Towards Instance Query Answering for Concepts Relaxed by Similarity Measures

Andreas Ecke[*]
Theoretical Computer Science,
TU Dresden, Germany

Rafael Peñaloza[†]
Theoretical Computer Science,
TU Dresden, Germany

Anni-Yasmin Turhan[‡]
Theoretical Computer Science,
TU Dresden, Germany

Center for Advancing Electronics Dresden

Abstract

In Description Logics (DL) knowledge bases (KBs) information is typically captured by crisp concept descriptions. However, for many practical applications querying the KB by crisp concepts is too restrictive. A controlled way of gradually relaxing a query concept can be achieved by the use of similarity measures.

To this end we formalize the task of instance query answering for crisp DL KBs using concepts relaxed by similarity measures. We identify relevant properties for the similarity measure and give first results on a computation algorithm.

1 Introduction

Description Logics (DLs) are a family of knowledge representation formalisms that have unambiguous semantics. A particular DL is characterized by a set of concept constructors, which allow to build complex concept descriptions. Intuitively, *concept descriptions* characterize categories from an application domain. In addition, binary relations on the domain of interest can be captured by *roles*. These in turn can be used in concept descriptions. The terminological knowledge of an application domain is stored in the *TBox*, where complex concept descriptions can be assigned to concept names. Facts from the application domain and relations between them are represented by *individuals* in the *ABox*. TBox and ABox together form the DL *knowledge base* (KB).

The formal semantics of DLs allow the definition of a variety of reasoning services. The most prominent ones are *subsumption*, i.e. to compute whether a sub-concept relationship holds between two concept descriptions and *instance query answering*, where for a given concept description all individuals from an ABox that are instances of the concept are computed. These reasoning services are implemented in highly optimized reasoning systems, see for example [Tsarkov and Horrocks, 2006; Kazakov *et al.*, 2012; Haarslev *et al.*, 2012].

DLs of varying expressivity are the underlying logics for the W3C standardized ontology language OWL 2 and its profiles [Motik *et al.*, 2009]. This has led to an increased use of DLs and DL reasoning systems in the recent years in many application areas. By now there is a large collection of KBs written in these languages. However, many applications need to query the knowledge base in a less strict fashion.

In the application area of service matching OWL TBoxes are employed to describe types of services. Here, a user request for a service specifies several conditions for the desired service. These conditions are represented by a concept description. For such a concept description the OWL ABox that contains the individual services is searched for a service matching the specified request by employing instance query answering. In cases where an exact match with the provided requirements is not possible, a "feasible" alternative needs to be retrieved from the ABox containing the services. This means that those individuals from the ABox should be retrieved for the given query concept that fulfill the main conditions, while for some conditions only a relaxed variant is fulfilled.

A natural idea on how to relax the notion of instance query answering is to simply employ fuzzy DLs and perform query answering on a fuzzy variant of the initial query concept. However, on the one hand reasoning in fuzzy DLs easily becomes undecidable [Borgwardt *et al.*, 2012; Borgwardt and Peñaloza, 2012; Cerami and Straccia, 2013] and on the other hand depending on the user and on the request, different ways of relaxing the query concept are needed. For instance, for a request to a car rental company to rent a particular car model in Beijing, it might be acceptable to get an offer for a similar car model to be rented in Beijing, instead of getting the offer to rent the requested car model in London. Whereas for a handicapped user in a wheelchair it might not be acceptable to relax the requested car model from a two-door one to a four-door one. Here fuzzy concepts would relax the initial concept in an unspecific and uniform way. Ideally, relaxed instance query answering should allow to

1. choose *which aspects* of the query concept can be relaxed and

[*] Supported by the German Research Foundation (DFG) Graduiertenkolleg 1763 (QuantLA).

[†] Partially supported by DFG within the Cluster of Excellence 'cfAED'

[‡] Partially supported by DFG in the Collaborative Research Center 912 "Highly Adaptive Energy-Efficient Computing".

2. choose the *degree* to how much these aspects can be relaxed.

The reasoning service addressed in this paper is a relaxed notion of instance querying, such that it allows for a given query concept the selective and gradual extension of the answer set of individuals. We develop a formal definition of this reasoning service in Section 3.

Our approach for achieving selective and gradual extension of the answer sets is to employ concept similarity measures to relax the query concept. A *concept similarity measure* yields, for a pair of concept descriptions, a value from the interval $[0, 1]$—indicating how similar the concepts are. The goal is to compute for a given concept C, a concept similarity measure $\sim$ and a degree t ($t \in [0, 1]$), a set of concept descriptions such that each of these concepts is similar to C by a degree of at least t, if measured by $\sim$, and finding all their instances.

For DLs there is whole range of similarity measures defined (see for example [Borgida *et al.*, 2005; d'Amato *et al.*, 2005; Lehmann and Turhan, 2012]), which could be employed for this task. In particular the similarity measures generated by the framework described in [Lehmann and Turhan, 2012] allow users to specify which part of the vocabulary used in their knowledge base is to be regarded more important when it comes to the assessment of similarity of concepts. Thus, these measures naturally allow to select which aspect of the query concept to relax.

