Similarity-based Relaxed Instance Queries

Andreas Ecke[a,*], Rafael Peñaloza[a,b], Anni-Yasmin Turhan[a]

[a] *Institute for Theoretical Computer Science, Technische Universität Dresden*
[b] *Center for Advancing Electronics Dresden*

Abstract

In Description Logics (DL) knowledge bases (KBs), information is typically captured by clear-cut concepts. For many practical applications querying the KB by crisp concepts is too restrictive; a user might be willing to lose some precision in the query, in exchange of a larger selection of answers. Similarity measures can offer a controlled way of gradually relaxing a query concept within a user-specified limit.

In this paper we formalize the task of instance query answering for DL KBs using concepts relaxed by concept similarity measures (CSMs). We investigate computation algorithms for this task in the DL $\mathcal{EL}$, their complexity and properties for the CSMs employed regarding whether unfoldable or general TBoxes are used. For the case of general TBoxes we define a family of CSMs that take the full TBox information into account, when assessing the similarity of concepts.

Keywords: Description Logics, instance queries, concept similarity measures

1. Introduction

Description Logics (DLs) are a family of knowledge representation formalisms that have unambiguous logic-based semantics. Each particular DL is characterized by a set of concept constructors, which allow to build complex concepts. Intuitively, *concepts* 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 to build more complex concepts with the help of a class of concept constructors. The terminological knowledge of an application domain is stored in the *TBox*, that expresses the relationships between concepts. Facts from the application domain and relations between them are represented by assertions about *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 concepts and *instance query answering*, where for a given concept all individuals from an ABox that are instances of the query concept are computed. These reasoning services are implemented in highly optimized reasoning systems, see for example [1–4].

*Corresponding author

Email addresses: ecke@tcs.inf.tu-dresden.de (Andreas Ecke), penaloza@tcs.inf.tu-dresden.de (Rafael Peñaloza), turhan@tcs.inf.tu-dresden.de (Anni-Yasmin Turhan)

DLs of varying expressivity are the underlying logics for the W3C standardized ontology language OWL 2 and its profiles [5]. This standardization 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 more relaxed manner. For instance, 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 requirements for the desired service. These conditions are represented by a complex concept. For such a concept the OWL ABox that contains the individual services is searched for a service matching the specified request by performing instance query answering. In cases where an exact match with the provided requirements is not possible, a 'feasible' alternative should be retrieved from the ABox containing the services to be able to offer an alternative. Essentially, for a given query concept, the system should retrieve all those individuals of the ABox that fulfill the main requirements, while allowing a relaxation of some of the less crutial requirements.

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 [6–8] 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. Fuzzy concepts would relax the initial concept in an unspecific and uniform way. In contrast, relaxed instance query answering should allow to

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

The selective and gradual relaxation of the answer sets returned by instance query answering is achieved by the use of concept similarity measures. A *concept similarity measure* (CSM) yields, for a pair of concepts, a value from the interval $[0, 1]$—indicating how similar the concepts are. To answer a relaxed instance query is to compute for a given concept C, a CSM $\sim$ and a degree t between 0 and 1, a set of concepts such that each of these concepts is similar to C by a degree of at least t, if measured by the CSM $\sim$, and then finding all their instances.

Concept similarity measures are widely used in ontology-based applications. In the biomedical field, for example the Gene ontology [9], they are employed to discover functional similarities of genes (see e.g. [10, 11]). Furthermore, CSMs are used in ontology alignment algorithms [12]. For DLs there exists a whole range of CSMs , which could be employed for the task of answering relaxed instance queries [13–16]. In particular the CSMs generated by the framework described in [15] 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, the measures generated by this framework naturally allow users to select important features

of the query concept and which aspect of the query concept to relax.

We investigate algorithms for computing answers to instance queries relaxed by CSMs for the light-weight DL $\mathcal{EL}$. Our choice for the DL $\mathcal{EL}$ is motivated by the fact that reasoning in $\mathcal{EL}$ has good computational properties—most standard reasoning problems can be solved in polynomial time [17]. Large, well-known bio-medical ontologies such as the Gene Ontology [9] or SNOMED [18] are written in (polynomial extensions of) $\mathcal{EL}$. Furthermore, $\mathcal{EL}$ is a fragment of the DL that corresponds to the OWL 2 EL profile, which is part of the W3C standard for an ontology language for the Semantic Web [5] and thus widely used in practice.

The contributions presented in this paper are the following: after the formal definition of the reasoning task of interest, namely answering relaxed instance queries, we investigate reasoning algorithms for it in two settings:

1. Computing relaxed instances w.r.t. $\mathcal{EL}$-terminologies
   This setting has initially been investigated by us in [19]. Terminologies are a simple kind of TBox that allows to treat the TBox information in a preprocessing step. By far most CSMs are defined for this kind of TBoxes. We identify formal properties of CSMs that allow to compute relaxed instances and devise an algorithm to compute relaxed instances for unfoldable TBoxes w.r.t. CSMs that enjoy these properties.

2. Computing relaxed instances w.r.t. general $\mathcal{EL}$-TBoxes
   This setting was recently explored by us in [20]. To the best of our knowledge there are hardly any CSMs for DLs defined in the literature that take the whole information of general TBoxes into account. In [21] a CSM in regard of general TBoxes is defined, but it uses only the subsumption information between named concepts. We define a family of CSMs $\sim_c$ that is founded on (a similarity measure for) the canonical interpretations of general $\mathcal{EL}$-TBoxes. We show that members of this family of CSMs have certain formal properties. We give a computation algorithm for relaxed instances w.r.t. general TBoxes that rely on the shown formal properties for the CSM $\sim_c$.

The paper is structured as follows: in the next section we give the preliminaries on Description Logics, in particular $\mathcal{EL}$, and on concept similarity measures. In Section 3 we describe our approach for relaxed instance queries, and define this reasoning task formally. In Section 4 we develop a computation algorithm for answering relaxed instance queries for concepts defined in $\mathcal{EL}$-terminologies and give an upper bound for its complexity. Then, in Section 5, we turn to concepts defined w.r.t. general $\mathcal{EL}$-TBoxes and define the CSM $\sim_c$ based on canonical interpretations of general $\mathcal{EL}$-TBoxes, show some of its formal properties. We devise a computation algorithm for computing relaxed instances w.r.t. general $\mathcal{EL}$-TBoxes, where these properties are employed. As usual we end with some conclusions and considerations for future work. Due to their length, the proofs for Section 5 appear in the appendix.

2. Preliminaries

In this section we first give a brief introduction to the main notions of Description Logics, knowledge bases, and different inference problems defined for them. Afterwards, we introduce concept similarity measures and their relevant properties.

2.1. The Description Logic $\mathcal{EL}$

For the scope of this paper, we consider only the description logic $\mathcal{EL}$, which we briefly introduce next. For a broader introduction to DLs we refer the reader to [22, 23].

The DL $\mathcal{EL}$ is a light-weight DL, which has limited expressivity, but nice computational properties as it allows for reasoning in polynomial time [17].

Definition 2.1 ($\mathcal{EL}$-concepts)**.** Let N_C and N_R be countably infinite disjoint sets of concept names and role names, respectively. The set of *$\mathcal{EL}$-concepts* is the smallest set such that

- all concept names $A \in N_C$ are $\mathcal{EL}$-concepts;
- the top-concept $\top$ is an $\mathcal{EL}$-concept;
- if C and D are $\mathcal{EL}$-concepts, then $C \sqcap D$ is also an $\mathcal{EL}$-concept;
- if C is an $\mathcal{EL}$-concept and $r \in N_R$, then $\exists r.C$ is also an $\mathcal{EL}$-concept.

The set of all $\mathcal{EL}$-concepts is denoted by $\mathfrak{C}(\mathcal{EL})$.

The semantics of this logic 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 and subsets of $\Delta^{\mathcal{I}}$ to concept names. The interpretation function is recursively extended to (complex) $\mathcal{EL}$-concepts as shown in the upper part of Table 1. We denote the *set of all interpretations* as $\mathfrak{I}$.

A *pointed interpretation* $p = (\mathcal{I}, d)$ consists of an interpretation $\mathcal{I} \in \mathfrak{I}$, and an element $d \in \Delta^{\mathcal{I}}$. $\mathfrak{P}$ is the *set of all pointed interpretations*, i.e., $\mathfrak{P} := \{(\mathcal{I}, d) \mid \mathcal{I} \in \mathfrak{I}, d \in \Delta^{\mathcal{I}}\}$. Given a pointed interpretation $p = (\mathcal{I}, d)$, the set of all $\mathcal{EL}$-concepts that have d as an instance in $\mathcal{I}$ is the *concept set* of a pointed interpretation $\mathfrak{C}(p) = \{C \in \mathfrak{C}(\mathcal{EL}) \mid d \in C^{\mathcal{I}}\}$. When considering the complexity of reasoning for concepts, the size or the role-depth of a concept are commonly taken as input size. The *size* $|C|$ of a given $\mathcal{EL}$-concept C is defined as:

$$|C| := \begin{cases} 1 & \text{if } C \in N_C \cup \{\top\} \\ 1 + |D| & \text{if } C = \exists r.D \\ |C_1| + |C_2| & \text{if } C = C_1 \sqcap C_2 \end{cases}$$

The *role-depth* $\mathrm{rd}(C)$ of a given $\mathcal{EL}$-concept C is defined as:

$$\mathrm{rd}(C) := \begin{cases} 0 & \text{if } C \in N_C \cup \{\top\} \\ 1 + \mathrm{rd}(D) & \text{if } C = \exists r.D \\ max\{\mathrm{rd}(C_1), \mathrm{rd}(C_2)\} & \text{if } C = C_1 \sqcap C_2 \end{cases}$$

In DLs, one is not only interested in expressing concepts, but in representing the knowledge about them. This knowledge is encoded using different kinds of *axioms*. Concept axioms, displayed in the middle part of Table 1, express relationships between concepts. A *concept definition* assigns a concept name to a (complex) concept and general concept axioms (GCIs) state that one concept is implied by another. An *$\mathcal{EL}$-TBox* $\mathcal{T}$ is a finite set of such concept axioms. An *unfoldable TBox*, also called a *terminology*, is a set of concept definitions such that

	Syntax	Semantics
Concepts		
concept name	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
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	$\exists r.C$	$(\exists r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \exists e.(d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$
TBox axioms		
concept definition	$A \equiv C$	$A^{\mathcal{I}} = C^{\mathcal{I}}$
general concept axiom	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
ABox assertions		
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 $\mathcal{EL}$.

- 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, i.e., no concept name is defined in direct or indirect reference to itself.

Note that concept inclusions of the form $A \sqsubseteq C$ can be incorporated into unfoldable TBoxes as a concept definition $A \equiv C \sqcap X$ by simply adding a new concept name X, so long as the left-hand side is only a concept name and the resulting TBox is still acyclic. Any model of the original TBox can then be converted into a model of the new unfoldable TBox by simply giving X an appropriate interpretation and vice versa, so this transformation preserves all standard reasoning tasks.

TBoxes store the intensional knowledge of the categories from the application domain. In a terminology, the (complex) concepts on the right-hand sides are abbreviated by the concept names appearing on the left-hand side of the definition.

Example 2.2. Terminologies can be used to encode knowledge from the service matching domain as follows:

$$
\begin{aligned}
\mathcal{T}_{ex} = \{ \quad \mathsf{Server} &\equiv \mathsf{Computer} \sqcap \exists \mathsf{provides.Service} \sqcap \\
&\quad \exists \mathsf{hasLatency.Amount} \sqcap \exists \mathsf{hasLoad.Amount}, \\
\mathsf{VideoStreamService} &\sqsubseteq \mathsf{Service} \sqcap \exists \mathsf{hasQuality.Amount} \sqcap \\
&\quad \exists \mathsf{hasFeature.VideoStreamFeature}, \\
\mathsf{Seekable} &\sqsubseteq \mathsf{VideoStreamFeature}, \\
\mathsf{Low} &\sqsubseteq \mathsf{Amount}, \\
\mathsf{Medium} &\sqsubseteq \mathsf{Amount}, \\
\mathsf{High} &\sqsubseteq \mathsf{Amount} \ \}
\end{aligned}
$$

Besides clients, also services can use other services. For example, one can define restricted services that only work for registered users by relying on an external login-service. Similarly, a

login-service would be a special type of service that provides the credentials of the users to those restricted services.

$$\mathcal{T}_{ex2} = \mathcal{T}_{ex} \cup \{\ \mathsf{RestrictedService} \equiv \mathsf{Service} \sqcap \exists \mathsf{dependsOn}.\mathsf{LoginService},$$
$$\mathsf{LoginService} \equiv \mathsf{Service} \sqcap \exists \mathsf{providesCredentialsTo}.\mathsf{RestrictedService}\ \}$$

This TBox is no longer unfoldable, since it contains a cycle.

The semantics of interpretations is extended to TBox axioms as shown in Table 1. More precisely, the interpretation $\mathcal{I}$ satisfies the concept definition $A \equiv C$ iff $A^{\mathcal{I}} = C^{\mathcal{I}}$, and satisfies the GCI $C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. The interpretation $\mathcal{I}$ is a *model* of the TBox $\mathcal{T}$, if it satisfies all concept axioms in $\mathcal{T}$.

Knowledge about facts is expressed using individuals and assertional axioms. We consider a countably infinite set N_I of *individual names*, which is disjoint with both N_C and N_R. The notion of interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is extended such that the interpretation function $\cdot^{\mathcal{I}}$ additionally maps each individual name $a \in N_I$ to an element of $\Delta^{\mathcal{I}}$. A *concept assertion* is a statement of the form $C(a)$ where $a \in N_I$ and C is a concept. A *role assertion* is a statement of the form $r(a, b)$ where $r \in N_R$ and $a, b \in N_I$. The semantics of these assertions are displayed at the bottom of Table 1. Extensional knowledge about facts is collected in an ABox,which is a finite set of concept and role assertions. An interpretation $\mathcal{I}$ is a *model* of the ABox $\mathcal{A}$, if it satisfies all assertions in $\mathcal{A}$. An *$\mathcal{EL}$-knowledge base* is a pair $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consisting of an *$\mathcal{EL}$-TBox* $\mathcal{T}$ and an *$\mathcal{EL}$-ABox* $\mathcal{A}$. The interpretation $\mathcal{I}$ is a model of a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if it is a model of both $\mathcal{T}$ and $\mathcal{A}$.

