KB $\mathcal{K}$, we can w.l.o.g assume that the following conditions hold.

1. All concepts are in their negative normal forms (NNFs), i.e. negation occurs only in front of concept names. Through de Morgan law, the duality between existential restriction ($\exists R.C$) and universal restriction ($\forall R.C$), and the duality between atmost restriction ($\leq pS$) and atleast restriction ($\geq pS$), each concept can be transformed into its equivalent NNF by pushing negation inwards.
2. All fuzzy concept assertions are in their positive inequality normal forms (PINFs). A negative concept assertion can be transformed into its equivalent PINF by applying fuzzy complement operation on it. For example, $C(o) < n$ is converted to $\neg C(o) > 1 - n$.
3. All fuzzy assertions are in their normalized forms (NFs). As is shown in [21], by introducing a positive, infinite small value ϵ, a fuzzy assertion of the form $C(o) > n$ can be normalized to $C(o) \geq n + \epsilon$. The model equivalence of KB $\mathcal{K}$ and $\mathcal{K}$'s normalized form is also proved in [21].
4. There are only fuzzy GCIs in the TBox. A fuzzy concept definition axiom $A \equiv D$ can be eliminated by replacing every occurrence of A with D. The elimination is also known as knowledge base expansion. Note that the size of the expansion can be exponential in the size of the TBox. But if we follow the principle of "Expansion is done on demand" [22], the expansion will have no impact on the algorithm complexity of deciding fuzzy query entailment.

2.3 Fuzzy Conjunctive Queries

In general, existing fuzzy DL reasoners can provide most of the basic fuzzy inference services [3], such as checking of fuzzy concept satisfiability, fuzzy concept subsumption, and ABox consistency. In addition, some fuzzy DL reasoners support different kinds of simple queries over a KB $\mathcal{K}$ for obtaining assertional knowledge. These queries include

– *retrieval* given a fuzzy KB $\mathcal{K}$, a fuzzy concept C, and $n \in (0,1]$, to retrieve all instances o occurring in the ABox, such that $\mathcal{K}| = C(o) \geq n$ holds,
– *realisation* given a fuzzy KB $\mathcal{K}$, a individual name o, and $n \in (0,1]$, to determine the most specific concept C, such that $\mathcal{K}| = C(o) \geq n$ holds. In other words, for any fuzzy concept D, $\mathcal{K}| = D(o) \geq n$ implies $\mathcal{K}| = C \sqsubseteq D$.
– *instantiation* given a fuzzy KB $\mathcal{K}$, a fuzzy concept assertion $C(o) \geq n$, to decide whether or not $\mathcal{K}| = C(o) \geq n$.

In fact, fuzzy DL reasoners deal with these queries by transferring them into basic inference tasks. For example, the instantiation problem $\mathcal{K}| = C(o) \geq n$ can be reduced to the (un)satisfiability problem of the KB $\mathcal{K} \cup \{\neg C(o) < n\}$, while the latter one is a basic inference problem. There is, however, no support for queries that ask for n-tuples of related individuals or for the use of variables to formulate a query, just as conjunctive queries do.

Conjunctive queries stemmed from the domain of relational databases, and have attracted much attention recently in Semantic Web. With the emergence of a good number of large-scale domain ontologies, it is of particular importance to provide users with expressive querying service.

Recently there have been quite a lot of work on answering Conjunctive queries over fuzzy DLs. In [14], Straccia defined a conjunctive query language over fuzzy KBs. Conjunctive queries over a fuzzy KB $\mathcal{K}$ are expressions of the form $q(\boldsymbol{x}) \leftarrow \exists \boldsymbol{y}.conj(\boldsymbol{x}, \boldsymbol{y})$, where vector $\boldsymbol{x}$ is constituted by *distinguished variables* (also known as *answer variables*), which can be bound with individual names in a given knowledge base to answer the conjunctive queries, $\boldsymbol{y}$ is constituted by *non-distinguished* variables, which are treated as existential quantified, i.e., we just require the existence of a suitable element in the model, but this element does not have to correspond to an individual explicitly named in the ABox. $conj(\boldsymbol{x}, \boldsymbol{y})$ is a conjunction of atoms of the form $B(z)$ or $R(z_1, z_2)$, where B and R are basic concept and role in $\mathcal{K}$ respectively. The concept and role atoms are syntactically equal to concept and role assertions except that z, z_1 and z_2 may be variables in $\boldsymbol{x}$ or $\boldsymbol{y}$, besides constants in $\mathcal{K}$.

The query language in [14] has the same syntax as that of crisp DLs, and thus cannot express queries such as "find me hotels that are very close to the conference venue (with membership degree at lest 0.9) and offer inexpensive (with membership degree at lest 0.7) rooms". For this reason, a query language, which allows a threshold for every query atom, was proposed in [23]. Clearly, threshold queries give users more flexibility in that users can specify different thresholds for different atoms.

In this study, we mainly deal with conjunctive query entailment problem. As is generally believed in database community, however, there is a tight connection among the problems of conjunctive query containment, conjunctive query answering, and query entailment [24].

2.4 Fuzzy Query Language

Following [18], we provide the formal definition of the syntax and semantics of the fuzzy querying language used in this paper.