The core reasoning problem encountered in our algorithm for relaxed instance query answering is to compute for an individual a and the query concept description C a concept description C' that *mimics* C, i.e. a concept description that is 'sufficiently similar' to C w.r.t. the used similarity measure $\sim$ and the degree t.

We propose in this paper an algorithm to compute the above mentioned reasoning service of relaxed instance query answering in the lightweight DL $\mathcal{EL}$. For instance, for the Gene ontology [Gene Ontology Consortium, 2000], which is written in $\mathcal{EL}$ and is used (among other things) to solve the task of finding genes that realize similar functionality [Lord *et al.*, 2003], a proliferation of different similarity measures has been defined [Lord *et al.*, 2003; Schlicker *et al.*, 2006; Mistry and Pavlidis, 2008; Alvarez and Yan, 2011]. In principle these measures could be used in our approach to query ABoxes. We identify properties of concept similarity measures that allow to compute relaxed instances of concepts.

The paper is organized as follows: after introducing basic notions on DLs and concept similarity measures in Section 2, we develop a formal notion of relaxed instances in Section 3. In order to compute relaxed instances it is necessary, as we shall see, to compute mimics of a concept and an individual. An way of finding a mimic and its application to construct an algorithm that computes all relaxed instances of a query concept is provided in Section 4. As customary, the paper ends with conclusions and future work.

2 Preliminaries

In this section we introduce the basic notions of Description Logics and similarity measures between concepts. For a thorough introduction to Description Logics, see [Baader *et al.*, 2003]. While we try to formalize the notion of relaxed instances of a concept w.r.t. a similarity measure independently from a specific DL, Section 4 will show how instance querying for relaxed concepts can be computed in the restricted DL $\mathcal{EL}$.

	Syntax	Semantics
top concept	$\top$	$\top^\mathcal{I} = \Delta^\mathcal{I}$
conjunction	$C \sqcap D$	$(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$
existential restriction	$\exists r.C$	$(\exists r.C)^\mathcal{I} = \{d \in \Delta^\mathcal{I} \mid \exists e.(d, e) \in r^\mathcal{I} \land e \in C^\mathcal{I}\}$
concept definition	$A \equiv C$	$A^\mathcal{I} = C^\mathcal{I}$
concept assertion	$C(a)$	$a^\mathcal{I} \in C^\mathcal{I}$
role assertion	$r(a, b)$	$(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$

Table 1: Concept constructors, TBox axioms and ABox assertions for $\mathcal{EL}$.

Let N_C, N_R, and N_I be non-empty, disjoint sets of *concept names*, *role names*, and *individual names*. A *concept description* (or short concept) is constructed from concept names by applying *concept constructors* such as conjunction, negation, quantification, or the top concept $\top$. In particular, $\mathcal{EL}$ only admits the concept constructors conjunctions, existential restrictions and the top concept, as seen in Table 1. We denote the set of all $\mathcal{L}$-concept descriptions constructed is such a way by $\mathcal{C}(\mathcal{L})$.

For example, using the following $\mathcal{EL}$-concept description, one can describe a service which currently waits for requests, but runs on an overloaded server:

$$\begin{aligned}&\mathsf{Service} \sqcap \exists \mathsf{has\text{-}state}.\mathsf{WaitingForRequest}\\ &\quad \sqcap \exists \mathsf{runs\text{-}on}(\mathsf{Server} \sqcap \exists \mathsf{has\text{-}condition}.\mathsf{Overloaded})\end{aligned}$$

The semantics of concept descriptions is defined by means of *interpretations* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ consisting of a non-empty *domain* $\Delta^\mathcal{I}$ and an *interpretation function* $\cdot^\mathcal{I}$ that assigns binary relations on $\Delta^\mathcal{I}$ to role names, subsets of $\Delta^\mathcal{I}$ to concept names, and elements of $\Delta^\mathcal{I}$ to individual names. The interpretation function can be recursively extended to $\mathcal{EL}$-concept descriptions as shown in Table 1.

An *$\mathcal{EL}$-knowledge base* (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of an *$\mathcal{EL}$-TBox* $\mathcal{T}$, which captures the terminological knowledge, and an *$\mathcal{EL}$-ABox* $\mathcal{A}$, which contains the assertions about specific individual. In this paper we only consider *unfoldable* TBoxes, i.e., sets of concept definitions such that each concept name occurs at most once on the left-hand side of a concept definition and there are no cyclic dependencies between defined concepts. An ABox is a set of concept and role assertions. The semantics of interpretations is extended to concept definitions and assertions as shown in Table 1. We say that an interpretation $\mathcal{I}$ is a model of a TBox $\mathcal{T}$ (ABox $\mathcal{A}$), if it satisfies all concept definition in $\mathcal{T}$ (assertions in $\mathcal{A}$). $\mathcal{I}$ is a model of a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if it is a model for both $\mathcal{T}$ and $\mathcal{A}$.