Typical reasoning problems in DLs are to decide consistency of a KB, subsumption between concepts, and checking whether an individual is an instance of a concept. A KB is *consistent* if it has a model. Since $\mathcal{EL}$ is not capable of expressing contradictions, testing consistency is trivial in this logic. *Concept subsumption* is the problem of deciding whether a concept C is subsumed by a concept D w.r.t. a TBox $\mathcal{T}$ (denoted by $C \sqsubseteq_{\mathcal{T}} D$), i.e. whether $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all models $\mathcal{I}$ of $\mathcal{T}$. Similarly, two concepts C and D are *equivalent* w.r.t. $\mathcal{T}$ (denoted as $C \equiv_{\mathcal{T}} D$), iff $C \sqsubseteq_{\mathcal{T}} D$ and $D \sqsubseteq_{\mathcal{T}} C$. An individual a is an *instance of* a concept C w.r.t. a KB $\mathcal{K}$ (denoted by $\mathcal{K} \models C(a)$) iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{K}$. Given a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ and a concept C, an *instance query* returns all individuals from $\mathcal{A}$ that are instances of C. In this paper we are interested in a generalization of instance queries. Rather than finding all instances of a given concept C, we aim to compute the instances of all concepts D that are *sufficiently similar* to C; to achieve this, we relax the query concept.

It is known that all these standard inferences can be characterized by means of simulations between interpretations [24]. These simulations basically outline the indistinguishable elements in the domains of two interpretations.

Definition 2.3 (simulation)**.** Let $\mathcal{I}$ and $\mathcal{J}$ be interpretations. A relation $S \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}$ is a *simulation* between $\mathcal{I}$ and $\mathcal{J}$, if the following two conditions hold:

1. For all $(d, e) \in S$ and $A \in N_C$, if $d \in A^{\mathcal{I}}$ then $e \in A^{\mathcal{J}}$.
2. For all $(d, e) \in S$, $r \in N_R$ and $(d, d') \in r^{\mathcal{I}}$, there is an $(e, e') \in r^{\mathcal{J}}$ with $(d', e') \in S$.

Given two pointed interpretations $p = (\mathcal{I}, d)$ and $q = (\mathcal{J}, e)$, we say that

- *p simulates q* (denoted by $p \lesssim q$), if there exists a simulation $S \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}$ between $\mathcal{I}$ and $\mathcal{J}$ with $(d, e) \in S$, and
- p and q are *equisimilar* (denoted by $p \simeq q$), if $p \lesssim q$ and $q \lesssim p$.

There is a strong connection between simulations between pointed interpretations and their concept sets, as described in the following theorem.

Theorem 2.4 (By Lutz and Wolter [24])**.** *Let $p, q \in \mathfrak{P}$. Then:*

1. *$p \lesssim q$ iff $\mathfrak{C}(p) \subseteq \mathfrak{C}(q)$, and*
2. *$p \simeq q$ iff $\mathfrak{C}(p) = \mathfrak{C}(q)$.*

For the DL $\mathcal{EL}$, most reasoning procedures rely on the fact that canonical models can be built, from which it is possible to read entailments directly. Before we can formally define these canonical models, we need to introduce some notation. If X is a concept description, TBox, ABox, or KB, then:

- $\mathsf{Sig}(X)$ denotes the signature of X; that is, the set of concept, role, and individual names appearing in X, and
- $\mathsf{sub}(X)$ is the set of all sub-concepts of concepts occurring in X.

Definition 2.5. (canonical models) Let C be an $\mathcal{EL}$-concept and $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ an $\mathcal{EL}$-KB. The *canonical model $\mathcal{I}_{C,\mathcal{T}} = (\Delta^{\mathcal{I}_{C,\mathcal{T}}}, \cdot^{\mathcal{I}_{C,\mathcal{T}}})$ of C* w.r.t. the TBox $\mathcal{T}$ is defined as follows:

- $\Delta^{\mathcal{I}_{C,\mathcal{T}}} = \{d_C\} \cup \{d_D \mid \exists r.D \in \mathsf{sub}(C) \cup \mathsf{sub}(\mathcal{T})\}$
- $A^{\mathcal{I}_{C,\mathcal{T}}} = \{d_D \mid D \sqsubseteq_{\mathcal{T}} A\}$, for all concept names A, and
- $r^{\mathcal{I}_{C,\mathcal{T}}} = \{(d_D, d_E) \mid D \sqsubseteq_{\mathcal{T}} \exists r.E\}$ for all role names r.

The *canonical model $\mathcal{I}_{\mathcal{K}} = (\Delta^{\mathcal{I}_{\mathcal{K}}}, \cdot^{\mathcal{I}_{\mathcal{K}}})$ of the KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$* is defined as follows:

- $\Delta^{\mathcal{I}_{\mathcal{K}}} = \{d_a \mid a \in \mathsf{Sig}(\mathcal{A}) \cap N_I\} \cup \{d_C \mid \exists r.C \in \mathsf{sub}(\mathcal{A}) \cup \mathsf{sub}(\mathcal{T})\}$,
- $A^{\mathcal{I}_{\mathcal{K}}} = \{d_D \mid D \sqsubseteq_{\mathcal{T}} A\} \cup \{d_a \mid \mathcal{K} \models A(a)\}$,
- $r^{\mathcal{I}_{\mathcal{K}}} = \{(d_D, d_E) \mid D \sqsubseteq_{\mathcal{T}} \exists r.E\} \cup \{(d_a, d_D) \mid \mathcal{K} \models \exists r.D(a)\} \cup \{(d_a, d_b) \mid r(a, b) \in \mathcal{A}\}$.

Note that canonical models for $\mathcal{EL}$ are always finite. The canonical model $\mathcal{I}_{C,\mathcal{T}}$ is in some sense a compact representation of the most general model for C and $\mathcal{T}$; for any other model $\mathcal{J}$ of $\mathcal{T}$ with an $d \in C^{\mathcal{J}}$, $(\mathcal{I}, d)$ can be simulated by d_C in $\mathcal{I}_{C,\mathcal{T}}$. Similarly, for any model $\mathcal{J}$ of $\mathcal{K}$ with $d = a^{\mathcal{J}}$ for an individual a, $(\mathcal{J}, d)$ is simulated by d_a in $\mathcal{I}_{\mathcal{K}}$.

Theorem 2.6 (By Lutz and Wolter [24])**.** *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox, C and D be $\mathcal{EL}$-concepts. Then:*

1. *for all models $\mathcal{I}$ of $\mathcal{T}$ and all elements $d \in \Delta^{\mathcal{I}}$ holds $d \in C^{\mathcal{I}}$ iff $(\mathcal{I}_{C,\mathcal{T}}, d_C) \lesssim (\mathcal{I}, d)$; and*
2. *$C \sqsubseteq_{\mathcal{T}} D$ iff $d_C \in D^{\mathcal{I}_{C,\mathcal{T}}}$ (or equivalently: $D \in \mathfrak{C}((\mathcal{I}_{C,\mathcal{T}}, d_C))$) iff $(\mathcal{I}_{D,\mathcal{T}}, d_D) \lesssim (\mathcal{I}_{C,\mathcal{T}}, d_C)$.*

When testing whether a given individual is a relaxed instance, we need to compute 'the best-fitting $\mathcal{EL}$-concept' that has the individual as an instance. This can be realized by the task of computing the most specific concept.

Definition 2.7 (most specific concept)**.** Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KB and a an individual from $\mathcal{A}$. A concept C is the *most specific concept (msc) of a w.r.t.* $\mathcal{K}$ (denoted $\text{msc}_{\mathcal{K}}(a)$) if it satisfies:

1. $\mathcal{K} \models C(a)$, and
2. for any concept D, $\mathcal{K} \models D(a)$ implies $C \sqsubseteq_{\mathcal{T}} D$.

A concept C is the *role-depth bounded most specific concept (k-msc) of a w.r.t.* $k \in \mathbf{N}$ *and* $\mathcal{K}$ (denoted k-$\text{msc}_{\mathcal{K}}(a)$) if it satisfies:

1. $\text{rd}(C) \leq k$,
2. $\mathcal{K} \models C(a)$, and
3. for any concept D with $\text{rd}(D) \leq k$, $\mathcal{K} \models D(a)$ implies $C \sqsubseteq_{\mathcal{T}} D$.

In general, the msc does not need to exist [25], if the ABox or the TBox contains any cycles. The k-msc however always exists, since the role-depth bound will cut off any cycles at the depth k. Both the msc, if it exists, and the k-msc are unique up to equivalence in $\mathcal{EL}$ [26]. Algorithms for computing the k-msc in $\mathcal{EL}$, and some of its extensions, have been studied [26, 27], and implemented [2].

2.2. Concept Similarity Measures

A concept similarity measure for concepts written in an arbitrary DL $\mathcal{L}$ is a function $\sim : \mathfrak{C}(\mathcal{L}) \times \mathfrak{C}(\mathcal{L}) \to [0, 1]$,[1] such that $C \sim C = 1$ for all concepts $C \in \mathfrak{C}(\mathcal{L})$. A value $C \sim D = 0$ means that the concepts C and D are totally dissimilar, while a value of 1 indicates total similarity.

A set of properties for CSMs, which are well-established properties for similarity measures collected from the literature or built on reasoning services for DLs, was presented in [15]. The framework devised in [15] allows to construct CSMs for $\mathcal{EL}$-concepts (possibly defined w.r.t. unfoldable TBoxes) that have these formal properties by instantiating the functions used in the framework. Such CSMs can be the basis for relaxed instance queries, if computed for concepts defined w.r.t. a (possibly empty) unfoldable TBox.

In this paper we also investigate CSMs for $\mathcal{EL}$-concepts defined w.r.t. general TBoxes. To the best of our knowledge there is no CSM to be found in the literature that takes *all* information from a general TBoxes into account. We extend the definition of the properties of CSMs from [15] to the case where general TBoxes are used.

Definition 2.8. Let $\mathcal{T}$ be a TBox. Then a concept similarity measure $\sim : \mathfrak{C}(\mathcal{EL}) \times \mathfrak{C}(\mathcal{EL}) \to [0, 1]$ w.r.t. $\mathcal{T}$ can have the following properties:

- *symmetric,* iff $C \sim D = D \sim C$;
- *equivalence invariant,* iff for all $C \equiv_{\mathcal{T}} D$ and all concepts E it holds that $C \sim E = D \sim E$;
- *equivalence closed,* iff $C \equiv_{\mathcal{T}} D \iff C \sim D = 1$;
- *bounded,* iff the existence of $E \neq \top$ with $C \sqsubseteq_{\mathcal{T}} E$ and $D \sqsubseteq_{\mathcal{T}} E$ implies $C \sim D > 0$;

[1] We generalize the previous notation and denote as $\mathfrak{C}(\mathcal{L})$ the set of all concepts that can be built in the DL $\mathcal{L}$.

- *dissimilar closed,* iff $C, D \neq \top$ and there is no $E \neq \top$ with $C \sqsubseteq_{\mathcal{T}} E$ and $D \sqsubseteq_{\mathcal{T}} E$ implies that $C \sim D = 0$;
- *subsumption preserving,* iff $C \sqsubseteq_{\mathcal{T}} D \sqsubseteq_{\mathcal{T}} E$ implies $C \sim D \geq C \sim E$;
- *reverse subsumption preserving,* iff $C \sqsubseteq_{\mathcal{T}} D \sqsubseteq_{\mathcal{T}} E$ implies $D \sim E \geq C \sim E$,

These formally defined properties make the outcome of a CSM with these properties more predictable for ontology users. The measures described in [15, 16] fulfill most of these properties. The parameterizable similarity measures from [15] additionally allow users to calibrate the measure to fit their expectations. In our setting of relaxed instance queries these parametrizable CSMs enable users to specify which features of query concepts can be relaxed and which should be kept.

3. Relaxed Instance Queries

In this section we introduce the main reasoning problem that we investigate; namely, answering instance queries which are relaxed through a CSM. We also provide a first approach for solving this problem.

Our main goal is to generalize query answering to allow for more relaxed solutions. Intuitively, given a concept C, we are not only interested in finding all the certain instances of C, but also those individuals that are *close* to being instances of C; we call these individuals the *relaxed instances* of C. Our motivation, as explained earlier, comes from the fact that users might be willing to loose some of the properties of the query concept to obtain a larger sample of answers. However, these answers must be as close to the original query C as possible.

Clearly, there exist many different ways in which one can define the relaxed instances of a concept, depending on the notion of 'closeness' used, and on the degrees of liberty allowed for the generalization of the query. One natural approach would be to try to decide which individuals are *similar* to any of the certain instances of C. Such a method requires a similarity measure defined over the *elements* of the domain, rather than on the concepts. A DL with a similarity measure over the domain elements was introduced in [28]. 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. We propose an approach based on instance similarity measures in Section 5.

A different idea that has been proposed is to simply generalize the concept C by considering named concepts that subsume C. In other words, to find the relaxed instances of C, one simply needs to compute the certain instances of every concept name A that subsumes C in a minimal way. 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.