Let N_V be a countable infinite set of variables and is disjoint from N_C, N_R, and N_I. A *term* t is either an individual name from N_I or a variable name from N_V. A *fuzzy query atom* is an expression of the form $\langle C(t) \geq n \rangle$ or $\langle R(t, t') \geq m \rangle$ with C a concept, R a simple role, and t, t' terms. As with fuzzy assertions, we refer to these two different types of atoms as *fuzzy concept atoms* and *fuzzy role atoms*, respectively. We also w.l.o.g. assume all the query atoms are in their NNFs, PINFs, and NFs.

Definition 1. *(Fuzzy Boolean Conjunctive Queries) A fuzzy boolean conjunctive query q is a non-empty set of fuzzy query atoms of the form $q = \{\langle \mathrm{at}_1 \geq n_1 \rangle, \ldots, \langle \mathrm{at}_k \geq n_k \rangle\}$. Then for every fuzzy query atom, we can say $\langle \mathrm{at}_i \geq n_i \rangle \in q$.*

We use $\mathsf{Var}(q)$ to denote the set of variables occurring in q, $\mathsf{Ind}(q)$ to denote the set of individual names occurring in q, and $\mathsf{Term}(q)$ for the set of terms in q, i.e. $\mathsf{Term}(q) = \mathsf{Var}(q) \cup \mathsf{Ind}(q)$.

The semantics of a fuzzy query is given in the same way as for the related fuzzy DL by means of fuzzy interpretation consisting of an interpretation domain and a fuzzy interpretation function.

Definition 2. *(Models of Fuzzy Queries) Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a fuzzy interpretation of an f-$\mathcal{SHIN}$ KB, q a fuzzy boolean conjunctive query, and t, t' terms in q. We say $\mathcal{I}$ is a model of q, if there exists a mapping $\pi : \mathsf{Term}(q) \to \Delta^{\mathcal{I}}$ such that $\pi(a) = a^{\mathcal{I}}$ for each $a \in \mathsf{Ind}(q)$, $C^{\mathcal{I}}(\pi(t)) \geq n$ for each fuzzy concept atom $C(t) \geq n \in q$, $R^{\mathcal{I}}(\pi(t), \pi(t')) \geq n$ for each fuzzy role atom $R(t, t') \geq n \in q$.*

If $\mathcal{I} \models^{\pi} at$ for every atom $at \in q$, we write $\mathcal{I} \models^{\pi} q$. If there is a π, such that $\mathcal{I} \models^{\pi} q$, we say $\mathcal{I}$ satisfies q, written as $\mathcal{I} \models q$. We call such a π a *match* of q in $\mathcal{I}$. If $\mathcal{I} \models q$ for each model $\mathcal{I}$ of a KB $\mathcal{K}$, then we say $\mathcal{K}$ entails q, written as $\mathcal{K} \models q$. The *query entailment problem* is defined as follows: given a knowledge base $\mathcal{K}$ and a query q, decide whether $\mathcal{K} \models q$.

3 Query Entailment Algorithm

As for basic inference services and simple queries, our algorithm for deciding fuzzy query entailment is also based on tableau algorithms. The query entailment problem is, however, not reducible to the knowledge base satisfiability problem, since the negation of a fuzzy conjunctive query is not expressible with existing constructors provided by an f-$\mathcal{SHIN}$ knowledge base. For this reason, tableau algorithms for reasoning over knowledge bases is not sufficient. A knowledge base $\mathcal{K}$ may have infinitely many possibly infinite models, whereas tableau algorithms construct only a subset of finite models of the knowledge base. As is defined in Section 2.4, the query entailment holds only if the query is true in all models of the knowledge base, we thus have to show that inspecting only a subset of the models, namely the *canonical* ones, suffices to decide query entailment.

As with the tableau algorithm for f-$\mathcal{SHIN}$ [4], our algorithm works on a data structure called *completion forest*. A completion forests is a finite relational structure capturing sets of models of a KB $\mathcal{K}$. Roughly speaking, models of $\mathcal{K}$

are represented by an initial completion forest $\mathcal{F}_{\mathcal{K}}$. Then, by applying *expansion rules* repeatedly, new completion forests are generated. Since every model of $\mathcal{K}$ is preserved in some completion forest that results from the expansion, $\mathcal{K} \models q$ can be decided by considering a set $\mathbb{F}_{\mathcal{K}}$ of sufficiently expanded forests. From each such $\mathcal{F}$ a single *canonical model* is constructed. Semantically, the finite set of these canonical models is sufficient for answering all queries q of bounded size. Furthermore, we prove that entailment in the canonical model obtained from $\mathcal{F}$ can be checked effectively via a syntactic mapping of the terms in q to the nodes in $\mathcal{F}$.

3.1 Completion Forests

Definition 3. *(Completion Forest) A* completion tree T for a *f*-$\mathcal{SHIN}$ KB *is a tree all whose nodes are generated by expansion rules, except for the root node which might correspond to a individual name in N_I. A* completion forest $\mathcal{F}$ for a *f*-$\mathcal{SHIN}$ KB *consists of a set of completion trees whose nodes correspond to individual names in the ABox, an equivalent relation $\approx$ among nodes, and an inequivalent relation $\not\approx$ among nodes.*

Each node x in a completion forest (which is either a root node or a node in a completion tree) is labelled with a set $\mathcal{L}(x) = \{\langle C, \geq, n\rangle\}$, where $C \in sub(\mathcal{A})$, $n \in (0{,}1]$. Each edge $\langle x, y\rangle$ (which is either one between root nodes or one inside a completion tree) is labelled with a set $\mathcal{L}(\langle x, y\rangle) = \{\langle R, \geq, n\rangle\}$.