There exists a number of inferences for DLs. Three commonly used inferences are *concept subsumption*, *concept equivalence* and *instance checking*. Concept subsumption tests if a concept C is subsumed by a concept D w.r.t. a TBox $\mathcal{T}$ (denoted $C \sqsubseteq_{\mathcal{T}} D$), i.e. $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$. Similarly, two concepts C and D are equivalent w.r.t. $\mathcal{T}$ (denoted $C \equiv_{\mathcal{T}} D$), if $C \sqsubseteq_{\mathcal{T}} D$ and $D \sqsubseteq_{\mathcal{T}} C$. Finally, an individual a is an instance of a query concept description C w.r.t. a KB $\mathcal{K}$, if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$.

Besides these standard reasoning tasks, other inferences have been developed for certain applications. The most specific concept, first introduced in [Nebel, 1990], is such a non-standard inference. This inference computes a concept description that describes an individual a from the knowledge base as exact as it is possible in the used DL.

Definition 1. Let $\mathcal{L}$ be a DL and $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{L}$-KB. The concept description C is the *most specific concept* of an individual a w.r.t. $\mathcal{K}$ (denoted $\text{msc}(a)$) iff

- a is an instance of C, and
- for all concept descriptions $D \in \mathcal{C}(\mathcal{L})$, if a is an instance of D, then $C \sqsubseteq_{\mathcal{T}} D$.

Similarity measures. For a DL $\mathcal{L}$, a *concept similarity measure* $\sim\colon \mathcal{C}(\mathcal{L}) \times \mathcal{C}(\mathcal{L}) \to [0,1]$ is a function that assigns a similarity value $C \sim D$ to each pair C, D of $\mathcal{L}$-concept descriptions. A value $C \sim D = 0$ means that C and D are totally dissimilar, while a value $C \sim D = 1$ means that C and D are totally similar.

A collection of properties for concept similarity measures is given in [Lehmann and Turhan, 2012]. In particular, a similarity measure $\sim$ for $\mathcal{L}$-concept descriptions is:

1. *symmetric* iff $C \sim D = D \sim C$ for all $C, D \in \mathcal{C}(\mathcal{L})$;
2. fulfilling the *triangle inequality* iff
$$1 \,+\, D \sim E \;\geq\; D \sim C \,+\, C \sim E$$
for all $C, D, E \in \mathcal{C}(\mathcal{L})$;
3. *equivalence invariant* iff for all $C, D, E \in \mathcal{C}(\mathcal{L})$ with $C \equiv D$ it holds that $C \sim E = D \sim E$;
4. *equivalence closed* iff $C \sim D = 1 \iff C \equiv D$.

In this paper, we only consider symmetric similarity measures, since they better capture our intuitive understanding of similarity. However, all definitions and results can easily be extended to asymmetric similarity measures. Furthermore, the triangle inequality was found to be hard to achieve for similarity measures for even restricted DLs like $\mathcal{EL}$, and thus will not be discussed here.

Observe that the property ‘equivalence closed’ interacts with relaxed instances of a query concept C in the following way: clearly, if we want only relaxed instances with a similarity of exactly 1, then equivalence closed similarity measures should result in exactly the instances of C, while similarity measures that are not equivalence closed might result in additional individuals.

Most previously proposed concept similarity measures can be divided into two groups: *structural measures*, which are defined using the syntax of the concepts, and *interpretation based measures*, which are defined using interpretations and cardinality instead of the syntax. We later describe a result for structural similarity measures, therefore we will describe these in more detail: Basically, a similarity measure $\sim$ on $\mathcal{L}$-concepts descriptions is called structural, if it computes the similarity of two concepts C and D recursively by computing the similarity of concept names in C and D and the similarity of the existential restrictions occurring in C and D and combining these values monotonically to the overall similarity. For structural similarity measures to be equivalence invariant, the concepts often need to be transformed into a normal form before comparing them [Lehmann and Turhan, 2012]. For a similarity measure $\sim$, we call the normal form used for the computation of the similarity the $\sim$*-normal form*.

3 Relaxed Instances

In this section we introduce the main reasoning problems that we want to solve, as well as a first approach for obtaining a solution.

Our main goal is to generalize query answering to allow for more relaxed solutions. Intuitively, given a concept C, we are interested in finding all the certain instances of C, but also in finding those individuals that are *close* to being instances of C; we call these individuals the *relaxed instances* of C. To emphasize the contrast, we some times call the instances of C *certain instances* of C.

Before we can try to compute these relaxed instances, we need to formalize the notion of relaxed instances of a query concept. In principle there are are many ways to do so and we discuss next some of these options.

One natural approach would be to try to decide which individuals are *similar* to any of the certain instances of C. However, this method would require the definition of a similarity measure on the *elements* of the domain, rather than on the concepts. Such a DL with a similarity measure on the domain elements was introduced in [Lutz *et al.*, 2003]. However, for this DL the similarity measure (or more precisely, a distance metric) is part of the interpretation and cannot be adjusted to different user needs.

A different idea that has been proposed is to simply generalize the concept C by considering named concepts that subsume C. Thus for a named concept C, consider its direct subsumers in the concept hierarchy. This idea is easy to implement and understand, but provides only very rough approximations to the concept C determined by the set of concept names only. Moreover, users have no control on the quality of the approximation provided; in fact even the direct subsumers might describe a concept that is already very dissimilar to C.

We follow a different approach, in which we ask for the instances of those concepts that are similar to C. We can then control how inclusive the relaxed instance solutions should be, by adjusting the degree t of similarity allowed.