A different idea would be to use different degrees of membership, as done for fuzzy and rough DLs. Relaxed instances of the concept C would be those with a large membership degree to C in the former case, and those that are indiscernible from any certain instance of C in the latter. Query answering over fuzzy DLs has been studied (see e.g. [29–31]), although not as thoroughly for the DL $\mathcal{EL}$. On the other hand, research on rough DLs is quite scarce, and to the best of

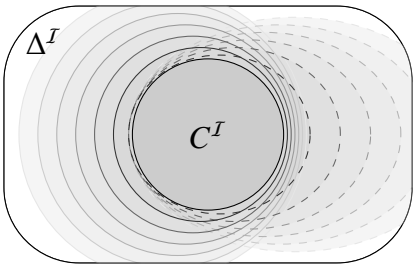

Figure 1: Relaxed instances w.r.t. two different CSMs, represented by continuous and dashed lines, respectively. Darker colors represent the relaxed instances of C w.r.t. higher degrees t.

our knowledge, the development of query answering techniques for those logics is only at its beginning [32]. The main drawback for these approaches is that that they allow

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. Obviously, the definition of this inference does not depend on the specific DL used. Hence, we define relaxed instances in general, although we investigate it only for $\mathcal{EL}$ throughout this paper.

Definition 3.1 (relaxed instance)**.** Let $\mathcal{L}$ be a DL, C be an $\mathcal{L}$-concept, $\sim$ a concept 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 by $a \in_t^{\sim} C$) iff there exists a concept $X \in \mathfrak{C}(\mathcal{L})$ such that $C \sim X > t$ and $\mathcal{K} \models X(a)$.

For brevity, we will denote as $\text{Relax}_t^{\sim}(C)$ the set of all relaxed instances of the concept 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; this dependency is depicted in Figure 1. For a fixed concept 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 larger thresholds t. We use two different kinds of lines (continuous and dashed, respectively) 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. For example, there are relaxed instances of C w.r.t. one similarity measure and threshold 0.99 which are not relaxed instances of C w.r.t. the other measure and threshold 0.5, and vice versa. However, these sets must always contain all the (certain) instances of C.

As mentioned before, our main goal is to find all the instances that belong to $\text{Relax}_t^{\sim}(C)$. From Definition 3.1, we know that

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

Thus, one could try to find all the relaxed instances of C by first computing all concepts X that are similar to C with degree greater than t, and then finding all the instances of these concepts X. However, this approach suffers from two main drawbacks. First, the set of all concepts that are similar to C with degree greater than t might be infinite, even modulo equivalence. Thus, although $\text{Relax}_t^{\sim}(C)$ is guaranteed to be finite, as it can contain only individual names appearing in the KB,

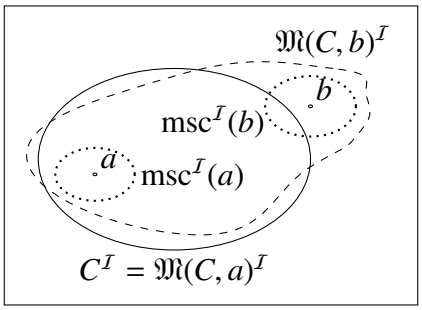

Figure 2: Two individuals, their most specific concepts (dotted), and the mimics of a concept C (continuous) w.r.t. these individuals (dashed).

it might be necessary to perform infinitely many instance queries to compute them all. Second, it is not known how to compute the concepts X that are similar to C with a degree greater than t. In fact, 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 first consider a different reasoning problem, where the task is to compute a concept that has a given individual a as an instance, and is the most similar to C w.r.t. the given CSM with this property. We call this concept the *mimic* of C w.r.t. a.

Definition 3.2 (mimic)**.** Let $\mathcal{L}$ be a DL, $\mathcal{K}$ an $\mathcal{L}$-knowledge base, $a \in N_I$ an individual name, C an $\mathcal{L}$-concept, and $\sim$ a concept similarity measure. An $\mathcal{L}$-concept D is called a *mimic* of C w.r.t. a, denoted with $D \in \mathfrak{M}(C,a)$, 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 $\mathcal{L}$-concepts E, if a is an instance of E, then $C \sim D \geq C \sim E$.

Obviously, as for relaxed instances, the mimic strongly depends on the similarity measure chosen. Intuitively, a mimic of C w.r.t. an individual a is a concept that is as similar to C as possible, while still having a as an instance. Figure 2 depicts the idea of mimics. In the figure, a and b are two individuals and the continuous line represents the interpretation of C. In this case, the individual a is an instance of C, while b is not. For each of these individuals, the dotted lines depict the interpretation of their most specific concepts. Since a is already an instance of C, C is also a mimic of C w.r.t. a; this follows trivially from the fact that $C \sim C = 1$. The dashed line depicts the interpretation of a mimic of C w.r.t. b. Since the mimic of b must keep b as an instance, it must subsume the msc of b. However, it is not necessarily a subsumer of C. In fact, as depicted in Figure 2, there might be instances of C that do not belong to the mimic of C w.r.t. b.

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 with an empty TBox and the ABox $\mathcal{A} = \{(A \sqcap B)(a)\}$, and let $\sim$ be any similarity measure such that $A \sim C = 0.5$, $B \sim C = 0.5$, $(A \sqcap B) \sim C = \max\{A \sim C, B \sim C\} = 0.5$, and $D \sim C = 0$ for all other concepts D. 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. For our task of computing relaxed instances, we are not interested in finding all of them, but only one. We will use the mimics only to decide whether an individual belongs to any concept that is similar to C with a

degree larger than the given threshold. Since all mimics have the exact same degree of similarity to C, the result obtained by using them is independent of the specific mimic computed.

We can use mimics to compute the relaxed instances of a given concept. The idea is to compute, for each individual a appearing in the knowledge base $\mathcal{K}$, a mimic of C w.r.t. a. If this mimic has similarity greater than t with C, then a must be a relaxed instance of C, and hence is given as an answer to the relaxed query; 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 3.3. *Let $\mathcal{K}$ be a knowledge base, a an individual occurring in $\mathcal{K}$, C a concept, $\sim$ a concept similarity measure, and $t \in [0, 1)$. Then $a \in \mathrm{Relax}_t^{\sim}(C)$ iff there is a mimic D of C w.r.t. a such that $C \sim D > t$.*

In the next section we 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, computing the similarity between C and this mimic provides us with enough information to decide whether a is a relaxed instance of C or not, up to degree t. However, as we will see, computing a mimic might already be an expensive task itself. We partially alleviate this issue by a simple optimization criterion: if a mimic D of C w.r.t. a is similar to C to degree greater than t, then all certain instances of D must also be relaxed instances of C w.r.t. this threshold. In particular this means that we can then avoid computing the mimics for all other individuals that are also instances of D.

4. Computing Relaxed Instances for Unfoldable $\mathcal{EL}$-TBoxes

As we have seen in the previous section, in order to answer relaxed instance queries, it suffices to compute, for every individual a, a mimic of the target concept C w.r.t. a, and then measure the similarity between these concepts. In general, there are infinitely many concepts for which an individual a is an instance of. Thus, we cannot expect to compute a mimic by a simple enumeration of these concepts. On the other hand, since only *one* mimic is needed, it might be possible to explore the concepts in a goal-oriented manner by increasing the similarity until a mimic is found. In this section we show some conditions that guarantee that the computation of a mimic takes only finite time, by exploring only a limited number of concepts.

Recall that the notion of a mimic combines a property that is based on the semantics, namely, that it must have a as an instance, and a syntactic property, i.e., that it must be similar to C to degree greater than t. The semantic property provides us with a strategy to initialize the search for a mimic. Since any mimic D of C w.r.t. a must always have a as an instance, by definition of the msc, we have that $\mathrm{msc}(a) \sqsubseteq_{\mathcal{T}} D$ must hold. In other words, any mimic is always a generalization of $\mathrm{msc}(a)$. If the concept similarity measure $\sim$ is equivalence invariant, then we can find such a mimic through a syntactic manipulation of $\mathrm{msc}(a)$. The idea is that, by removing some of the conjuncts in the description of the concept, we always generalize it. Hence, we need only to detect which of these generalizations increases the similarity to C.

Definition 4.1 (generalized concept)**.** Let C be a concept of the form

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

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

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

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

Intuitively, a generalized concept is obtained simply by removing conjuncts from the description of C. Obviously, the result obtained through this generalization depends strongly on the syntactic shape of the concept C; that is, two equivalent concepts C and C' may have different generalized concepts. Recall that our aim is to generalize the msc of a; as long as we restrict ourselves to unfoldable TBoxes, we can simply expand the concepts to remove any ambiguity.

Definition 4.2 (fully expanded concept)**.** Let $\mathcal{T}$ be an unfoldable $\mathcal{EL}$-TBox. The concept C is *fully expanded* w.r.t. $\mathcal{T}$ iff it only contains primitive concept names, i.e., concept names that only occur on the right-hand side of the concept definitions in $\mathcal{T}$.

Notice that any concept can be expanded by simply replacing all defined concept names in the concept by their definitions. The intuition is that a fully expanded concept C contains all its (fully expanded) subsumers explicitly as sub-concept descriptions. 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 4.3. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an unfoldable $\mathcal{EL}$-knowledge base, a be an individual from $\mathcal{A}$, C be an $\mathcal{EL}$-concept, and $\sim$ be an equivalence invariant concept similarity measure. Let further $E = \mathrm{msc}(a)$ be the fully expanded most specific concept of a. Then there is a mimic $D \in \mathfrak{M}(C, a)$ of C w.r.t. a and $\mathcal{K}$ that is a generalized concept of E.*

Proof sketch. We show that any concept having a as an instance is equivalent to a generalized concept of $\mathrm{msc}(a)$. Let F be a concept with $\mathcal{K} \models F(a)$. 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 F must also be part of the concept E, but F may contain redundancies that can simply be removed to get an equivalent concept. Thus F is equivalent to a generalized concept of E. □

Notice that if the ABox $\mathcal{A}$ is cyclic, then the msc for an individual a might contain a chain of infinitely nested existential restrictions. Thus, this msc might not be expressible as a concept with a finite description, and in particular its fully expanded version would be infinite. In this case, $\mathrm{msc}(a)$ has infinitely many generalized concepts of finite size, making it unfeasible to use Lemma 4.3 to find a mimic.

On the other hand, the fully expanded query concept C is always finite, and hence has a finite role-depth. Moreover, many structural CSMs used in practice, like those presented in [15, 16], are based on a recursive computation of the similarities between concepts at the same role-depth. This means that the computation of the similarity only recursively visits existential restrictions, until one subconcept has no further existential restrictions. If the two concepts have a role-depth k_1 and k_2, respectively, then the similarity depends only on those parts of the concepts up to role-depth $\min(k_1, k_2) + 1$, and, more importantly, to compute the maximal similarity between generalized

concepts of the msc(a) and C, we only need to consider concepts with a maximal role-depth up to $k = \mathrm{rd}(C)$, i.e., can restrict to the k-msc(a). We generalize this property of recursive similarity measures as follows:

Definition 4.4. A concept similarity measure $\sim$ is *successor-closed*, if it has the following properties:

1. For all concepts C and D, and all $A_i \in N_C$ it holds
$$C \sim \bigsqcap_{i \in I} A_i \;\geq\; C \sqcap \exists r.D \sim \bigsqcap_{i \in I} A_i,$$
2. $\sim$ is monotone in the similarities of its successors, i.e., for all concepts A, B, C, D with $A \sim B \leq A \sim C$, we have
$$(D \sqcap \exists r.A \;\sim\; D \sqcap \exists r.B) \;\leq\; (D \sqcap \exists r.A \;\sim\; D \sqcap \exists r.C)$$

The first property expresses that the similarity between a simple conjunction of concept names and a concept C can only decrease, if we add existential restrictions to C. If our similarity measure $\sim$ is successor-closed, then we can limit the computation of the msc to a given role-depth, i.e., use the k-msc, without losing generality in our approach, which gives a result similar to Lemma 4.3.

Lemma 4.5. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{EL}$-knowledge base with an unfoldable TBox, a an individual from $\mathcal{A}$, C be an $\mathcal{EL}$-concept in $\sim$-normal form, and $\sim$ an equivalence invariant and successor-closed concept similarity measure. If $k = \mathrm{rd}(C)$ and E is the fully expanded role-depth bounded most specific concept of a to role-depth k, 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 4.3 we know that there exists a mimic F of C w.r.t. a that is a generalized concept of the (possibly infinite) fully expanded description of msc(a). Let F_k be the concept obtained by restricting F to role-depth k; that is, by removing all existential restrictions beyond this depth. Since E is the fully expanded k-msc of a, F_k must also be a generalized concept of E. 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 $k = 0$, the role-depth of both C and E is limited to 0; hence, they are of the form $C = \bigsqcap_{i \in I} A_i$ and $E = \bigsqcap_{j \in J} B_j$ where each A_i and B_j is a concept name. Since F_0 is a generalized concept of E, it must be of the form $F_0 = \bigsqcap_{j \in J'} B_j$ with $J' \subseteq J$, and hence $F = F_0 \sqcap \bigsqcap_{h \in H} \exists r_h.G_h$. Since $\sim$ is successor-closed we then have that

$$F_0 \sim C \;\geq\; F_0 \sqcap \bigsqcap_{h \in H} \exists r_h.G_h \sim C \;=\; F \sim C.$$

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 $\mathrm{rd}(C_h), \mathrm{rd}(E_l) \leq k-1$ for all $h \in H$, $l \in L$. Once again, since F_k is a generalized concept of E, it must be of the form $F_k = \bigsqcap_{j \in J'} B_j \sqcap \bigsqcap_{l \in L'} \exists r_l.G_l$, where $J' \subseteq J$, $L' \subseteq L$ and each G_l is a generalized concept of E_l. Moreover, F is of the form $F = \bigsqcap_{j \in J'} B_j \sqcap \bigsqcap_{l \in L'} \exists r_l.G'_l$, where G_l and G'_l coincide at role-depth 0. By induction hypothesis, we then have that $G_l \sim C_h \;\geq\; G'_l \sim C_h$ holds for all $h \in H$ and $l \in L'$. Since the similarity measure $\sim$ is structural, this implies that $F_k \sim C \;\geq\; F \sim C$, which finishes the proof. □

Procedure: relaxed-instance? $(a, C, \mathcal{K}, \sim, t)$
Input: a: individual in $\mathcal{K}$; C: $\mathcal{EL}$-concept; $\mathcal{K}$: $\mathcal{EL}$-KB with unfoldable TBox; $\sim$: equivalence-invariant and successor-closed CSM; t: threshold;
Output: true if $a \in_t^{\sim} C$ w.r.t. $\mathcal{K}$; otherwise, false

1: $k := \mathrm{rd}(C)$
2: $E := k\text{-msc}(a)$ w.r.t. $\mathcal{K}$
3: guess a generalized concept F of E
4: **return** $F \sim C > t$

Algorithm 1: Computation algorithm for relaxed instances in $\mathcal{EL}$.