If $\langle x, y\rangle$ is an edge in a completion forest with $\langle R', \geq, n\rangle \in \mathcal{L}(\langle x, y\rangle)$ and $R' \sqsubseteq^* R \in \mathcal{R}$, then y is called an $R_{\geq,n}$-*successor* of x and x is called an $R_{\geq,n}$-*predecessor* of y. Ignoring the inequality and membership degree, we can also call y an R-successor of x and x an R-predecessor of y. *Ancestor* and *descendant* are the transitive closure of predecessor and successor, respectively. The union of the successor and predecessor relation is the *neighbor* relation. The *distance* between two nodes x, y in a completion forest is the shortest path between them.

Starting with an *f*-$\mathcal{SHIN}$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A}\rangle$, the completion forest $\mathcal{F}_{\mathcal{K}}$ is initialized such that it contains a root node o, with $\mathcal{L}(o) = \{\langle C, \geq, n\rangle \mid C(o) \geq n \in \mathcal{A}\}$, for each individual name o occurring in $\mathcal{A}$, and an edge $\langle o, o'\rangle$ with $\mathcal{L}(\langle o, o'\rangle) = \{\langle R, \geq, n\rangle \mid \langle R(o, o') \geq n\rangle \in \mathcal{A}\}$, for each pair $\langle o, o'\rangle$ of individual names for which the set $\{R \mid R(o, o') \geq n \in \mathcal{A}\}$ is non-empty. We initialize the relation $\not\approx$ as $\{\langle o, o'\rangle | o \not\approx o' \in \mathcal{A}\}$, and the relation $\approx$ to be empty.

Now we can formally define a new blocking condition, called *k-blocking*, for fuzzy query entailment depending on a depth parameter $k \geq 0$.

Definition 4. *(k-tree equivalence) The k-tree of a node v in T, denoted as T_v^k, is the subtree of T rooted at v with all the descendants of v within distance k. We use $\mathsf{Nodes}(T_v^k)$ to denote the set of nodes in T_v^k. Two nodes v and w in T are said to be k-tree equivalent in T, if T_v^k and T_w^k are isomorphic, i.e., there exists a bijection $\psi : \mathsf{Nodes}(T_v^k) \to \mathsf{Nodes}(T_w^k)$ such that (i) $\psi(v) = w$, (ii) for every node $o \in \mathsf{Nodes}(T_v^k)$, $\mathcal{L}(o) = \mathcal{L}(\psi(o))$, (iii) for every edge connecting two nodes o and o' in T_v^k, $\mathcal{L}(\langle o, o'\rangle) = \mathcal{L}(\langle \psi(o), \psi(o')\rangle)$.*

Definition 5. *(k-witness) A node w is a k-witness of a node v, if v and w are k-tree equivalent in T, w is an ancestor of v in T and v is not in T_w^k. Furthermore, T_w^k tree-blocks T_v^k and each node o in T_w^k tree-blocks node $\psi^{-1}(o)$ in T_v^k.*

Definition 6. *(k-blocking) A node o is k-blocked in a completion forest $\mathcal{F}$ iff it is not a root node and it is either directly or indirectly k-blocked. Node o is* directly k-blocked *iff none of its ancestors is k-blocked, and o is a leaf of a tree-blocked k-tree. Node o is* indirectly k-blocked *iff one of its ancestors is k-blocked or it is a successor of a node o' and $\mathcal{L}(\langle o', o\rangle) = \emptyset$.*

An initial completion forest is expanded according to a set of *expansion rules* that reflect the constructors allowed in f-$\mathcal{SHIN}$. The expansion rules, which syntactically decompose the concepts in node labels, either infer new constraints for a given node, or extend the tree according to these constraints (see Table 1). Termination is guaranteed by k-blocking. We denote by $\mathbb{F}_{\mathcal{K}}$ the set of all completion forests obtained this way.

There should be some explanation for the $\sqsubseteq$-rule, which is applicable for fuzzy GCIs in the TBox. In the context of crisp DLs, the solution to GCIs is, for each $C \sqsubseteq D$, to construct an universal concept $\neg C \sqcup D$ and let the label of every node within the completion forests contain this universal concept, thus ensuring the model satisfies the TBox. In fuzzy cases, however, this solution is not feasible in that $\neg C \sqcup D$ cannot capture the semantics of $C \sqsubseteq D$. Li et al. [25] and Stoilos et al. [21] proposed two similar methods for dealing with fuzzy GCIs in parallel. In the $\sqsubseteq$-rule, $N^{\mathcal{A}}$ and N^q denotes the sets of membership degrees in the ABox $\mathcal{A}$ and in CQ q, respectively, i.e., $N^{\mathcal{A}} = X^{\mathcal{A}} \cup \{1 - n | n \in X^{\mathcal{A}}\}$, where $X^{\mathcal{A}} = \{0, 0.5, 1\} \cup \{n | C(o) \geq n \in \mathcal{A}$, or $R(o, o') \geq n \in \mathcal{A}\}$, $N^q = \{n | C(t) \geq n \in q$, or $R(t, t') \geq n \in q\}$. In addition, for each fuzzy concept C occurring in q, we augment the TBox with a fuzzy GCI $C \sqsubseteq C$. This clearly has no logical impact on the knowledge base, but it ensures that, for each node o in the completion forest, a decision is made as to whether $C(o) \geq n$ or $\neg C(o) \geq 1-n+\epsilon$ ($C(o) < n$) holds, according to the $\sqsubseteq$-rule in Table 1.