Definition 2 (relaxed instance)**.** Let $\mathcal{L}$ be some DL, C be an $\mathcal{L}$-concept, $\sim$ a similarity measure over $\mathcal{L}$-concepts, and $t \in (0,1]$. The individual $a \in N_I$ is a *relaxed instance* of C w.r.t. the $\mathcal{L}$-knowledge base $\mathcal{K}$, $\sim$ and the threshold t, denoted $a \in_t^{\sim} C$, iff there exists a concept description $X \in \mathcal{C}(\mathcal{L})$ such that $C \sim X \geq t$ and $a \in X^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$.

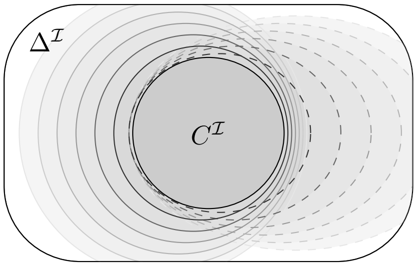

Figure 1: Relaxed instances w.r.t. two different similarity measures. Darker colors represent the relaxed instances of C w.r.t. higher degrees t.

For brevity, we will denote as $\text{Relax}_t^{\sim}(C)$ the set of all relaxed instances of C w.r.t. $\mathcal{K}$, $\sim$ and t. Clearly, the elements of $\text{Relax}_t^{\sim}(C)$ depend strongly on the value of t, but also on the similarity measure $\sim$ chosen, as shown in Figure 1. For a fixed similarity measure $\sim$, if $t \leq t'$, then it holds that $\text{Relax}_{t'}^{\sim}(C) \subseteq \text{Relax}_t^{\sim}(C)$. In the figure, the central circle represents the interpretation of the concept C. The other lines show the interpretation of $\text{Relax}_t^{\sim}(C)$ with darker lines gradually representing large values t. We use two different kinds of lines (continuous vs. dashed) to represent two different similarity measures, that relax the concepts based on different features. As can be seen, the sets obtained can greatly differ from each other.

As mentioned before, our goal is to find all the instances in $\text{Relax}_t^{\sim}(C)$. Following Definition 2, this task could be performed by first computing all concepts X that are similar to C with degree at least t, and then obtaining all the instances of these concepts X; in symbols,

$$\text{Relax}_t^{\sim}(C) = \bigcup_{C \sim X \geq t} \{a \mid a \text{ is an instance of } X\}.$$

However, this approach suffers from two main drawbacks. First, the set of all concepts that are similar to C with degree at least t might be infinite, thus requiring an infinite number of queries to obtain $\text{Relax}_t^{\sim}(C)$, even though this set contains only finitely many individuals. Second, it is not known how to compute the similar concepts X. Similarity measures tell us only how similar two given concepts are, but not how to build a concept that is similar to another with at least some given degree.

To avoid these issues, we consider a different reasoning problem, that considers the computation of a concept that has a given individual a as an instance and resembles C most. We call this the *mimic* of C w.r.t. a.

Definition 3 (mimic)**.** Let $\mathcal{L}$ be a DL, $\mathcal{K}$ be an $\mathcal{L}$-knowledge base, $a \in N_I$ be an individual name, C be an $\mathcal{L}$-concept description, and $\sim$ be a similarity measure. An $\mathcal{L}$-concept D is called a *mimic* of C w.r.t. a, denoted $\mathfrak{M}(C, a)$, iff the following two conditions hold:

- a is an instance of D, i.e., $a^{\mathcal{I}} \in D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$, and
- for all $\mathcal{L}$-concept descriptions E holds, if a is an instance of E, then $C \sim D \geq C \sim E$.

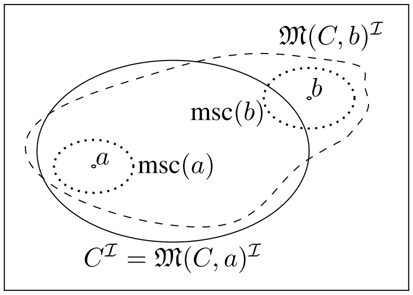
$\mathfrak{M}(C,b)^{\mathcal{I}}$
b
msc(b)
a
msc(a)
$C^{\mathcal{I}} = \mathfrak{M}(C,a)^{\mathcal{I}}$

Figure 2: Two individuals, their most specific concepts (dotted), and the mimics of a concept C w.r.t. the individuals (dashed).

Intuitively, a mimic of C w.r.t. a is a concept that is as similar to C as possible, while still having a as an instance. As for relaxed instances, the mimic strongly depends on the similarity measure chosen. Figure 2 depicts the idea of mimics. In the figure, a and b are two named individuals. The former is an instance of C while the latter is not. The dotted lines depict their most specific concepts. Since a is an instance of C, C is also a mimic of C w.r.t. a: $C \sim C = 1$. The dashed line depicts a mimic of C w.r.t. b. Notice that this mimic must contain the msc of b, but need not be a subsumer of C.