This lemma provides us with restrictions on the concept similarity measure used that guarantee that a mimic of C w.r.t. a can always be found by comparing a finite set of concepts, which is formed by all the generalized concepts of the fully expanded role-depth bounded msc of the individual a.

Recall that our original goal was to decide whether an individual a is a relaxed instance of C w.r.t. a given threshold t. To achieve this, it is not necessary to compute a mimic of C w.r.t. a and compute its similarity degree with C. Indeed, it suffices to compute *any* concept D that contains a as an instance and $C \sim D > t$ to guarantee that a is a relaxed instance of C. Algorithm 1 describes a non-deterministic procedure for checking relaxed instances based on this idea.

The algorithm receives as input an $\mathcal{EL}$-KB $\mathcal{K}$, an individual a, a concept C, a successor-closed and equivalence invariant concept similarity measure $\sim$ and a specified threshold t, and decides, non-deterministically, whether a concept D that is similar enough to C and contains a exists. In practical terms, this algorithm behaves better than trying to compute a mimic first, since there is no need to verify that there is no other concept that is more similar to C than the one guessed, as long as the similarity is beyond the threshold t.

Corollary 4.6. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{EL}$-knowledge base, C an $\mathcal{EL}$ concept, a an individual in $\mathcal{K}$, $\sim$ an equivalence invariant, and successor-closed concept similarity measure, and $t \in [0, 1)$. Then* relaxed-instance?$(a, C, \mathcal{K}, \sim, t)$ *decides whether $a \in_t^{\sim} C$ w.r.t. $\mathcal{K}$.*

Proof. Lemma 4.5 shows that a mimic of C w.r.t. a is a generalized concept of $E = k\text{-msc}(a)$ for $k = \mathrm{rd}(C)$. Thus, if the algorithm returns false, we know that no generalized concept F exists with $C \sim F > t$, 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 $a \notin \mathrm{Relax}_t^{\sim}(C)$. If the algorithm returns true, the guessed concept F shows $a \in \mathrm{Relax}_t^{\sim}(C)$, since a is an instance of F and $F \sim C > t$. □

Guessing a generalized concept F of a concept description E can be done in time linear in the 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 $k\text{-msc}(a)$ can be exponential in k although still polynomial in $|\mathcal{K}|$ [26]. Since k is chosen to be $\mathrm{rd}(C)$, it is bounded linearly by $|C|$. Thus the non-deterministic algorithm runs in exponential time, assuming that $\sim$ can be computed in

at most non-deterministic exponential time, too. However, the exponential blow-up depends only on the role-depth k of C. Since C is the query concept, it is reasonable to assume that k would be typically small, and the blow-up will play only a minor role for practical applications. Moreover, if we consider a constant bound on the role-depth of the query concepts, then the algorithm runs in (non-deterministic) polynomial time, assuming that the computation of $\sim$ is in NP.

To obtain a deterministic algorithm, a mimic of C w.r.t. a can be computed by enumerating all generalized concepts of k-msc(a) and choosing one with the maximal similarity to C. Obviously, there exist several optimizations that can be implemented. For example, for every instance a of C, we can always automatically return true, since C itself is always a mimic, with similarity above any given threshold to C. A second optimization, as described before, is to stop the search as soon as a generalized concept F with $C \sim F > t$ is found; at that point, a is guaranteed to be a relaxed instance of C. Finally, once that a concept D that guarantees that a is a relaxed instance of C has been found, all other instances of D must also be relaxed instances of C, by definition. Hence, they can all be added to the set of relaxed instances, without the need to compute their mimics, or any other reasoning task.

5. Computing Relaxed Instances for General TBoxes

The approach to answer relaxed instance queries presented in the previous section cannot be used if the KB uses a general $\mathcal{EL}$-TBox. The main reason for this is that it requires the query concept to be fully expanded. This expansion step can only be done for unfoldable terminologies, and does not terminate once cyclic definitions are involved. Indeed, most of the structural similarity measures introduced in the literature so far also rely on expansion: Once the concepts are expanded w.r.t. the terminological knowledge, the similarity between them can be computed by just comparing the tree structures of the (expanded) concepts without further reference to the TBox. While this approach is conceptually appealing, it is always limited to unfoldable TBoxes, which is a strong restriction for modern knowledge representation applications.

On the other hand, the previous approach works for arbitrary similarity measures, that only need to satisfy a few properties—notably, they should be equivalence-invariant and successor-closed. The flexibility of these similarity measures might be appealing from a knowledge engineering point of view; however, it makes the algorithm quite inefficient. While optimizations to the strategy of finding the mimic in all generalized concepts of the msc are imaginable for specific similarity measures, in the general case all possible generalized concepts need to be checked, as described in the previous section.

This section therefore has two goals. The first is to introduce a family of similarity measures that works w.r.t. general TBoxes. While the framework of these similarity measures is fixed, they are still flexible enough to be useful in many situations. The second goal is to devise an algorithm to compute relaxed instances of a knowledge base w.r.t. this new family of similarity measures. Since this algorithm only deals with these specific, new similarity measures, it can exploit this additional knowledge and is able to operate more efficiently than the general algorithm introduced in the last section. In particular, we will show that, when applied to unfoldable TBoxes, this algorithm computes relaxed instances in polynomial time in the size of the input, which greatly improves the NEXP-time upper bound obtained for the previous algorithm.

Our similarity measure does not use expanded concepts w.r.t. the TBox, which may not always be finitely expressible in $\mathcal{EL}$, but instead uses canonical models. These models also expand the concept with the knowledge from the TBox, but are always finite, although they may contain cycles. Thus, instead of computing the similarity between two concepts C and D w.r.t. a general $\mathcal{EL}$-TBox $\mathcal{T}$ directly, we use the canonical models $\mathcal{I}_{C,\mathcal{T}}$ and $\mathcal{I}_{D,\mathcal{T}}$ and compute the similarity between the elements d_C and d_D in these interpretations. This means that we have to define a similarity measure on finite pointed interpretations, i.e. elements together with the interpretation they occur in.

An interpretation similarity measure (ISM) is defined as a similarity measure on finite pointed interpretations, i.e., a function of the type $\mathfrak{P} \times \mathfrak{P} \to [0, 1]$. It maps any pair of pointed interpretations to a similarity value between 0 and 1. We denote ISMs by $\sim_{\mathfrak{P}}$. The restriction of ISMs to finite pointed interpretations is important later – in the following, whenever we mention pointed interpretations in the context of ISMs, we indeed assume them to be finite.

There are various desirable properties that ISMs can have. We concentrate here on those that directly transfer from similar properties of CSMs introduced before. Given suitable simulation relations $\lesssim$ and $\simeq$ (like those in Definition 2.3 for $\mathcal{EL}$), we call an interpretation similarity measure:

- *symmetric*, iff $p \sim_{\mathfrak{P}} q = q \sim_{\mathfrak{P}} p$ for all $p, q \in \mathfrak{P}$;
- *bounded*, iff $\mathfrak{C}(p) \cap \mathfrak{C}(q) \supset \{\top\}$ implies $p \sim_{\mathfrak{P}} q > 0$ for all $p, q \in \mathfrak{P}$;
- *dissimilar closed*, iff $\mathfrak{C}(p) \cap \mathfrak{C}(q) = \{\top\}$ implies $p \sim_{\mathfrak{P}} q = 0$ for all $p, q \in \mathfrak{P}$ with $\mathfrak{C}(p) \supset \{\top\}$ and $\mathfrak{C}(q) \supset \{\top\}$;
- *equisimulation invariant*, iff $p \simeq q$ implies $p \sim_{\mathfrak{P}} u = q \sim_{\mathfrak{P}} u$ for all $p, q, u \in \mathfrak{P}$;
- *equisimulation closed*, iff $p \simeq q \iff p \sim_{\mathfrak{P}} q = 1$ for all $p, q \in \mathfrak{P}$;
- *simulation preserving*, iff $r \lesssim q \lesssim p$ implies $p \sim_{\mathfrak{P}} q \geq p \sim_{\mathfrak{P}} r$ for all $p, q, r \in \mathfrak{P}$;
- *reverse simulation preserving*, iff $r \lesssim q \lesssim p$ implies $q \sim_{\mathfrak{P}} r \geq p \sim_{\mathfrak{P}} r$ for all $p, q, r \in \mathfrak{P}$.

5.1. The Interpretation Similarity Measure $\sim_i$

We now define a parameterizable ISM $\sim_i$, using the simulation relations defined in Definition 2.3, which correspond to concept subsumption and equivalence in $\mathcal{EL}$. This is important when lifting those properties to the CSMs $\sim_c$.

Given a pointed interpretation $p = (\mathcal{I}, d)$, we denote with

$$\mathrm{CN}(p) = \{A \in N_C \mid d \in A^{\mathcal{I}}\}$$
$$\mathrm{SC}(p) = \{(r, (\mathcal{I}, e)) \in N_R \times \mathfrak{P} \mid (d, e) \in r^{\mathcal{I}}\}$$

the set of concept names that d is an instance of in $\mathcal{I}$, and the set of direct successors of d in $\mathcal{I}$, respectively.

For two pointed interpretations to be perfectly similar, they need to have the same set of concept names and have edges labeled with the same roles going to perfectly similar successor elements. Otherwise, the most similar concept names and the most similar direct successors are compared and a similarity value is computed from this pair. In essence, $\sim_i$ is a feature-based similarity measure where the concept names and successors of an interpretation element are its features.

Note that, compared to the interpretation similarity measure introduced in [20], we do not rely on pairings between the concept names and successors of the two pointed interpretations, but instead look at each concept name and successor of one pointed interpretation and find the best concept name or successor of the other pointed interpretation. This is done both ways. The advantages of this approach over pairings is that the results are more predictable, as pairings may contain many more comparisons than necessary for successors or concept names, which can increase the total similarity, while the approach presented here will give similar weights to all features.

The ISM $\sim_i$ extends a primitive measure that is described next. Formally, we consider a *primitive measure*

$$\sim_{\text{prim}} : N_C \times N_C \cup N_R \times N_R \to [0, 1]$$

that assigns similarity values to each pair of concept names and each pair of role names. Any primitive measure has to satisfy the property that $x \sim_{\text{prim}} x = 1$ for any concept or role name x. Additionally, for the similarity measure $\sim_i$ to be symmetric, $\sim_{\text{prim}}$ needs to be symmetric as well. We give a default primitive measure, that simply assigns similarity 0 to pairs of different concept or role names x and y:

$$x \sim_{\text{default}} y = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

However, other primitive measures are imaginable and useful. For example, one might want to express that two amounts Medium and High are more similar than Low and High, which can be achieved by using a primitive measure with Medium $\sim_{\text{prim}}$ High = 0.5 and Low $\sim_{\text{prim}}$ High = 0.

Additionally, one can assign weights to different concept and role names using a *weighting function*

$$g : N_C \cup N_R \to \mathbb{R}_{>0}$$

to prioritize different features in the similarity measure. This function g is extended to pairs of concept or role names as $g(A, B) = \max(g(A), g(B))$ and $g(r, s) = \max(g(r), g(s))$. Finally, we need a constant w that allows for discounting of successors, and should have a value $0 < w < 1$.

Any primitive measure $\sim_{\text{prim}}$, weighting function g, and discounting factor w can then be extended to a similarity measure on pointed interpretations by recursively traversing the interpretation graphs, for each pair of elements looking at all features (concept names and successors), and finding the best-matching feature of the second element.

Definition 5.1. Given a primitive measure $\sim_{\text{prim}}$, a weighting function g and the discounting factor w, the interpretation similarity measure $\sim_i(\sim_{\text{prim}}, g, w) : \mathfrak{P} \times \mathfrak{P} \to [0, 1]$ is defined as follows:

$$p \sim_i q = \frac{\text{sim}_{\text{CN}}(p, q) + \text{sim}_{\text{CN}}(q, p) + \text{sim}_{\text{SC}}(p, q) + \text{sim}_{\text{SC}}(q, p)}{|\text{CN}(p)| + |\text{CN}(q)| + |\text{SC}(p)| + |\text{SC}(q)|} \tag{1}$$

where

$$\text{sim}_{\text{CN}}(p, q) = \sum_{A \in \text{CN}(p)} \max_{B \in \text{CN}(q)} A \sim_{\text{prim}} B, \text{ and}$$

$$\text{sim}_{\text{SC}}(p, q) = \sum_{(r,p') \in \text{SC}(p)} \max_{(s,q') \in \text{SC}(q)} (r \sim_{\text{prim}} s)(w + (1 - w)(p' \sim_i q')).$$

If all of the sets $\mathrm{CN}(p)$, $\mathrm{CN}(q)$, $\mathrm{SC}(p)$, and $\mathrm{SC}(q)$ are empty for pointed interpretations p, q, we define $p \sim_i q = 1$. This case only happens if $\mathfrak{C}(p) = \mathfrak{C}(q) = \{\top\}$.