For a node o, $\mathcal{L}(o)$ is said to contain a *clash*, if it contain one of the following: (i) a pair of triples $\langle C, \geq, n\rangle$ and $\langle \neg C, \geq, m\rangle$ with $n + m > 1$, (ii) one of the triples: $\langle \bot, \geq, n\rangle$ with $n > 0$, $\langle C, \geq, n\rangle$ with $n > 1$, (iii) some triple $\langle \leq pR, \geq, n\rangle$, and o has $p+1$ $R_{\geq, n'}$-neighbors $o_1, \ldots, o_{p+1}$, with $o_i \not\approx o_j$ for all $1 \leq i < j \leq p+1$ and $n' = 1 - n + \epsilon$.

Definition 7. *(clash-free completion forest) A completion forest $\mathcal{F}$ is called* clash free *if none of its nodes and edges contains a clash.*

Definition 8. *(k-complete completion forest) A completion forest is called k-*complete *if (under k-blocking) no rule can be applied to it. We denote by $\mathtt{ccf}_k(\mathbb{F}_{\mathcal{K}})$ the set of k-complete and class-free completion forests in $\mathbb{F}_{\mathcal{K}}$.*

Example 1. Let $\mathcal{K} = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ be an f-$\mathcal{SHIN}$ KB with $\mathcal{T} = \{C \sqsubseteq \exists R.C\}$, $\mathcal{A} = \{C(a) \geq n\}$ and $\mathcal{R} = \emptyset$. Figure 1 shows a 1-complete and clash-free completion forest $\mathcal{F}$ for $\mathcal{K}$. The node x_4 in $T_{x_3}^1$-tree is directly blocked by x_2 in $T_{x_1}^1$-tree, indicated by the dashed line.

Table 1. Expansion rules

Rule	Description
$\sqcap_{\geq}$	if 1. $\langle C \sqcap D, \geq, n\rangle \in \mathcal{L}(x)$, x is not indirectly k-blocked, and 2. $\{\langle C, \geq, n\rangle, \langle D, \geq, n\rangle\} \nsubseteq \mathcal{L}(x)$ then $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{\langle C, \geq, n\rangle, \langle D, \geq, n\rangle\}$
$\sqcup_{\geq}$	if 1. $\langle C \sqcup D, \geq, n\rangle \in \mathcal{L}(x)$, x is not indirectly k-blocked, and 2. $\{\langle C, \geq, n\rangle, \langle D, \geq, n\rangle\} \cap \mathcal{L}(x) = \emptyset$ then $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C'\}$, where $C' \in \{\langle C, \geq, n\rangle, \langle D, \geq, n\rangle\}$
$\exists_{\geq}$	if 1. $\langle \exists R.C, \geq, n\rangle \in \mathcal{L}(x)$, x is not k-blocked. 2. x has no $R_{\geq,n}$-neighbor y s.t. $\langle C, \geq, n\rangle \in \mathcal{L}(y)$, then create a new node y with $\mathcal{L}(x,y) = \{\langle R, \geq, n\rangle\}$ and $\mathcal{L}(y) = \{\langle C, \geq, n\rangle\}$
$\forall_{\geq}$	if 1. $\langle \forall R.C, \geq, n\rangle \in \mathcal{L}(x)$, x is not indirectly k-blocked. 2. x has an $R_{\geq,n'}$-neighbor y with $\langle C, \geq, n\rangle \notin \mathcal{L}(y)$, where $n' = 1 - n + \epsilon$, then $\mathcal{L}(y) \rightarrow \mathcal{L}(y \cup \{\langle C, \geq, n\rangle\}$
$\forall_{+}$	if 1. $\langle \forall R.C, \geq, n\rangle \in \mathcal{L}(x)$ with $\mathsf{Trans}(R)$, x is not indirectly k-blocked, and 2. x has an $R_{\geq,n'}$-neighbor y with $\langle \forall R.C, \geq, n\rangle \notin \mathcal{L}(y)$, where $n' = 1 - n + \epsilon$, then $\mathcal{L}(y) \rightarrow \mathcal{L}(y \cup \{\langle \forall R.C, \geq, n\rangle\}$
$\forall'_{+}$	if 1. $\langle \forall S.C, \geq, n\rangle \in \mathcal{L}(x)$, x is not indirectly k-blocked, and 2. there is some R, with $\mathsf{Trans}(R)$ and $R \sqsubseteq^{*} S$, 3. x has an $R_{\geq,n'}$-neighbor y with $\langle \forall R.C, \geq, n\rangle \notin \mathcal{L}(y)$, where $n' = 1 - n + \epsilon$, then $\mathcal{L}(y) \rightarrow \mathcal{L}(y \cup \{\langle \forall R.C, \geq, n\rangle\}$
$\geq_{\geq}$	if 1. $\langle \geq pR, \geq, n\rangle \in \mathcal{L}(x)$, x is not k-blocked, $\sharp\{x_i \in N_I \mid \langle R, \geq, n\rangle \in \mathcal{L}(x, x_i)\} < p$, then introduce new nodes, s.t. $\sharp\{x_i \in N_I \mid \langle R, \geq, n\rangle \in \mathcal{L}(x, x_i)\} \geq p$
$\leq_{\geq}$	if 1. $\langle \leq pR, \geq, n\rangle \in \mathcal{L}(x)$, x is not indirectly k-blocked, 2. $\sharp\{x_i \in N_I \mid \langle R, \geq, 1 - n + \epsilon\rangle \in \mathcal{L}(x, x_i)\} > p$ and 3. there exist x_l and x_k, with no $x_l \not\approx x_k$, 4. x_l is neither a root node nor an ancestor of x_k. then (i) $\mathcal{L}(x_k) \rightarrow \mathcal{L}(x_k) \cup \mathcal{L}(x_l)$ (ii) $\mathcal{L}(x, x_k) \rightarrow \mathcal{L}(x, x_k) \cup \mathcal{L}(x, x_l)$ (iii) $\mathcal{L}(x, x_l) \rightarrow \emptyset$, $\mathcal{L}(x_l) \rightarrow \emptyset$ (iv) set $x_i \not\approx x_k$ for all x_i with $x_i \not\approx x_l$
$\leq_{r\geq}$	if 1. $\langle \leq pR, \geq, n\rangle \in \mathcal{L}(x)$, 2. $\sharp\{x_i \in N_I \mid \langle R, \geq^{-}, 1 - n\rangle \in \mathcal{L}(x, x_i)\} > p$ and 3. there exist x_l and x_k, both root nodes, with no $x_l \not\approx x_k$, then 1. $\mathcal{L}(x_k) \rightarrow \mathcal{L}(x_k) \cup \mathcal{L}(x_l)$ 2. For all edges $\langle x_l, x'\rangle$, i. if the edge $\langle x_k, x'\rangle$ does not exist, create it with $\mathcal{L}(\langle x_k, x'\rangle) = \emptyset$, ii. $\mathcal{L}(\langle x_k, x'\rangle) \rightarrow \mathcal{L}(\langle x_k, x'\rangle) \cup \mathcal{L}(\langle x_l, x'\rangle)$. 3. For all edges $\langle x', x_l\rangle$, i. if the edge $\langle x', x_k\rangle$ does not exist, create it with $\mathcal{L}(\langle x', x_k\rangle) = \emptyset$, ii. $\mathcal{L}(\langle x', x_k\rangle) \rightarrow \mathcal{L}(\langle x', x_k\rangle) \cup \mathcal{L}(\langle x', x_l\rangle)$. 4. Set $\mathcal{L}(x_l) = \emptyset$ and remove all edges to/from x_l. 5. Set $x'' \not\approx x_k$ for all x'' with $x'' \not\approx x_l$ and set $x_l \approx x_k$
$\sqsubseteq$	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 $n \in N^{\mathcal{A}} \cup N^{q}$, then $\mathcal{L}(x) \rightarrow \mathcal{L}(x) \cup \{C'\}$ for some $C' \in \{\langle \neg C, \geq, 1 - n + \epsilon\rangle, \langle D, \geq, n\rangle\}$