We must point out that the mimic of C w.r.t. an individual a need not be unique, even modulo concept equivalence. For example, let $\mathcal{K}$ be a knowledge base consisting of the empty TBox $\mathcal{T}$ and the ABox $\mathcal{A} = \{A \sqcap B(a)\}$, and $\sim$ be a similarity measure with $A \sim C = 0.5$, $B \sim C = 0.5$ and $(A \sqcap B) \sim C = \max\{A \sim C, B \sim C\} = 0.5$. Then A, B, and $A \sqcap B$, are all mimics of C w.r.t. a, as they all have a similarity value of 0.5 to C. In fact, there can be infinitely many such mimics for a given concept C and individual a. As we will see, it suffices to compute one of them.

Using mimics, we can compute the relaxed instances of a concept. The idea is to compute, for each individual a appearing in the knowledge base $\mathcal{K}$, the mimic of C w.r.t. a. If this mimic has similarity at least t with C, then a is a relaxed instance of C; otherwise, it cannot be a relaxed instance, as no concept can have a greater similarity degree with C while still containing a. This is formalized in the following proposition. The proof is a simple consequence of the arguments given above.

Proposition 4. *Let $\mathcal{K}$ be a knowledge base, a be an individual occurring in $\mathcal{K}$, C be a concept description, $\sim$ be a similarity measure and $t \in [0, 1]$. Then $a \in \text{Relax}_t^{\sim}(C)$ iff there is a mimic D of C w.r.t. individual a such that $C \sim D \geq t$.*

In the next section we will study the problem of computing a mimic for a given concept C w.r.t. an individual a. Since all mimics must have the same degree of similarity w.r.t. C, a simple similarity computation provides us with a decision whether a is a relaxed instance of C or not, up to degree t. As computing a mimic may be an expensive task, we also provide an optimization criterion: if a mimic D of C w.r.t. a is similar to C to degree at least t, then all certain instances of D must also be relaxed instances of C, and hence there is no need of computing their corresponding mimics.

4 Computing Mimics in $\mathcal{EL}$

In general there are infinitely many concepts, for which an individual a is an instance of, and thus enumerating them and computing the similarity to C to find the mimic is not a feasible option. However, under some circumstances we can limit the number of concepts that need to be tested in order to find a mimic.

Recall that the notion of a mimic combines a property that is based on the semantics (it must have a as an instance) and a syntactic property (it must be similar to C). The semantic property gives us a starting point on how to find a mimic. A mimic D of C w.r.t. a must always have a as an instance, and hence, by definition of the msc, $\text{msc}(a) \sqsubseteq_{\mathcal{T}} D$ holds. For equivalence invariant similarity measures the idea is to use the $\text{msc}(a)$ as a lower bound for the mimic guaranteeing the semantic property, and to only consider concept descriptions that can be obtained from syntactic manipulations of $\text{msc}(a)$ that result in a generalized concept, i.e., by removing some concept names or existential restrictions.

Definition 5 (generalized concept)**.** Let C be a concept description of the form

$$C = \textstyle\bigsqcap_{i \in I} A_i \sqcap \bigsqcap_{j \in J} \exists r_j.E_j,$$

with $A_i \in N_C$ for all $i \in I$, and $r_j \in N_R$, E_j is a concept description for all $j \in J$. Then a concept description D is a *generalized concept* of C iff it has the form

$$D = \textstyle\bigsqcap_{i \in I'} A_i \sqcap \bigsqcap_{j \in J'} \exists r_j.E'_j$$

with $I' \subseteq I$, $J' \subseteq J$ and E'_j is a generalized concept of E_j for $j \in J'$.

This idea, however, only works if the msc is given in a particular syntactic form. It needs to be fully expanded.

Definition 6 (fully expanded concept)**.** Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox. A concept description C is *fully expanded* w.r.t. $\mathcal{T}$ iff for all concept definitions $D = E \in \mathcal{T}$ with $C \sqsubseteq_{\mathcal{T}} D$ we have that E is a generalized concept of C.

The idea is that C contains all its subsumers explicitly as sub-concept descriptions. Now, we can show that the mimic of C w.r.t. a must be a generalized concept of the fully expanded most specific concept of a.

Lemma 7. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{EL}$-knowledge base, a be an individual from $\mathcal{A}$, C be an $\mathcal{EL}$-concept description, and $\sim$ be an equivalence invariant similarity measure. Let further $E = \text{msc}(a)$ be the fully expanded most specific concept of a. Then there is a mimic $D = \mathfrak{M}(C, a)$ of C w.r.t. a and $\mathcal{K}$ that is a generalized concept of E.*

Proof. We show that any concept F which has a as an instance must be equivalent to a generalized concept of the fully expanded msc. Since the mimic of C w.r.t. a has a as an instance and $\sim$ is equivalence invariant, the lemma follows.

Let F be a concept description with $a^{\mathcal{I}} \in F^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$. Then $E \sqsubseteq_{\mathcal{K}} F$ by definition of the msc. Since E is fully expanded and contains all its subsumers explicitly, any part of the concept description F must also be part of the concept description E. Thus F is a generalized concept of E. □

In general, the msc may contain a chain of infinitely nested existential restrictions for cyclic ABoxes, and hence describing it as a concept would require infinite size. Then there are still infinitely many generalized concepts (of finite size) that need to be checked to find a mimic. This means that Lemma 7 does not always provide a solution to the problem. However, the query concept C (in $\sim$-normal form) has always a finite role-depth, and most structural similarity measures used in practice compute the similarity recursively between concepts at the same role-depth. Therefore, for these similarity measures, it is possible to limit the role-depth of the most specific concept and still get the same result.