We often write simply $\sim_i$ to denote all the different similarity measures $\sim_i(\sim_{\mathrm{prim}}, g, w)$ for some primitive measure $\sim_{\mathrm{prim}}$, weighting function g, and discounting factor w.

Example 5.2. We now show how the similarity measure works for the toy ontology introduced in Example 2.2. We assume the primitive measure $\sim_{\mathrm{prim}}$, which is nearly the same as the default primitive measure, with two exceptions: The similarity between Low and Medium as well as between Medium and High is 0.5 instead of 0. We also assume the default weighting function g that assigns weight 1 to all concept and role names, and a discounting factor of $w = 0.8$.

We want to compute the similarity between the two pointed interpretations $(\mathcal{I}, d)$ and $(\mathcal{J}, e)$ described in Figure 3. To compute the similarity between these two pointed interpretations, we need to find for each concept name and each successor of any of the two elements the best matching concept name or successor of the other element. For this we need the similarities of all successors of the elements d and e. However, since the primitive similarity between different role names is always 0, it is enough to only look at successors with the same role name:

- The hasLoad-successors have a similarity 0.667, as both elements have the concept name Amount, but the successor of e is missing the concept name High, resulting in a similarity value of $\frac{1+(1+0)}{3} = 0.667$.
- The most similar concept names for the two hasLatency-successors are (Amount, Amount) and (Low, Medium), which yields a similarity value of $\frac{(1+0.5)+(1+0.5)}{4} = 0.75$.
- For the two provides-successors of d and e, respectively, both are instance of Service, while the concept names VideoStreamService and DatabaseService have no correspondence in the other element. Similarly, the outgoing roles of these elements all have different role names, resulting in a value of 0 for $\mathrm{sim}_{\mathrm{SC}}$ in both directions. Overall, this yields a similarity of $\frac{(1+0)+(1+0)+0+0}{2+2+1+2} = 0.286$ for the two services.

Using this, we can finally compute the similarity between d and e by computing $\mathrm{sim}_{\mathrm{CN}}$ and $\mathrm{sim}_{\mathrm{SC}}$

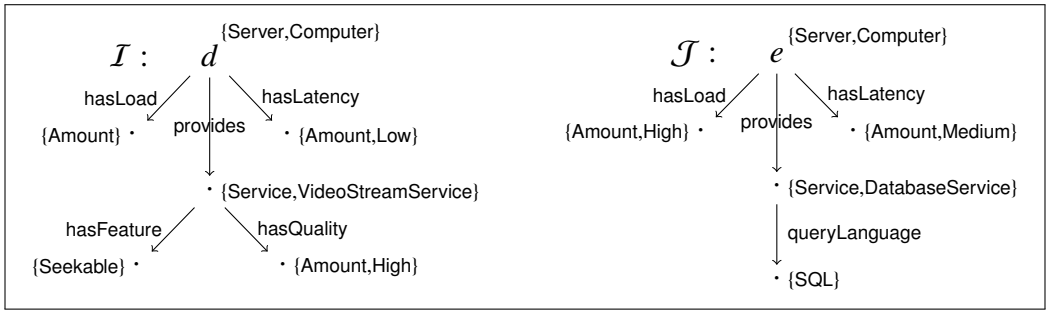

Figure 3: The pointed interpretations $(\mathcal{I}, d)$ and $(\mathcal{J}, e)$ from Example 5.2

for both directions (p, q) and (q, p) and dividing by the sum of all weights:

$$\begin{aligned}
\mathrm{sim}_{\mathrm{CN}}(p, q) &= \mathrm{sim}_{\mathrm{CN}}(q, p) = 1 + 1 \\
\mathrm{sim}_{\mathrm{SC}}(p, q) &= \mathrm{sim}_{\mathrm{SC}}(q, p) = (0.2 + 0.8 \cdot 0.667) + (0.2 + 0.8 \cdot 0.75) + (0.2 + 0.8 \cdot 0.286) \\
&= 1.962 \\
(\mathcal{I}, d) \sim_i (\mathcal{I}, e) &= \frac{2 + 2 + 1.962 + 1.962}{2 + 2 + 3 + 3} = 0.792
\end{aligned}$$

This similarity value may not be satisfying: Even though both servers have a different latency, the load of one server is not given, and they provide totally different services, the similarity is quite high. The main reason is that the more general concept names like Computer, Service, and Amount, that both interpretations have in common, contribute as much as the other concept names that carry the actual values, like Low or DatabaseService. To rectify this, one would change the weighting function g to decrease the weight of the more general concepts. Since the user is probably most interested in finding a server that provides the service he needs, the weight of the role name provides might also be increased. Additionally, it might be a good idea to also increase the discounting factor w to further increase the influence of the actual similarity between successors in the successors pairing, like the two services.

Before we show the formal properties of $\sim_i$, we need to show that its recursive definition is actually well-defined, even for cyclic interpretations.

Theorem 5.3. *The similarity measure $\sim_i$ is well-defined, i.e., $p \sim_i q$ defined in Equation* (1) *has a unique solution for all pointed interpretations $p, q \in \mathfrak{P}$.*

Proof. If we fix the two interpretations $\mathcal{I}$ and $\mathcal{J}$, we can view $\sim_i$ as an iterative function that 'refines' the similarities between any two elements $(c, d) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}$, i.e., a function on the vector space $\mathbb{R}^{|\Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}|}$. In particular, since the value of $p' \sim_i q'$ in Equation 1 is always multiplied with w (there may be other factors, which are always less than 1), all partial derivative of $\sim_i$ are at most w and thus $\sim_i$ is Lipschitz continuous with a Lipschitz constant of at most w. As $w < 1$, this means that $\sim_i$ is a contraction mapping on $\mathbb{R}^{|\Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}|}$. But then, the Banach fixed-point theorem implies that $\sim_i$ has a unique fixed point in $\mathbb{R}^{|\Delta^{\mathcal{I}} \times \Delta^{\mathcal{J}}|}$, and indeed the iteration of $\sim_i$ on any starting tuple (like starting with a similarity of 0 between any pair of elements) converges to this fixed point [33]. This unique fixed-point means that Equation (1) has a unique solution for any $(\mathcal{I}, a) \sim_i (\mathcal{J}, b)$ (which corresponds exactly to the value between a and b for the fixed point) and is thus well-defined. □

Note that $\sim_i$ as defined above is not necessarily equisimulation closed and equisimulation invariant. The reason is that the similarity between successors is always undirected, while simulations are directed, which gives rise to problems for the case where one successor of an element simulates a second successor in one pointed interpretation, but not in an equisimilar one. To regain the properties equisimulation invariant and closed, one can first normalize the interpretations $\mathcal{I}$ and $\mathcal{J}$ before applying the similarity measure.

Definition 5.4 (normal form for interpretations)**.** An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is in *normal form* if there are no elements $a, b, c \in \Delta^{\mathcal{I}}$, $b \neq c$, with $\{(a, b), (a, c)\} \subseteq r^{\mathcal{I}}$ and $(\mathcal{I}, b) \lesssim (\mathcal{I}, c)$, i.e., no node has two successor nodes for the same role name that are in a simulation relation.

Any interpretation $\mathcal{I}$ can be transformed into normal form as follows:

1. Remove all edges $(a, b) \in r^{\mathcal{I}}$ in the interpretation graph, for which there exists an edge $(a, c) \in r^{\mathcal{I}}$ with $(\mathcal{I}, b) \lesssim (\mathcal{I}, c)$ but not $(\mathcal{I}, b) \simeq (\mathcal{I}, c)$.
2. For all edges $(a, b_0) \in r^{\mathcal{I}}$, check if there are other edges $(a, b_i) \in r^{\mathcal{I}}$, $i > 0$, with $(\mathcal{I}, b_0) \simeq (\mathcal{I}, b_i)$ and choose one representative b_j; then remove all other edges (a, b_i), $i \neq j$, from $r^{\mathcal{I}}$.

Equisimilar pointed interpretations will always be normalized into a unique structural normal form, i.e., both pointed interpretations will have the same number of pairwise equisimilar successors. This is true even though the normalization steps given above are nondeterministic. Since we only consider finite pointed interpretations, the normalization procedure is well-defined and can be computed in polynomial time in the size of the pointed interpretation, as simulations can also be computed in P-time.

Now, we can finally show the properties of the ISM $\sim_i$. For the default primitive measure, $\sim_i$ is symmetric, bounded, dissimilar closed, equisimulation invariant, and equisimulation closed for all normalized pointed interpretations. For other primitive measures $\sim_{\text{prim}}$, $\sim_i$ will always be bounded and equisimulation invariant, but the other properties depend on the properties of $\sim_{\text{prim}}$, as given by the following theorem.

Theorem 5.5. *Let $\sim_i(\sim_{\text{prim}}, g, w)$ be instantiated with a primitive measure $\sim_{\text{prim}}$, a weighting function g, and constant $w \in (0, 1)$. Then $\sim_i$ has the following properties (w.r.t. to the simulation relations $\lesssim$ and $\simeq$ given in Definition 2.3):*

1. *$\sim_i$ is symmetric, if the primitive measure $\sim_{\text{prim}}$ is symmetric;*
2. *$\sim_i$ is bounded;*
3. *$\sim_i$ is dissimilar closed, if the primitive measure $\sim_{\text{prim}}$ does not assign a similarity value greater than 0 to different concept or role names.*
4. *$\sim_i$ is equisimulation invariant for normalized interpretations; and*
5. *$\sim_i$ is equisimulation closed for normalized interpretations, if the primitive measure $\sim_{\text{prim}}$ does not assign the similarity value 1 to different concept or role names.*

Equation (1) in the definition of $\sim_i$ cannot be used directly to compute the similarity value, since cycles in the interpretation would lead to infinite recursion. Instead, one can view the equation as an iterative algorithm: When starting with a similarity value of 0 between all elements of two interpretations $\mathcal{I}$ and $\mathcal{J}$, and iteratively applying the equation to update those similarity values between all elements, they will converge to the solution. This follows from the proof of Theorem 5.3.

Using the ISM $\sim_i$, we can now define a concept similarity measure $\sim_c (\sim_{\text{prim}}, g, w)$ on $\mathcal{EL}$-concepts w.r.t. a general $\mathcal{EL}$-TBox $\mathcal{T}$ as follows:

$$C \sim_c D = (\mathcal{I}'_{C,\mathcal{T}}, d_C) \sim_i (\mathcal{I}'_{D,\mathcal{T}}, d_D),$$

where $\mathcal{I}'_{C,\mathcal{T}}$ and $\mathcal{I}'_{D,\mathcal{T}}$ are the normalized canonical models of the concepts C and D w.r.t. $\mathcal{T}$.

Example 5.6. Incidentally, the interpretations $\mathcal{I}$ and $\mathcal{J}$ given in Example 5.2 correspond exactly to the (normalized) canonical models $\mathcal{I}_{C,\mathcal{T}}$ and $\mathcal{I}_{D,\mathcal{T}}$ of the concepts

$$\begin{aligned} C = {} & \mathsf{Server} \sqcap \exists \mathsf{hasLatency.Low} \sqcap \exists \mathsf{provides.(VideoStreamService} \sqcap \\ & \exists \mathsf{hasFeature.Seekable} \sqcap \exists \mathsf{hasQuality.High)} \\ D = {} & \mathsf{Server} \sqcap \exists \mathsf{hasLoad.High} \sqcap \exists \mathsf{hasLatency.Medium} \sqcap \\ & \exists \mathsf{provides.(DatabaseService} \sqcap \exists \mathsf{queryLanguage.SQL)} \end{aligned}$$

Thus $C \sim_c D = 0.792$.

The concept similarity measure $\sim_c$ inherits the formal properties of the ISM $\sim_i$, since the properties for interpretation similarity measures were defined to correspond exactly to the properties for concept similarity measures given in the preliminaries.

Theorem 5.7 (Properties of $\sim_c$)**.** *For a primitive measure $\sim_{\text{prim}}$, a weighting function g, and a discounting factor w, the concept similarity measure $\sim_c(\sim_{\text{prim}}, g, w)$ is symmetric, bounded, dissimilar closed, equivalence invariant, and equivalence closed, if $\sim_i(\sim_{\text{prim}}, g, w)$ is symmetric, bounded dissimilar closed, equisimulation invariant and equisimulation closed, respectively.*

5.2. Computing Relaxed Instances w.r.t. $\sim_c$

First we define the notion of *fully expanded concepts* also for the case of general $\mathcal{EL}$-TBoxes:

Definition 5.8 (fully expanded concept)**.** Let $\mathcal{T}$ be a general $\mathcal{EL}$-TBox. A (possibly complex) concept C is *fully expanded* w.r.t. $\mathcal{T}$ iff for all GCIs $D \sqsubseteq E \in \mathcal{T}$ with $C \sqsubseteq_{\mathcal{T}} \exists r_1 \ldots \exists r_n.D$ we have that $\exists r_1 \ldots \exists r_n.E$ is a generalized concept of C.

This basically means that a fully expanded concept explicitly includes all knowledge expressed in the TBox. Note that this definition is not constructive in the sense that it may yield concepts of infinite size, but we will see now how to avoid this.

For the computation of relaxed instances for $\sim_c$, recall that $a \in Relax_t^{\sim}(Q)$ can be computed for terminologies by checking all generalized concepts of the k-msc(a) for $k = \text{rd}(Q)$, if Q us fully expanded. As soon as we have a general TBox, expanding Q may result in an infinite role-depth by expanding cyclic definitions, so this approach does not work directly here. If the msc of a w.r.t. $\mathcal{K}$ does not exist, and if any of the definitions used in Q is cyclic, we would need to compute the limit of the maximal similarity between Q and generalized concepts of k-msc(a) for $k \to \infty$.