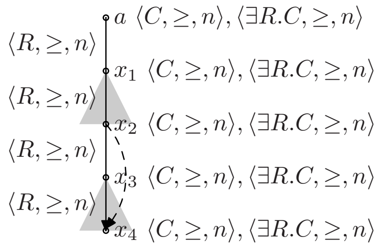

Fig. 1. A 1-complete and clash-free completion forest $\mathcal{F}$ for $\mathcal{K}$

3.2 Models of a Completion Forest

We now show that every model of a KB $\mathcal{K}$ is preserved in some complete and clash-free completion tree $\mathcal{F}$. We first define models of $\mathcal{F}$, then prove that, for each model $\mathcal{I}$ of $\mathcal{K}$, there exists some $\mathcal{F}$, such that a extended model $\mathcal{I}'$ of $\mathcal{I}$ is a model of $\mathcal{F}$.

If we view all the nodes (either root nodes or generated nodes) in a completion forest $\mathcal{F}$ as individual names, we can define models of $\mathcal{F}$ in terms of models of $\mathcal{K}$ over an extended vocabulary.

Definition 9. *(Models of completion forests) An interpretation $\mathcal{I}$ is a model of a completion forest $\mathcal{F}$ for $\mathcal{K}$, denoted $\mathcal{I} \models \mathcal{F}$, if $\mathcal{I} \models \mathcal{K}$ and for all nodes v, w in $\mathcal{F}$ it holds that (i) $C^{\mathcal{I}}(v^{\mathcal{I}}) \geq n$ if $\langle C, \geq, n\rangle \in \mathcal{L}(v)$, (ii) $R^{\mathcal{I}}(v^{\mathcal{I}}, w^{\mathcal{I}}) \geq n$ if there exists an edge $\langle v, w\rangle$ in $\mathcal{F}$ and $\langle R, \geq, n\rangle \in \mathcal{L}(\langle v, w\rangle)$, (iii) $v^{\mathcal{I}} \neq w^{\mathcal{I}}$ if $v \not\approx w \in \mathcal{F}$.*

Apparently, the initial completion forest $\mathcal{F}_{\mathcal{K}}$ and $\mathcal{K}$ share the same models in that there are only root nodes in $\mathcal{F}_{\mathcal{K}}$, which correspond individual names in $\mathcal{K}$. Then, each time an expansion rule is applied, every model of $\mathcal{K}$ is preserved in some expanded completion forest [4]. It thus holds that for each model $\mathcal{I}$ of $\mathcal{K}$, there exists some $\mathcal{F} \in \mathtt{ccf}_k(\mathcal{F}_{\mathcal{K}})$ and a model of $\mathcal{F}$ which extends $\mathcal{I}$ (for any $k > 0$). Since the set of k-complete and clash-free completion forests for $\mathcal{K}$ semantically captures $\mathcal{K}$ (modulo new generated nodes), query entailment $\mathcal{K} \models q$ can be transferred to logical consequence of q from completion forests as follows. For any completion forest $\mathcal{F}$ and CQ q, let $\mathcal{F} \models q$ denote that $\mathcal{I} \models q$ for every model $\mathcal{I}$ of $\mathcal{F}$.