Definition 8. Let $\mathcal{K}$ be an $\mathcal{EL}$-KB. By $\text{rd}(C)$ we denote the *role-depth* of a concept C, i.e. the maximal number of nested quantifiers.

The $\mathcal{EL}$-concept description C is the *role-depth bounded most specific concept* (denoted k-$\text{msc}(a)$) of an individual a w.r.t. $\mathcal{K}$ and the role-depth bound k iff

- $\text{rd}(C) \leq k$,
- $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$, and
- for all $\mathcal{EL}$-concepts $D \in \mathcal{C}(\mathcal{L})$ with $\text{rd}(D) \leq k$ and all $a^{\mathcal{I}} \in D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$ it holds that $C \sqsubseteq_{\mathcal{T}} D$.

The role-depth bounded msc is a commonly used approximation of the msc, since it always exists and is unique. An algorithm to compute the k-msc in the $\mathcal{EL}$-family, even w.r.t. general TBoxes, has been introduced in [Peñaloza and Turhan, 2011] and [Ecke *et al.*, 2013]. Using this, we can now show that for structural similarity measures we can find the mimic always as a generalized concept of the role-depth bounded msc.

Lemma 9. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{EL}$-knowledge base, a be an individual from $\mathcal{A}$, C be an $\mathcal{EL}$-concept description in $\sim$-normal form, and $\sim$ be a structural, equivalence invariant similarity measure with the following property:*

$$X \sim \textstyle\bigsqcap_{i \in I} A_i \;\geq\; X \sqcap \exists r.B \sim \bigsqcap_{i \in I} A_i. \qquad (1)$$

Let further $k = \text{rd}(C)$ and $E = k$-$\text{msc}(a)$ be the fully expanded role-depth bounded most specific concept of a. Then there is a mimic $D = \mathfrak{M}(C, a)$ of C w.r.t. a that is a generalized concept of E.

Proof. By Lemma 7 we know that there exists a mimic F of C w.r.t. a that is a generalized concept of the (possibly infinite) $\text{msc}(a)$. Since E is the fully expanded k-msc of a, F must also be a generalized concept of E up to role-depth k (but of course, it may contain additional existential restrictions which increase the role-depth of F). We show by induction on k, that there is a generalized concept F' of E with $F' \sim C \geq F \sim C$. This will imply that F' is a mimic of C w.r.t. a, which proves the lemma.

For the case $k = 0$, $C = \bigsqcap_{i \in I} A_i$ and $E = \bigsqcap_{j \in J} B_j$ are conjunctions of concept names and since F a generalized concept of E up to role-depth $k = 0$, we know that F is of the form $F = \bigsqcap_{j \in J'} B_j \sqcap \bigsqcap_{h \in H} \exists r_h.F_h$ with $J' \subseteq J$. But then property (1) yields for $F' = \bigsqcap_{j \in J'} B_j$:

$$F' \sim C \;\geq\; F' \sqcap \textstyle\bigsqcap_{h \in H} \exists r_h.F_h \sim C \;=\; F \sim C.$$

```
Procedure relaxed-instance?$(a, C, \mathcal{K}, \sim, t)$
Input: $a$: individual in $\mathcal{K}$; $C$: $\mathcal{EL}$-concept description;
       $\mathcal{K}$: $\mathcal{EL}$-knowledge base; $\sim$: similarity measure;
       $t$: similarity degree;
Output: whether $a \in_t^{\sim} C$ w.r.t. $\mathcal{K}$
$k := \text{rd}(C)$
$E := k\text{-msc}(a)$ w.r.t. $\mathcal{K}$
guess a generalized concept $F$ of $E$
if $F \sim C \geq t$ then
    return true
else
    return false
```

Figure 3: Computation algorithm for relaxed instances in $\mathcal{EL}$.

For the case $k > 0$, $C = \bigsqcap_{i \in I} A_i \sqcap \bigsqcap_{h \in H} \exists s_h.C_h$ and $E = \bigsqcap_{j \in J} B_j \sqcap \bigsqcap_{l \in L} \exists r_l.E_l$ are conjunctions of concept names and existential restrictions with $\text{rd}(C_h), \text{rd}(E_l) \leq k-1$ for $h \in H$, $l \in L$. Once again, since F is a generalized concept of E up to role-depth k, it must be of the form $F = \bigsqcap_{j \in J'} B_j \sqcap \bigsqcap_{l \in L'} \exists r_l.F_l$ with $J' \subseteq J$, $L' \subseteq L$ and each F_l is a generalized concept of E_l up to role-depth $k-1$. But then, the induction hypothesis yields for each $h \in H$ and $l \in L'$ that $F'_l \sim C_h \geq F_l \sim C_h$ for generalized concepts F'_l of E_l. Then also $F' = \bigsqcap_{j \in J'} B_j \sqcap \bigsqcap_{l \in L'} \exists r_l.F'_l$ is a generalized concept of E and since the similarity measure $\sim$ is structural, this yields: $F' \sim C \geq F \sim C$. □

We have now identified some constraints on the similarity measure such that we can always find the mimic of C w.r.t. a from a finite set of concept descriptions: the generalized concepts of the fully expanded role-depth bounded msc of the individual a.