However, one can use the correspondence of $\sim_c$ and $\sim_i$, express the concept Q by its canonical model, and express the fully expanded msc(a) in $\mathcal{EL}$ as the tree unraveling of $\mathcal{I}_{\mathcal{K}}$ starting from d_a. Thus for any concept C we have

$$\lim_{k\to\infty} C \sim_c k\text{-msc}(a) = (\mathcal{I}_{C,\mathcal{T}}{}', d_C) \sim_i (\mathcal{I}_{\mathcal{K}}{}', d_a),$$

where $\mathcal{I}_{C,\mathcal{T}}{}'$ and $\mathcal{I}_{\mathcal{K}}{}'$ are the normalized canonical models of C and $\mathcal{A}$ w.r.t. the TBox $\mathcal{T}$. The canonical model $\mathcal{I}_{\mathcal{K}}$, in contrast to the fully expanded msc, is always finite.

We do not need to compute the similarity between the query concept Q and the msc(a) directly, but find the maximal similarity between Q and generalized concepts of msc(a). Generalizing a

Procedure: maxsim $(\mathcal{I}, \mathcal{J}, \sim_{\text{prim}}, g, w)$

Input: $\mathcal{I}, \mathcal{J}$: finite interpretations; $\sim_{\text{prim}}$: primitive measure; g: weighting function; $w \in (0, 1)$: discounting factor

Output: maximal similarities between pointed interpretations $p = (\mathcal{I}, a)$ and all generalizations of the pointed interpretation $q = (\mathcal{J}, b)$

$\text{msim}_0(d, e) \leftarrow 0$ for all $d \in \Delta^{\mathcal{I}}$ and $e \in \Delta^{\mathcal{J}}$
for $i \leftarrow 1, 2, 3, \ldots$ **do**
  for all $d \in \Delta^{\mathcal{I}}$ and $e \in \Delta^{\mathcal{J}}$ **do**
    $\text{msim}_i(d, e) \leftarrow \max_{\substack{S_{\text{CN}} \subseteq \text{CN}(e) \\ S_{\text{SC}} \subseteq \text{SC}(e)}} \text{similarity}(d, S_{\text{CN}}, S_{\text{SC}}, \sim_{\text{prim}}, g, w, i)$
  end for
end for
return $\text{msim}_n(d, e)$ for all $d \in \Delta^{\mathcal{I}}$ and $e \in \Delta^{\mathcal{J}}$

Procedure: similarity $(p, S_{\text{CN}}, S_{\text{SC}}, \sim_{\text{prim}}, g, w, i)$

$\text{sim}_{\text{CN}}(p, q) = \sum_{A \in \text{CN}(p)} \max_{B \in S_{\text{CN}}} A \sim_{\text{prim}} B$
$\text{sim}_{\text{CN}}(q, p) = \sum_{B \in S_{\text{CN}}} \max_{A \in \text{CN}(p)} A \sim_{\text{prim}} B$
$\text{sim}_{\text{SC}}(p, q) = \sum_{(r,p') \in \text{SC}(p)} \max_{(s,q') \in S_{\text{SC}}} (r \sim_{\text{prim}} s)(w + (1 - w)(p' \sim_i q'))$
$\text{sim}_{\text{SC}}(q, p) = \sum_{(s,q') \in S_{\text{SC}}} \max_{(r,p') \in \text{SC}(p)} (r \sim_{\text{prim}} s)(w + (1 - w)(p' \sim_i q'))$
return $\dfrac{\text{sim}_{\text{CN}}(p, q) + \text{sim}_{\text{CN}}(q, p) + \text{sim}_{\text{SC}}(p, q) + \text{sim}_{\text{SC}}(q, p)}{|\text{CN}(p)| + |\text{CN}(q)| + |\text{SC}(p)| + |\text{SC}(q)|}$

Algorithm 2: Compute the maximal similarities w.r.t. $\sim_i$ between all elements of two finite interpretations $\mathcal{I}$ and $\mathcal{J}$.

concept C is possible by removing concept names or existential restriction, which corresponds on the interpretation side to only taking subsets of the concept names $S_{\text{CN}} \subseteq \text{CN}(q)$ and successors $S_{\text{SC}} \subseteq \text{SC}(q)$ of the pointed interpretations $q = (\mathcal{I}_{\mathcal{K}}{}', d_a)$ and all of its successors. This results in Algorithm 2 to iteratively compute the maximal similarity between a pointed interpretation p and all generalizations of the pointed interpretation q.

Note however, that the algorithm does not check all generalized concepts, since the canonical models are always finite and using the subset construction, only finitely many generalized concepts can be created, whereas the $\text{msc}_{\mathcal{K}}(a)$ may be infinite and thus can have infinitely many generalized concepts. However, to find the maximal similarity, the above subset construction is sufficient, since any infinite $\text{msc}_{\mathcal{K}}(a)$ is at some point cyclic, and thus we can reuse the same subsets for recurring elements (which correspond exactly to the same pair (p, q) of pointed interpretations).

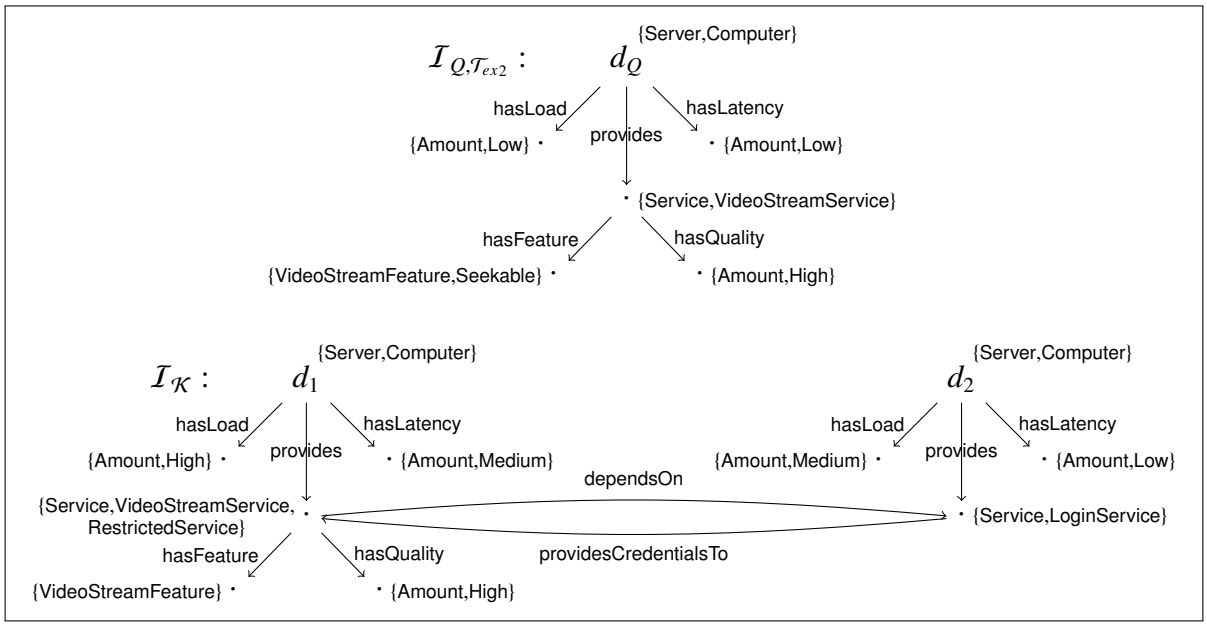

Figure 4: Canonical models of an ABox and a query concept.

Table 2: Computation steps of maxsim.

iteration i	$\text{msim}_i(d_Q, d_1)$	$\text{msim}_i(d_Q, d_2)$
0	0	0
1	0.52	0.52
2	0.856	0.877
3	0.909	0.877

Example 5.9. Consider the TBox $\mathcal{T}_{ex2}$ from Example 2.2, and the query concept

$$Q = \mathsf{Server} \sqcap \exists \mathsf{hasLatency.Low} \sqcap \mathsf{hasLoad.Low} \sqcap \exists \mathsf{provides.(VideoStreamService} \sqcap \exists \mathsf{hasQuality.High} \sqcap \exists \mathsf{hasFeature.Seekable}).$$

Also consider the ABox $\mathcal{A}$ given in Figure 4 as the canonical model $\mathcal{I}_{\mathcal{K}}$, and the canonical model $\mathcal{I}_{Q,\mathcal{T}_{ex2}}$ of the query concept: The similarity measure used is the same as in Example 5.2, with the default weighting function, the discounting factor $w = 0.8$ and the primitive measure that assigns 0.5 to Low $\sim_{\text{prim}}$ Medium and Medium $\sim_{\text{prim}}$ High and acts like the default primitive measure otherwise. Note that the service provided by d_1 is much more similar to the query concept than the one provided by d_2, while d_2 has a lower load and latency. The algorithm maxsim will compute the similarity values msim_i between d_Q and d_1, d_2 in each iteration i as shown in Table 2. Since the query concept is not cyclic and has role-depth 2, the similarity values will not change after the 3rd iteration. For the threshold $t = 0.9$, d_1 is a relaxed instance of Q, while d_2 is not.

Using this, the algorithm to actually compute all relaxed instances of a query concept Q w.r.t. $\sim_c$ is conceptually quite easy, as it only needs to compute the maximal similarities between Q and

all individuals a and check whether they are larger than t. The algorithm is depicted in Algorithm 3.

The msim_i values computed in the algorithm monotonically converge from below to the maximal similarities between generalized concepts of the most specific concept of an individual and the query concept. Thus, for any individual a, which is a relaxed instance of Q with a threshold strictly larger than t, there exists $i \in \mathbb{N}$ such that for all $j > i$ we have $\mathrm{msim}_j(Q, a) > t$. Thus, the algorithm is sound and complete in the following sense.

Theorem 5.10. *Let $\sim_c$ be the CSM derived from $\sim_i(\sim_{\mathrm{prim}}, g, w)$ with the primitive measure $\sim_{\mathrm{prim}}$, the weighting function g, and the discounting factor w. Then the algorithm* relaxed-instances *is sound and complete:*

1. Soundness*: If $a \in$ relaxed-instances$(Q, \mathcal{K}, t, \sim_{\mathrm{prim}}, g, w)$ for a number n of iterations, then $a \in \mathrm{Relax}_t^{\sim_c}(Q)$.*
2. Completeness*: If $a \in \mathrm{Relax}_t^{\sim_c}(Q)$, then there exists an $i \in \mathbb{N}$ such that for all $n \geq i$ iterations $a \in$ relaxed-instances$(Q, \mathcal{K}, t, \sim_{\mathrm{prim}}, g, w)$.*

The algorithm converges quite fast: For any iteration, the difference between the actual similarity and the computed value reduces by a factor of w. This is a direct consequence of the Banach fixed-point theorem used in the proof of Theorems 5.3 and 5.10 in the Appendix. This means that, to reduce the error tolerance of the solutions by a constant factor, e.g. one tenth, only a constant number of iterations need to be done additionally. However, one cannot compute how many iterations are needed beforehand and cannot be sure if, at any given point, the algorithm already found all relaxed instances, or if some relaxed instances with a maximal similarity very close to the threshold t are still missing.

5.3. Complexity of Relaxed Instance Queries for General TBoxes

To show an upper bound on the complexity of relaxed instance queries for general TBoxes, the iterative procedure from Algorithms 2 and 3 is not useful, since it only converges to the correct solution, but may never reach it. However, we can translate the problem into a linear optimization problem (i.e., a system of linear inequalities and a linear objective function). We first show how this approach can be used to prove that the similarity measure $\sim_i$ can be computed for all elements in the domains of two interpretations $\mathcal{I}$ and $\mathcal{J}$ in polynomial time.

Procedure: relaxed-instances $(Q, \mathcal{K}, t, \sim_{\mathrm{prim}}, g, w)$
Input: Q: $\mathcal{EL}$-concept; $\mathcal{K} = (\mathcal{T}, \mathcal{A})$: $\mathcal{EL}$-KB; $t \in [0, 1]$: threshold; $\sim_{\mathrm{prim}}$: primitive measure; g: weighting function; $w \in (0, 1)$: discounting factor
Output: individuals $a \in Relax_t^{\sim_c}(Q)$

1: compute canonical models $\mathcal{I}_{Q,\mathcal{T}}$ and $\mathcal{I}_{\mathcal{K}}$
2: $\mathrm{maxsim}(d, e) \leftarrow$ maxsim$(\mathcal{I}_{Q,\mathcal{T}}, \mathcal{I}_{\mathcal{K}}, \sim_{\mathrm{prim}}, g, w)$
3: **return** $\{a \in N_I \cap \mathrm{Sig}(\mathcal{A}) \mid \mathrm{maxsim}(d_Q, d_a) > t\}$

Algorithm 3: Computation of relaxed instances of query concept Q w.r.t. KB $\mathcal{K}$ and threshold t.