Proposition 1. *Let $k > 0$ be arbitrary. Then $\mathcal{K} \models q$ iff $\mathcal{F} \models q$ for each $\mathcal{F} \in \mathtt{ccf}_k(\mathbb{F}_{\mathcal{K}})$.*

3.3 Checking Query Entailment within Completion Forest

Now we will show that, if k is large enough, we can decide $\mathcal{F} \models q$ for each $\mathcal{F} \in \mathsf{ccf}_k(\mathbb{F}_{\mathcal{K}})$ by syntactically mapping the query q into $\mathcal{F}$.

Definition 10. *(Query mapping) A fuzzy query q can be* mapped into $\mathcal{F}$, *denoted $q \hookrightarrow \mathcal{F}$, if there is a mapping $\mu : \mathsf{Terms}(q) \to \mathsf{Nodes}(\mathcal{F})$, such that*

- *$\mu(a) = a$ for every individual name a,*
- *for each fuzzy concept atom $C(x) \geq n$ in q, $\langle C(x), \geq, n\rangle \in \mathcal{L}(\mu(x))$,*
- *for each fuzzy role atom $\langle R(x, y) \geq n\rangle$ in q, $\mu(y)$ is a $R_{\geq n}$-neighbor of $\mu(x)$.*

We use n_q to denote the number of fuzzy role atoms in a fuzzy query q.

Theorem 1. *Let $k \geq n_q$, where n_q denote the number of fuzzy role atoms in a fuzzy query q. Then $\mathcal{K} \models q$ iff for each $\mathcal{F} \in \texttt{ccf}_k(\mathbb{F}_\mathcal{K})$, it holds that $q \hookrightarrow \mathcal{F}$.*

The if direction is easy. If q can be mapped to $\mathcal{F}$ via μ, then q is satisfied in each model $\mathcal{I}$ of $\mathcal{F}$ by assigning to each variable x in q the value of its image $\mu^\mathcal{I}(x)$. By Proposition 1, $\mathcal{K} \models q$.

To prove that the converse also holds, we have to show that if k is large enough, a mapping of q into $\mathcal{F} \in \texttt{ccf}_k(\mathbb{F}_\mathcal{K})$ can be constructed from a distinguished canonical model of $\mathcal{F}$. The canonical model $\mathcal{I}_\mathcal{F}$ of $\mathcal{F}$ is constructed by unravelling the forest $\mathcal{F}$ in the standard way , where the blocked nodes act like 'loops' [4]. Its domain comprises the set of all paths from some root in $\mathcal{F}$ to some node of $\mathcal{F}$ (thus, it can be infinite). Note that in order for $\mathcal{I}_\mathcal{F}$ to be a model, $\mathcal{F}$ must be in $\texttt{ccf}_k(\mathbb{F}_\mathcal{K})$ for some $n \geq 1$. For the complexity of the formal definition of $\mathcal{I}_\mathcal{F}$, we provide an example.

Example 2. By unravelling $\mathcal{F}$ in Figure 1, we obtain a model $\mathcal{I}_\mathcal{F}$ that has as domain the infinite set of paths from a to each x_i. Note that a path actually comprises a sequence of pairs of nodes, in order to witness the loops introduced by the blocked variables. When a node is not blocked, like x_1, the pair $\frac{x_1}{x_1}$ is added to the path. Since $T^1_{x_1}$ tree-blocks $T^1_{x_3}$, every time a path reaches x_4, which is a leaf of a blocked tree, we add $\frac{x_2}{x_4}$ to the path and loop back to the successors of x_2. In this way, we obtain the following infinite set of paths:

$$
\begin{aligned}
p_0 =& \left[\frac{a}{a}\right], p_1 = \left[\frac{a}{a}, \frac{x_1}{x_1}\right], p_2 = \left[\frac{a}{a}, \frac{x_1}{x_1}, \frac{x_2}{x_2}\right],\\
p_3 =& \left[\frac{a}{a}, \frac{x_1}{x_1}, \frac{x_2}{x_2}, \frac{x_3}{x_3}\right], p_4 = \left[\frac{a}{a}, \frac{x_1}{x_1}, \frac{x_2}{x_2}, \frac{x_3}{x_3}, \frac{x_2}{x_4}\right],\\
p_5 =& \left[\frac{a}{a}, \frac{x_1}{x_1}, \frac{x_2}{x_2}, \frac{x_3}{x_3}, \frac{x_2}{x_4}, , \frac{x_3}{x_3}\right], \ldots
\end{aligned}
$$