Instead of computing the mimic $D = \mathfrak{M}(C, a)$ of C w.r.t. a and testing whether the similarity between the C and D is at least t, it is enough to find *any* concept D' with a as an instance and $C \sim D' \geq t$ to show that a is a relaxed instance of C; Such a non-deterministic algorithm that, given an $\mathcal{EL}$-KB $\mathcal{K}$, an individual a, an $\mathcal{EL}$-concept description C, a similarity measure $\sim$, and a similarity degree t, computes whether a is a relaxed instance of C w.r.t. $\sim$ and t, is given in Figure 3. The algorithm works by computing the k-msc of a with $k = \text{rd}(C)$ and then guessing a generalized concept F of E with similarity $F \sim C \geq t$, if such a concept exists.

Corollary 10. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{EL}$-knowledge base, C be an $\mathcal{EL}$ concept in $\sim$-normal form, a be an individual in $\mathcal{K}$, $\sim$ be a structural equivalence invariant similarity measure fulfilling Property 1 from Lemma 9 and $t \in [0, 1]$. Then* relaxed-instance?$(a, C, \mathcal{K}, \sim, t)$ *computes whether $a \in_t^{\sim} C$ w.r.t. $\mathcal{K}$.*

Proof. Lemma 9 shows that a mimic of C w.r.t. a is a generalized concept of $E = k\text{-msc}(a)$ for $k = \text{rd}(C)$. Thus, if the algorithm returns false, we know that no generalized concept F exists with $C \sim F \geq t$, and in particular also the mimic of C w.r.t. a must have a similarity of less than t to C. Thus no concept that has a as an instance is similar enough to C and thus $a \notin_t^{\sim} C$. If the algorithm returns true, the guessed concept F shows $a \in_t^{\sim} C$, since a is an instance of F and $F \sim C \geq t$. □

Guessing a generalized concept F of a concept description E can be done in time linear to size $\|E\|$ of E by recursively guessing for each concept name and each existential restriction in E whether they should occur in F or not. However, the size of $E = k\text{-msc}(a)$ can be exponential in k and polynomial in $\|\mathcal{K}\|$ [Peñaloza and Turhan, 2011]. Since $k = \text{rd}(C)$ is bounded linearly by $\|C\|$, the algorithm runs in NEXP-time (provided that $\sim$ can be computed in NEXP-time). However, the algorithm runs in NP-time in $\|\mathcal{K}\|$ (provided that $\sim$ can be computed in NP), and since C is an input concept, its role-depth can be assumed to be rather low. Hence, we conjecture that the exponential blow-up of the msc usually plays only a minor role in practical applications.

To obtain a deterministic algorithm, the mimic of C w.r.t. a can be computed by enumerating all generalized concepts of $k\text{-msc}(a)$ and taking one with the maximal similarity to C. Of course, there are a few optimizations possible: if the individual a belongs to C, we can directly return true, since the mimic will always be C itself. If we find a generalized concept F with $C \sim F \geq t$, we can stop to search for even more similar concepts and return true. And finally, if we find a mimic D for an individual a with $C \sim D \geq t$, we know that all other instances of D besides a will be relaxed instances of C as well, without needing to compute *their* mimics.

5 Conclusions

In this paper we have studied a new inference service for description logics, which consists in computing the relaxed instances of a given query concept C w.r.t. a similarity measure $\sim$ and a similarity degree t. This problem is relevant to the field of artificial intelligence in general, and to knowledge representation and reasoning in particular, as it provides a formal and unambiguous method for computing answers for a relaxed notion of instance query. Thus it is useful for ontology-based applications that need to obtain answers that fit the query criteria only to a certain degree.

The inference has two main degrees of freedom: in the choice of the similarity measure, and in the degree of relaxation of the concept. The similarity degree t allows the user to tune how strict or relaxed the answers provided are: a degree closer to 1 will yield only a few additional individuals that do not belong to C, while relaxing to a level closer to 0 yields almost all individuals in the ontology as relaxed instances. The similarity measure provides also criteria on how the relaxed instances are obtained. Intuitively, different similarity measures yield different weights on specific criteria. For example, one could require that small changes inside existential restrictions produce a high level of dissimilarity.

As a step for computing the relaxed instances of a concept C, we introduced the problem of finding a mimic of the query concept C w.r.t. a given individual a. Such a mimic is a concept D that contains a as instance, and has the highest similarity possible to C; i.e., it is a concept that tries to imitate C while containing a. Computing mimics w.r.t. all individuals appearing in an ontology provides a method for finding the relaxed instances of C.

The problem of finding a mimic is non-trivial. We have provided an algorithm capable of finding such a mimic, based on the msc of an individual a for certain structural similarity measures. While this computation is expensive, some obvious optimizations can be used to reduce the number of times these mimics are constructed.