Theorem 5.11. *The similarities $(\mathcal{I}, d) \sim_i (\mathcal{J}, e)$ for all $d \in \Delta^{\mathcal{I}}$ and $e \in \Delta^{\mathcal{J}}$ can be computed in polynomial time in the size of the interpretations $\mathcal{I}$ and $\mathcal{J}$.*

Proof sketch. For each $p = (\mathcal{I}, d)$ and $q = (\mathcal{J}, e)$, we treat the similarity value $p \sim q$ as a variable $V_{p,q}$. We further introduce variables $V_{s_1,X}$ and V_{X,s_2} for the maximum similarity between s_1 and one of successors in the set X, and between one of the successors in X and s_2, respectively:

$$V_{p,q} \geq \frac{\mathrm{sim}_{\mathrm{CN}}(p,q) + \mathrm{sim}_{\mathrm{CN}}(q,p) + \sum_{s_1 \in \mathrm{SC}(p)} V_{s_1,\mathrm{SC}(q)} + \sum_{s_2 \in \mathrm{SC}(q)} V_{\mathrm{SC}(p),s_2}}{|\mathrm{CN}(p)| + |\mathrm{CN}(q)| + |\mathrm{SC}(p)| + |\mathrm{SC}(q)|} \tag{2}$$

and for each variable $V_{s_1,\mathrm{SC}(q)}$ or $V_{\mathrm{SC}(p),s_2}$ introduced above:

$$V_{s_1,\mathrm{SC}(q)} \geq (r_1 \sim_p r_2)(w + (1-w)V_{p_1,p_2}) \qquad \text{for } s_1 = (r_1, p_1) \text{ and all } (r_2, p_2) \in \mathrm{SC}(q)$$
$$V_{\mathrm{SC}(p),s_2} \geq (r_1 \sim_p r_2)(w + (1-w)V_{p_1,p_2}) \qquad \text{for } s_2 = (r_2, p_2) \text{ and all } (r_1, p_1) \in \mathrm{SC}(p)$$

If we translate every equation $(\mathcal{I}, d) \sim (\mathcal{J}, e)$ to linear inequalities for all elements d, e of $\mathcal{I}$ and $\mathcal{J}$ as shown above, then a linear optimization with the aim to maximize the objective function $-\sum_{\substack{d \in \Delta^{\mathcal{I}} \\ e \in \Delta^{\mathcal{J}}}} V_{(\mathcal{I},d),(\mathcal{J},e)}$ will return the exact similarities between each pair of elements of the interpretations $\mathcal{I}$ and $\mathcal{J}$.

The reason why this works is that the original equation system defined in Equation (1) has only one unique solution (see Theorem 5.3), and thus for all solutions with at least one value $V_{p,q} < p \sim_i q$, one of the equations (and thus also one of the inequalities in the linear optimization problem) is not satisfied. Then, the only optimal solution $\overrightarrow{V_{p,q}}$ of the linear optimization problem must be the vector of solutions $p \sim_i q$ of the original equation system (1).

The optimization problem will always have a size that is polynomial in the size of the interpretations $\mathcal{I}$ and $\mathcal{J}$. Since the linear optimization problem can be solved in polynomial time, this finishes the proof. □

Since the canonical models $\mathcal{I}_{\mathcal{Q},\mathcal{T}}$ and $\mathcal{I}_{\mathcal{K}}$ are always polynomial in the size of $\mathcal{K}$ and Q (and can be computed in polynomial time), and the normalization can be computed in polynomial time and will never increase the size of the model, this implies that also $\sim_c$ can be computed in polynomial time.

To compute the maximal similarities, we have to find the best subsets of the concept names and successors for each element in the interpretation $\mathcal{I}_{\mathcal{K}}$; however, in the worst case, the number of concept names or successors may be linear in the size of $\mathcal{K}$ and Q – but then the number of subsets is exponential. Instead we can guess a subset of $\mathrm{CN}(q)$ and $\mathrm{SC}(q)$ for each pair p, q of pointed interpretations, and use those subsets in the definition of the linear optimization problem. In fact, we do not necessarily have to find those subsets that yield the maximum similarity value, since we only have to check if the similarity values are larger than t or not. To verify that one guess yields a similarity value larger than t, we can simply solve the linear optimization problem, which is possible in polynomial time. Thus, we have the following complexity for computing relaxed instances.

Corollary 5.12. *Relaxed instances of a query concept Q w.r.t. $\sim_c$ and a general knowledge base $\mathcal{K}$ is in NP.*

6. Conclusions

In this paper we have introduced a new reasoning service for DLs that allows to relax instance queries by means of concept similarity measures. By choosing appropriate similarity measures, this allows for domain- and context-dependent relaxation of the query. For example, it is possible to alter the weights, that the different features of the concept have in the final similarity value. This allows to put more emphasis on important features, which are not to be relaxed in contrast to less important features. Besides the choice of a suitable CSM, this method also allows to change the degree of relaxation by specifying a threshold t.

We explored two methods for computing relaxed instances in the description logic $\mathcal{EL}$. The first method works for arbitrary CSMs, that are equivalence invariant and successor-closed. Our method for computing relaxed instance by use of these CSMs works only for terminologies, i.e., unfoldable $\mathcal{EL}$-TBoxes. In this case we can simply expand the query concept Q w.r.t. the TBox, and check for each individual a in the ABox if it is a relaxed instance of Q by computing its k-msc for $k = \mathrm{rd}(Q)$ and computing the similarity of all its generalized concepts to Q: if the maximal similarity of those is larger than the threshold, a is a relaxed instance.

However, this method based on expansion does not work for general $\mathcal{EL}$-TBoxes due to cyclic concepts. By introducing a new family of CSMs $\sim_c$ for $\mathcal{EL}$-concepts defined w.r.t. general TBoxes and restricting to those, we are able to solve this problem. Moreover, this allows to avoid to check all generalized concepts (of which there are exponentially many even in case of terminologies). Those new CSMs depend on the definition of similarity measures for pointed interpretations, which get lifted to concept similarity via canonical models. That way, the CSM $\sim_c$ has many desirable formal properties, works w.r.t. general $\mathcal{EL}$-TBoxes, and can be adapted to many different situations and domains using its parameters: a primitive similarity between concept names and role names, a weighting function that weights the importance of each concept or role name, and a discounting factor. To the best of our knowledge, the CSMs $\sim_c$ are the first CSMs that incorporate all available knowledge from general TBoxes.

The computation algorithms for relaxed instances w.r.t. $\sim_c$ works iteratively by refining similarity values, which monotonically converge to the final similarity value. This algorithm can be easily adapted to compute relaxed instances of a query concept Q w.r.t. a knowledge base $\mathcal{K}$, by computing the maximal similarities of Q to generalized concepts of the $\mathrm{msc}(a)$ for all individuals a at once. This yields an efficient solution to relaxed instance queries. When applied to unfoldable TBoxes, it is possible for this algorithm to bound the number of iterations, after which the exact maximal similarities are found.

There are many options for future work. On the theoretical side it would be interesting to explore how this approach can be extended to more expressive DLs. In [34], we showed that this is possible for the DL $\mathcal{EL}++$, which extends $\mathcal{EL}$ by nominals, domain and range restrictions, and concrete domains. Since even more expressive Horn-DLs induce finite canonical models as well, we conjecture that our approach also works for those. How to generalize our approach and the computation of relaxed instances to DLs that offer all Boolean operators is not obvious.

Similarly, it would be interesting to extend the query language, for example to conjunctive queries instead of instance queries. However, the similarity measure itself is only defined for (essentially tree-shaped and rooted) concepts and not for arbitrary query graphs. Therefore, an

extension to conjunctive queries would also require to extent the notion of similarity measures to queries.

On the practical side there is plenty of room for optimizations. For instance, the use of a concept that states necessary conditions in combination with the query concept can considerably reduce the number of individuals to be checked in practice. Furthermore, while the complexity of each iteration in the general case is polynomial, the need to check all subsets is certainly inefficient. Methods to reduce the candidate subsets that need to be considered are expedient to make this work in practice, where ABoxes are typically large.

It is also interesting to consider other applications of similarity measures, in particular, not just using similarity to define new reasoning services, but to integrate them directly into the knowledge base. This might be possible by introducing a new concept constructor of the form $S_{>t}p$, which is interpreted as all elements of the domain, that have a similarity of at least t to some fixed pointed interpretation p. This approach would allow to define and reason over prototypes of concepts. The requirements for CSMs that allow to reason over concepts and KBs written in such extended DL is also future work.

Acknowledgements

A. Ecke is supported by the German Research Foundation (DFG) in the Graduiertenkolleg 1763 (QuantLA). R. Peñaloza is partially supported by DFG within the Cluster of Excellence 'Center for Advancing Electronics Dresden' (cfAED). A.-Y. Turhan is partially supported by DFG in the Collaborative Research Center 912 'Highly Adaptive Energy-Efficient Computing' (HAEC).

References

[1] B. Motik, R. Shearer, I. Horrocks, Hypertableau Reasoning for Description Logics, Journal of Artificial Intelligence Research 36 (2009) 165–228.

[2] J. Mendez, A. Ecke, A.-Y. Turhan, Implementing completion-based inferences for the $\mathcal{EL}$-family, in: R. Rosati, S. Rudolph, M. Zakharyaschev (Eds.), Proceedings of the 24th International Workshop on Description Logics (DL 2011), Vol. 745 of CEUR Workshop Proceedings, CEUR-WS.org, 2011.

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

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

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

[6] S. Borgwardt, F. Distel, R. Peñaloza, How fuzzy is my fuzzy description logic?, in: B. Gramlich, D. Miller, U. Sattler (Eds.), Proc. of the 6th Int. Joint Conf. on Automated Reasoning (IJCAR-12), Vol. 7364 of LNAI, Springer-Verlag, 2012, pp. 82–96.

[7] S. Borgwardt, R. Peñaloza, Undecidability of fuzzy description logics, in: G. Brewka, T. Eiter, S. A. McIlraith (Eds.), Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-12), AAAI Press, 2012, pp. 232–242.
URL `http://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4387`

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

[9] T. G. O. Consortium, Gene Ontology: Tool for the unification of biology, Nature Genetics 25 (2000) 25–29.

[10] P. W. Lord, R. D. Stevens, A. Brass, C. A. Goble, Investigating semantic similarity measures across the gene ontology: The relationship between sequence and annotation, Bioinformatics 19 (10) (2003) 1275–1283.
[11] C. Pesquita, CESSM: collaborative evaluation of GO-based semantic similarity measures, Challenges in Bioinformatics (JB2009).
[12] J. Euzenat, P. Valtchev, Similarity-based ontology alignment in OWL-lite, in: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-04), IOS Press, 2004, pp. 333–337.
[13] A. Borgida, T. Walsh, H. Hirsh, Towards measuring similarity in description logics., in: Proc. of the 2005 Description Logic Workshop (DL 2005), Vol. 147 of CEUR Workshop Proceedings, 2005.
[14] C. d'Amato, N. Fanizzi, F. Esposito, A semantic similarity measure for expressive description logics, in: Proc. of Convegno Italiano di Logica Computazionale, CILC05, 2005.
[15] K. Lehmann, A.-Y. Turhan, A framework for semantic-based similarity measures for $\mathcal{ELH}$-concepts, in: L. F. del Cerro, A. Herzig, J. Mengin (Eds.), Proc. of the 13th European Conf. on Logics in A.I. (JELIA 2012), LNAI, Springer, 2012, pp. 307–319.
[16] B. Suntisrivaraporn, A similarity measure for the description logic $\mathcal{EL}$ with unfoldable terminologies, in: 5th International Conference on Intelligent Networking and Collaborative Systems (INCoS), 2013, pp. 408–413. doi:10.1109/INCoS.2013.77.
[17] F. Baader, S. Brandt, C. Lutz, Pushing the $\mathcal{EL}$ envelope, in: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05), Morgan-Kaufmann Publishers, Edinburgh, UK, 2005.
[18] K. Spackman, Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with snomed-rt, Journal of the American Medical Informatics Assoc.Fall Symposium Special Issue.
[19] A. Ecke, R. Peñaloza, A.-Y. Turhan, Towards instance query answering for concepts relaxed by similarity measures, in: L. Godo, H. Prade, G. Qi (Eds.), Workshop on Weighted Logics for AI (in conjunction with IJCAI'13), Beijing, China, 2013.
[20] A. Ecke, R. Peñaloza, A.-Y. Turhan, Answering instance queries relaxed by concept similarity, in: Proceedings of the Fourteenth International Conference on Principles of Knowledge Representation and Reasoning (KR'14), AAAI Press, Vienna, Austria, 2014.
[21] C. d'Amato, S. Staab, N. Fanizzi, On the influence of description logics ontologies on conceptual similarity, in: A. Gangemi, J. Euzenat (Eds.), Proceedings of Knowledge Engineering: Practice and Patterns, 16th Int. Conf. (EKAW 2008), Vol. 5268 of LNCS, Springer, 2008, pp. 48–63.
[22] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.
[23] F. Baader, Description logics, in: Proceedings of Reasoning Web: Semantic Technologies for Information Systems, Vol. 5689 of LNCS, 2009, pp. 1–39.
[24] C. Lutz, F. Wolter, Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$, Journal of Symbolic Computation 45 (2) (2010) 194–228. doi:10.1016/j.jsc.2008.10.007.
[25] F. Baader, Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles, in: G. Gottlob, T. Walsh (Eds.), Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI-03), Morgan Kaufmann, 2003, pp. 325–330.
[26] R. Peñaloza, A.-Y. Turhan, A practical approach for computing generalization inferences in $\mathcal{EL}$, in: M. Grobelnik, E. Simperl (Eds.), Proceedings of the 8th European Semantic Web Conference (ESWC'11), LNCS, Springer, 2011.
[27] A. Ecke, R. Peñaloza, A.-Y. Turhan, Computing role-depth bounded generalizations in the description logic $\mathcal{ELOR}$, in: I. J. Timm, M. Thimm (Eds.), Proceedings of the 36th German Conference on Artificial Intelligence (KI 2013), Vol. 8077 of LNCS, Springer, Koblenz, Germany, 2013, pp. 49–60, extended version: `http://lat.inf.tu-dresden.de/research/papers/2013/EcPeTu-KI-13.long.pdf`.
[28] C. Lutz, F. Wolter, M. Zakharyaschev, Reasoning about concepts and similarity, in: Proceedings of the 2003 International Workshop on Description Logics (DL2003), CEUR-WS, 2003.
[29] U. Straccia, Towards top-k query answering in description logics: The case of dl-lite, in: Proc. of the 10th European Conf. on Logics in A.I. (JELIA 2006), Vol. 4160 of LNCS, Springer, 2006, pp. 439–451.
[30] U. Straccia, Answering vague queries in fuzzy dl-lite, in: Proc. of the 11th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU-06), 2006, pp. 2238–2245.