This set of paths constitute the domain $\Delta^{\mathcal{I}_\mathcal{F}}$. For each concept name A, we have $A^{\mathcal{I}_\mathcal{F}}(p_i) \geq n$, if $\langle A, \geq, n\rangle$ occurs in the label of the last node in p_i. For each role R, $R(p_i, p_j) \geq n$ if the last node in p_j is an R successor of p_i. If role $R \in \mathsf{Tran}$, the extension of R is expanded according to the sup-min transitive semantics. In the following, let n_q denote the number of fuzzy role atoms in q, and let $k \geq n_q$. Since $\mathcal{I}_\mathcal{F} \models q$, there exists a mapping $\sigma : \mathsf{Nodes} \to \Delta^{\mathcal{I}_\mathcal{F}}$ s.t. for each fuzzy concept atom$\langle C(x) \geq n\rangle$ in q, $C^{\mathcal{I}_\mathcal{F}}(\sigma(x)) \geq n$, and for each fuzzy role atom $R(x, y) \geq n$ in q, $R^{\mathcal{I}_\mathcal{F}}(\sigma(x), \sigma(y)) \geq n$. For any k-complete and clash free completion forest $\mathcal{F}$, a mapping μ of q into $\mathcal{F}$ can be obtained from σ. We use $G_{\mathcal{I}_\mathcal{F}}$ to denote the graph that has as nodes the domain of $\mathcal{I}_\mathcal{F}$, and as arcs the

R-successor edges of $\mathcal{I}_\mathcal{F}$ for each role occurring in$q$. For any two nodes q_i and q_j in $\mathcal{I}_\mathcal{F}$, let $d(q_i, q_j)$ denote the distance between q_i and q_j in $G_{\mathcal{I}_\mathcal{F}}$. Let the image of q under σ as a graph G_q, then the length of a path in G_q connecting the images $\sigma(x)$ and $\sigma(y)$ of any two variables x and y in q will be at most n_q. If $\mathcal{F}$ is k-complete with $k \geq n_q$, then for every path in G_q there will be an isomorphic one in $\mathcal{F}$. Therefore, a k-complete completion forest is large enough to find a mapping whose image is isomorphic to G_q.

We can, from the only if direction of Theorem 1, establish our key result, which reduce query entailment $\mathcal{K} \models q$ to finding a mapping of q into every $\mathcal{F}$ in $\mathtt{ccf}_k(\mathbb{F}_\mathcal{K})$.

Example 3. Given $\mathcal{F}$ and $q = \langle R(\langle x, y \rangle) \geq n \rangle \wedge C(x) \geq n$, we can easily recognize a mapping $q \hookrightarrow \mathcal{F}$.

3.4 Complexity Analysis

For the standard reasoning tasks, e.g., knowledge base consistency, the combined complexity is measured in the size of the input knowledge base. For query entailment, the size of the query is additionally taken into account. The size of a knowledge base $\mathcal{K}$ or a query q is simply the number of symbols needed to write it over the alphabet of constructors, concept, role, individual, and variable names that occur in $\mathcal{K}$ or q, where numbers are encoded in binary.

Theorem 2. *Given an f-$\mathcal{SHIN}$ KB $\mathcal{K}$ and a fuzzy conjunctive query q all of whose roles are simple, deciding whether $\mathcal{K} \models q$ is in* co-3NexpTime.

Proof. (*sketch*) The proof is quite similar with the one presented in[26]. We use $||\mathcal{K}, q||$ to denote the total size of the string encoding the knowledge base $\mathcal{K}$ and the query q in a query entailment $\mathcal{K} \models q$. The branches in each completion tree within a completion forest $\mathcal{F} \in \mathbb{F}_\mathcal{K}$ is polynomially bounded in $||\mathcal{K}, q||$, and the maximal height of a non-isomorphic k-tree is double exponential in $||\mathcal{K}, q||$, if k is polynomial in $||\mathcal{K}, q||$. $\mathcal{F}$ thus has at most triple exponentially many nodes. Since each expansion rule can be applied only polynomially often to a node, the expansion of the initial completion forest $\mathcal{F}_\mathcal{K}$ into some $\mathcal{F} \in \mathbb{F}_\mathcal{K}$ terminates in nondeterministic triple exponential time in $||\mathcal{K}, q||$ for $k = n_q$. Checking whether $q \hookrightarrow \mathcal{F}$ is thus in triple exponential time in $||\mathcal{K}, q||$.

As for data complexity, we consider the ABox as the only input for the algorithm, i.e., the size of the TBox, the role hierarchy, and the query is fixed. Therefore, each completion forest $\mathcal{F} \in \mathbb{F}_\mathcal{K}$ has linearly many nodes in $|\mathcal{A}|$ and any expansion of $\mathcal{F}_\mathcal{K}$ terminates in polynomial time. Deciding whether $q \hookrightarrow \mathcal{F}$ is thus polynomial in the size of $\mathcal{F}$.

Theorem 3. *Given an f-$\mathcal{SHIN}$ KB $\mathcal{K}$ and a fuzzy conjunctive query q all of whose roles are simple, deciding whether $\mathcal{K} \models q$ is in* co-NP *w.r.t. data complexity.*

4 Conclusion

Fuzzy Description Logics-based knowledge bases are envisioned to be useful in the Semantic Web. Existing fuzzy DL reasoners either are not capable of answering complex queries (mainly conjunctive queries), or only apply to DLs with less expressivity. We thus present an algorithm for answering expressive fuzzy conjunctive queries, which allow the occurrence of both lower bound and the upper bound of threshold in a query atom, over the relative expressive DL, namely fuzzy $\mathcal{SHIN}$. The algorithm we suggest here can easily be adapted to existing (and future) DL implementations. Future direction concern applying the proposed technique to even more expressive logics, for example fuzzy DLs additionally extended with nominals and datatype groups [27], or to more expressive fuzzy query language as suggested in [15].