As future work, we plan to expand on the two main inference problems described in this paper. First, we intend to improve the algorithms that compute the mimics. On the one hand, we will try to find one such mimic efficiently. On the other, it would also be beneficial to compute the most general mimic, if it exists; this concept would have the most possible instances, and hence would be useful as an optimization approach. Second, we will try to find tight complexity bounds on the problems of computing relaxed instances and finding mimics for a given concept. Third, we plan to obtain a better understanding on the properties of similarity measures that can impact (positively or negatively) on the complexity and run-time of solving these problems. As we have mentioned before, both inferences depend strongly on the similarity measure chosen. However, we do not know precisely which measures would allow for better results, be it in terms of execution time, or in terms of precision and fine-grained tuning.

References

[Alvarez and Yan, 2011] M. A. Alvarez and C. Yan. A graph-based semantic similarity measure for the gene ontology. *J. Bioinformatics and Computational Biology*, 9(6):681–695, 2011.

[Baader *et al.*, 2003] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[Borgida *et al.*, 2005] A. Borgida, T. Walsh, and H. Hirsh. Towards measuring similarity in description logics. In *Proc. of the 2005 Description Logic Workshop (DL 2005)*, volume 147 of *CEUR Workshop Proceedings*, 2005.

[Borgwardt and Peñaloza, 2012] S. Borgwardt and R. Peñaloza. Undecidability of fuzzy description logics. In *Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-12)*, pages 232–242. AAAI Press, 2012.

[Borgwardt *et al.*, 2012] S. Borgwardt, F. Distel, and R. Peñaloza. How fuzzy is my fuzzy description logic? volume 7364 of *Lecture Notes In Artificial Intelligence*, pages 82–96. Springer-Verlag, 2012.

[Cerami and Straccia, 2013] M. Cerami and U. Straccia. On the (un)decidability of fuzzy description logics under lukasiewicz t-norm. *Inf. Sci.*, 227:1–21, 2013.

[d'Amato *et al.*, 2005] C. d'Amato, N. Fanizzi, and F. Esposito. A semantic similarity measure for expressive description logics. In *Proc. of Convegno Italiano di Logica Computazionale, CILC05*, 2005.

[Ecke *et al.*, 2013] A. Ecke, R. Peñaloza, and A.-Y. Turhan. Computing role-depth bounded generalizations in the description logic $\mathcal{ELOR}$. In *Proceedings of the 36th German Conference on Artificial Intelligence (KI 2013)*, volume 8077 of *Lecture Notes in Artificial Intelligence*, Koblenz, Germany, 2013. To appear.

[Gene Ontology Consortium, 2000] The Gene Ontology Consortium. Gene Ontology: Tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[Haarslev *et al.*, 2012] V. Haarslev, K. Hidde, R. Möller, and M. Wessel. The RacerPro knowledge representation and reasoning system. *Semantic Web Journal*, 3(3):267–277, 2012.

[Kazakov *et al.*, 2012] Y. Kazakov, M. Krötzsch, and F. Simančík. ELK reasoner: Architecture and evaluation. In *Proceedings of the OWL Reasoner Evaluation Workshop (ORE'12)*, volume 858 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

[Lehmann and Turhan, 2012] K. Lehmann and A.-Y. Turhan. A framework for semantic-based similarity measures for $\mathcal{ELH}$-concepts. In *Proceedings of the 13th European Conference on Logics in Artificial Intelligence*, Lecture Notes in Artificial Intelligence, pages 307–319. Springer Verlag, 2012.

[Lord *et al.*, 2003] P. W. Lord, R. D. Stevens, A. Brass, and C. A. Goble. Investigating semantic similarity measures across the gene ontology: The relationship between sequence and annotation. *Bioinformatics*, 19(10):1275–1283, 2003.

[Lutz *et al.*, 2003] C. Lutz, F. Wolter, and M. Zakharyaschev. Reasoning about concepts and similarity. In *Proceedings of the 2003 International Workshop on Description Logics (DL2003)*, CEUR-WS, 2003.

[Mistry and Pavlidis, 2008] M. Mistry and P. Pavlidis. Gene ontology term overlap as a measure of gene functional similarity. *BMC Bioinformatics*, 9, 2008.

[Motik *et al.*, 2009] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, and C. Lutz. OWL 2 web ontology language profiles. W3C Recommendation, 27 October 2009. `http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/`.

[Nebel, 1990] B. Nebel. *Reasoning and revision in hybrid representation systems*. Springer-Verlag New York, Inc., New York, NY, USA, 1990.

[Peñaloza and Turhan, 2011] R. Peñaloza and A.-Y. Turhan. A practical approach for computing generalization inferences in $\mathcal{EL}$. In *Proceedings of the 8th European Semantic Web Conference (ESWC'11)*, Lecture Notes in Computer Science. Springer-Verlag, 2011.

[Schlicker *et al.*, 2006] A. Schlicker, F. S. Domingues, J. Rahnenführer, and T. Lengauer. A new measure for functional similarity of gene products based on gene ontology. *BMC Bioinformatics*, 7:302, 2006.

[Tsarkov and Horrocks, 2006] D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In *Proc. of the 3rd Int. Joint Conf. on Automated Reasoning (IJCAR-06)*, 2006. FaCT++ download page: `http://owl.man.ac.uk/factplusplus/`.