Acknowledgments. This work was supported by the National Natural Science Foundation of China (60873010).

References

1. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The description logic handbook: theory, implementation, and applications. Cambridge University Press, New York (2003)
2. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
3. Straccia, U.: Reasoning within fuzzy description logics. J. Artif. Intell. Res. (JAIR) 14, 137–166 (2001)
4. Stoilos, G., Stamou, G., Pan, J., Tzouvaras, V., Horrocks, I.: Reasoning with very expressive fuzzy description logics. Journal of Artificial Intelligence Research 30(8), 273–320 (2007)
5. Stoilos, G., Simou, N., Stamou, G.B., Kollias, S.D.: Uncertainty and the semantic web. IEEE Intelligent Systems 21(5), 84–87 (2006)
6. Calvanese, D., De Giacomo, G., Lenzerini, M.: On the decidability of query containment under constraints. In: Proc. of the 17th ACM SIGACT SIGMOD SIGART Sym. on Principles of Database Systems (PODS 1998), pp. 149–158 (1998)
7. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The dl-lite family. J. of Automated Reasoning 39(3), 385–429 (2007)
8. Rosati, R.: On conjunctive query answering in EL. In: Proceedings of the 2007 International Workshop on Description Logic (DL 2007), CEUR Electronic Workshop Proceedings (2007)
9. Rosati, R.: The limits of querying ontologies. In: Schwentick, T., Suciu, D. (eds.) ICDT 2007. LNCS, vol. 4353, pp. 164–178. Springer, Heidelberg (2006)
10. Krotzsch, M., Rudolph, S., Hitzler, P.: Conjunctive queries for a tractable fragment of OWL 1.1. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 310–323. Springer, Heidelberg (2007)
11. Levy, A.Y., Rousset, M.C.: Combining horn rules and description logics in carin. Artif. Intell. 104(1-2), 165–209 (1998)

12. Glimm, B., Horrocks, I., Lutz, C., Sattler, U.: Conjunctive query answering for the description logic shiq. In: IJCAI, pp. 399–404 (2007)
13. Glimm, B., Horrocks, I., Sattler, U.: Conjunctive query entailment for shoq. In: Proceedings of the 2007 International Workshop on Description Logic, DL 2007 (2007)
14. Straccia, U.: Answering vague queries in fuzzy dl-lite. In: Proceedings of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2006), pp. 2238–2245 (2006)
15. Pan, J.Z., Stamou, G., Stoilos, G., Thomas, E.: Expressive Querying over Fuzzy DL-Lite Ontologies. In: Proc. of 2007 International Workshop on Description Logics, DL 2007 (2007)
16. Cheng, J.W., Ma, Z.M., Zhang, F., Wang, X.: Conjunctive query answering over an f-alc knowledge base. In: Web Intelligence/IAT Workshops, pp. 279–282 (2008)
17. Cheng, J.W., Ma, Z.M., Zhang, F., Wang, X.: Deciding query entailment in fuzzy description logic knowledge bases. In: Bhowmick, S., Küng, J., Wagner, R. (eds.) DEXA 2009. LNCS, vol. 5690, pp. 830–837. Springer, Heidelberg (2009)
18. Mailis, T.P., Stoilos, G., Stamou, G.B.: Expressive reasoning with horn rules and fuzzy description logics. In: Marchiori, M., Pan, J.Z., Marie, C.d.S. (eds.) RR 2007. LNCS, vol. 4524, pp. 43–57. Springer, Heidelberg (2007)
19. Ortiz, M., Calvanese, D., Eiter, T.: Data complexity of query answering in expressive description logics via tableaux. J. Autom. Reasoning 41(1), 61–98 (2008)
20. Cheng, J.W., Ma, Z.M., Zhang, F., Wang, X.: Deciding query entailment for fuzzy shin ontologies. Technical report, Northeastern University (2009), `ftp://202.118.18.134/`
21. Stoilos, G., Straccia, U., Stamou, G.B., Pan, J.Z.: General concept inclusions in fuzzy description logics. In: ECAI, pp. 457–461 (2006)
22. Baader, F., Nutt, W.: Basic description logics. In: Description Logic Handbook, pp. 43–95 (2003)
23. Lukasiewicz, T., Straccia, U.: Top-k retrieval in description logic programs under vagueness for the semantic web. In: Prade, H., Subrahmanian, V.S. (eds.) SUM 2007. LNCS (LNAI), vol. 4772, pp. 16–30. Springer, Heidelberg (2007)
24. Chandra, A.K., Merlin, P.M.: Optimal implementation of conjunctive queries in relational data bases. In: STOC 1977: Proceedings of the ninth annual ACM symposium on Theory of computing, pp. 77–90. ACM, New York (1977)
25. Li, Y., Xu, B., Lu, J., Kang, D.: Discrete tableau algorithms for *shi*. In: Description Logics (2006)
26. Ortiz, M., Calvanese, D., Eiter, T.: Data complexity of answering unions of conjunctive queries in *shiq*. In: Description Logics (2006)
27. Wang, H., Ma, Z.M.: A decidable fuzzy description logic f-alc(g). In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 116–123. Springer, Heidelberg (2008)