[31] J. Z. Pan, G. B. Stamou, G. Stoilos, S. Taylor, E. Thomas, Scalable querying services over fuzzy ontologies, in: Proc. of the 17th Int. Conf. on World Wide Web (WWW'08), ACM, 2008, pp. 575–584.
[32] R. Peñaloza, V. Thost, A.-Y. Turhan, Conjunctive query answering in rough $\mathcal{EL}$, LTCS-Report 14-04, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany, see `http://lat.inf.tu-dresden.de/research/reports.html`. (2014).
[33] S. Banach, Sur les oprations dans les ensembles abstraits et leur application aux quations intgrales, Fundamenta Mathematicae 3 (1) (1922) 133–181.
[34] A. Ecke, Similarity-based relaxed instance queries in $\mathcal{EL}^{++}$, in: T. Lukasiewicz, R. Peñaloza, A.-Y. Turhan (Eds.), Proceedings of the First Workshop on Logics for Reasoning about Preferences, Uncertainty, and Vagueness, CEUR-WS, CEUR, 2014, to appear.
[35] B. Zarrieß, A.-Y. Turhan, Most Specific Generalizations w.r.t. General $\mathcal{EL}$-TBoxes, in: F. Rossi (Ed.), Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13), 2013.

Appendix A. Proofs for Theorems from Section 5

Appendix A.1. Properties of $\sim_i$

Before showing the formal properties for $\sim_i$, we need to show that the result of normalization is unique.

Lemma Appendix A.1. *Let $(\mathcal{I}, a)$ and $(\mathcal{J}, b)$ be two pointed interpretations and let $\mathcal{I}'$ and $\mathcal{J}'$ be the results of normalizing $\mathcal{I}$ and $\mathcal{J}$, respectively. Then the following holds:*

1. *Normalization preserves simulations, i.e., if $(\mathcal{I}, a) \lesssim (\mathcal{J}, b)$ then also $(\mathcal{I}', a) \lesssim (\mathcal{J}', b)$.*
2. *If $(\mathcal{I}, a) \simeq (\mathcal{J}, b)$, then for any successor $(r, p) \in \mathrm{SC}((\mathcal{I}', a))$ there exists a unique successor $(r, q) \in \mathrm{SC}((\mathcal{J}', b))$ with $p \simeq q$ and vice versa. We denote this property by saying that $(\mathcal{I}', a)$ and $(\mathcal{J}', b)$ are* structurally equivalent.

Proof.

1. Let $(\mathcal{I}, a)$ and $(\mathcal{J}, b)$ be two pointed interpretations with $(\mathcal{I}, a) \lesssim (\mathcal{J}, b)$. Then for each concept name A, we have $a \in A^{\mathcal{I}'} \Leftrightarrow a \in A^{\mathcal{I}} \Rightarrow b \in A^{\mathcal{J}} \Leftrightarrow b \in A^{\mathcal{J}'}$. Additionally, for each role name r, we have $(a, a') \in r^{\mathcal{I}'} \Rightarrow (a, a') \in r^{\mathcal{I}} \Rightarrow \exists b' : (b, b') \in r^{\mathcal{J}} \wedge (\mathcal{I}, a') \lesssim (\mathcal{J}, b')$. If $(b, b') \in r^{\mathcal{J}'}$, we are done: $(\mathcal{I}', a) \lesssim (\mathcal{J}', b)$ follows directly.

   Otherwise, we know by the construction of $\mathcal{J}'$, that there exists an element $c \in \Delta^{\mathcal{J}'}$ with $(b, c) \in r^{\mathcal{J}'}$ and $(\mathcal{J}', b') \lesssim (\mathcal{J}', c)$ or $(\mathcal{J}', b') \simeq (\mathcal{J}', c)$. Since $\lesssim$ is transitive and $(\mathcal{I}', a') \lesssim (\mathcal{I}, a) \lesssim (\mathcal{J}, c)$, this means that $(\mathcal{I}', a') \lesssim (\mathcal{J}', c)$ and the claim, $(\mathcal{I}', a) \lesssim (\mathcal{J}', b)$ again follows.
2. Let $(\mathcal{I}, a)$ and $(\mathcal{J}, b)$ be two pointed interpretations with $(\mathcal{I}, a) \simeq (\mathcal{J}, b)$. Let further $(a, c) \in r^{\mathcal{I}'}$, which also implies $(a, c) \in r^{\mathcal{I}}$. Since $\mathcal{I}'$ is in normal form, this means that there is no $c' \in \Delta^{\mathcal{I}}$ with $(a, c') \in r^{\mathcal{I}}$ and $(\mathcal{I}, c) \lesssim (\mathcal{I}, c')$, and $(\mathcal{I}, c') \not\lesssim (\mathcal{I}, c)$. Since $(\mathcal{I}, a) \simeq (\mathcal{J}, b)$, there exists an element $d \in \Delta^{\mathcal{J}}$ with $(b, d) \in r^{\mathcal{J}}$ and $(\mathcal{I}, c) \lesssim (\mathcal{J}, d)$, but not necessarily $(b, d) \in r^{\mathcal{J}'}$. By the construction of $\mathcal{J}'$, we know that there is an element $e \in \Delta^{\mathcal{J}'}$ with $(b, e) \in r^{\mathcal{J}'}$ and $(\mathcal{J}, d) \lesssim (\mathcal{J}, e)$. Again, $(\mathcal{I}, a) \simeq (\mathcal{J}, b)$ implies that a must have a successor $(a, f) \in r^{\mathcal{I}}$ with $(\mathcal{J}, e) \lesssim (\mathcal{I}, f)$; however, since with $(\mathcal{I}, c) \lesssim (\mathcal{J}, d)$ and $(\mathcal{J}, d) \lesssim (\mathcal{J}, e)$, this also means $(\mathcal{I}, c) \lesssim (\mathcal{I}, f)$. Since we assumed that there is no $c' \in \Delta^{\mathcal{I}}$ with $(a, c') \in r^{\mathcal{I}}$ and $(\mathcal{I}, c) \lesssim (\mathcal{I}, c')$, this means that $f = c$ and thus $(\mathcal{I}, c) \simeq (\mathcal{J}, e)$ and by point 1. also $(\mathcal{I}', c) \simeq (\mathcal{J}', e)$. The other direction is analogous. □

With this, we can finally show the formal properties of $\sim_i$.

Theorem 5.5. *Let $\sim_i(\sim_{\text{prim}}, g, w)$ be instantiated with a primitive measure $\sim_{\text{prim}}$, a weighting function g, and constant $w \in (0,1)$. Then $\sim_i$ has the following properties (w.r.t. to the simulation relations $\lesssim$ and $\simeq$ given in Definition 2.3):*

1. *$\sim_i$ is symmetric, if the primitive measure $\sim_{\text{prim}}$ is symmetric;*
2. *$\sim_i$ is bounded;*
3. *$\sim_i$ is dissimilar closed, if the primitive measure $\sim_{\text{prim}}$ does not assign a similarity value greater than 0 to different concept or role names.*
4. *$\sim_i$ is equisimulation invariant for normalized interpretations; and*
5. *$\sim_i$ is equisimulation closed for normalized interpretations, if the primitive measure $\sim_{\text{prim}}$ does not assign the similarity value 1 to different concept or role names.*

Proof.

1. *symmetric:* $\sim_i$ is symmetric, if the primitive measure $\sim_{\text{prim}}$ is symmetric, as the definition of $\sim_i$ only uses commutative operators.
2. *bounded:* $\sim_i$ is bounded, if $\mathfrak{C}(p) \cap \mathfrak{C}(q) \supset \{\top\}$ implies $p \sim_i q > 0$ for all $p, q \in \mathfrak{P}$. Assume that there exists a concept $C \neq \top$ in $\mathfrak{C}(p) \cap \mathfrak{C}(q)$. Then, there also exists either a concept name A or an existential restriction of the form $\exists r.\top$ in $\mathfrak{C}(p) \cap \mathfrak{C}(q)$, since for all conjunctions $C_1 \sqcap C_2 \in \mathfrak{C}(p) \cap \mathfrak{C}(q)$ we also have $C_1, C_2 \in \mathfrak{C}(p) \cap \mathfrak{C}(q)$ and for all $\exists r.C \in \mathfrak{C}(p) \cap \mathfrak{C}(q)$ we also have $\exists r.\top \in \mathfrak{C}(p) \cap \mathfrak{C}(q)$.

   However, for a concept name $A \in \mathfrak{C}(p) \cap \mathfrak{C}(q)$, we have that $A \sim_{\text{prim}} A = 1$ and thus $\sum_{A \in \text{CN}(p)} \max_{B \in \text{CN}(q)} g(A,B)(A \sim_{\text{prim}} B) > 0$. This yields $p \sim_i q > 0$. Correspondingly, for $\exists r.\top \in \mathfrak{C}(p) \cap \mathfrak{C}(q)$, we have $r \sim_{\text{prim}} r = 1$ and thus $g(r,s)(r \sim_{\text{prim}} r)((1-w) + w(p' \sim_i q')) > 1 - w > 0$ and $\sum_{(s,q') \in \text{SC}(p)} \max_{(p,q) \in \text{SC}(q)} g(r,s)(r \sim_{\text{prim}} s)((1-w) + w(p' \sim_n q')) > 0$. Again, this yields $p \sim_i q > 0$.
3. *dissimilar closed:* $\sim_i$ is dissimilar closed, if $\mathfrak{C}(p) \cap \mathfrak{C}(q) = \{\top\}$ implies $p \sim_i q = 0$ for all $p, q \in \mathfrak{P}$ with $\mathfrak{C}(p) \supset \{\top\}$ and $\mathfrak{C}(q) \supset \{\top\}$; of course, $\sim_i$ can only be dissimilarity closed if the primitive measure does not assign a similarity value greater than 0 to different concept or role names. Hence we only show this property for the default primitive measure $\sim_{\text{default}}$.

   Let $p, q \in \mathfrak{P}$ with $\mathfrak{C}(p) \supset \{\top\}$ and $\mathfrak{C}(q) \supset \{\top\}$, i.e., both p and q are instance of some concept name or have a successor. If $\mathfrak{C}(p) \cap \mathfrak{C}(q) = \{\top\}$, then $A \sim_{\text{default}} B = 0$ for all $A \in \text{CN}(p)$ and $B \in \text{CN}(q)$. Similarly, as there is no role name r with $(r, p') \in \text{SC}(p)$ and $(r, q') \in \text{SC}(q)$, we have $r \sim_{\text{default}} s = 0$ for all $(r, p') \in S(p)$ and $(s, q') \in S(q)$. This then yields $p \sim_i q = 0$.
4. *equisimulation invariant:* $\sim_i$ is equisimulation invariant for normalized interpretations, if $p \simeq q$ implies $p \sim_i u = q \sim_i u$ for all normalized pointed interpretations $p, q, u \in \mathfrak{P}$; it is a direct consequence of the fact that if $p \simeq q$, then the normalized pointed interpretations do not just simulate each other, but are structurally equivalent, as stated in Point 2 in Lemma Appendix A.1. Thus the computation of $p \sim_i u$ can be modified to compute $q \sim_i u$

by simply replacing the successors of p by the unique equisimilar successors of q and vice versa; this will always yield the same similarity value.

5. *equisimulation closed:* The direction from left to right, i.e., $p \simeq q$ implies $p \sim_i q = 1$, follows again by Point 2 in Lemma Appendix A.1. For the other direction, that $p \sim_i q = 1$ also implies $p \simeq q$, we need the property that the primitive measure does not assign a similarity value of 1 to different concept or role names. In this case, assume that $p \not\simeq q$ for $p = (\mathcal{I}, a)$ and $q = (\mathcal{J}, b)$. Then, w.l.o.g., we have one of the following conditions:
   (a) there exists a concept name A with $a \in A^{\mathcal{I}}$ and $b \notin A^{\mathcal{J}}$, or
   (b) a has a successor $(a, c) \in r^{\mathcal{I}}$ and there is no d with $(b, d) \in r^{\mathcal{J}}$, or
   (c) a has a successor $(a, c) \in r^{\mathcal{I}}$ and for all successors $t = (\mathcal{J}, d)$ of b with $(b, d) \in r^{\mathcal{J}}$ we have that $s \not\simeq t$. In this case, there must be a finite chain of such successors s_i, t_i starting from a, b such that condition 1 or 2 holds for s_n, t_n.

   Now, we can prove inductively that $p \sim_i q < 1$. In the first two cases a) and b), Equation 1 directly gives a similarity value < 1, since the concept name A in case a) or the role name r in case b) will always be matched with a different concept or role name and $\sim_{\text{prim}}$ never assigns similarity 1 to different concept or role names. In the third case, we assume that $c \sim_i d < 1$ by induction for all successors d of b. Then Equation 1 again yields a similarity value $p \sim_i q < 1$. Thus $\sim_i$ must equisimulation closed. □

Appendix A.2. Properties of $\sim_c$

Theorem 5.7. *For a primitive measure $\sim_{\text{prim}}$, a weighting function g, and a discounting factor w, the concept similarity measure $\sim_c(\sim_{\text{prim}}, g, w)$ is symmetric, bounded, dissimilar closed, equivalence invariant, and equivalence closed, if $\sim_i(\sim_{\text{prim}}, g, w)$ is symmetric, bounded dissimilar closed, equisimulation invariant and equisimulation closed, respectively.*

Proof. We prove that the properties of $\sim_i$ transfer to $\sim_c$:

1. symmetry: $C \sim_c D = (\mathcal{I}_{C,\mathcal{T}}, d_C) \sim_i (\mathcal{I}_{D,\mathcal{T}}, d_D) = (\mathcal{I}_{D,\mathcal{T}}, d_D) \sim_i (\mathcal{I}_{C,\mathcal{T}}, d_C) = D \sim_c C$ follows from the symmetry of $\sim_i$.
2. bounded: Assume that for two $\mathcal{EL}$-concept C and D, there exists a concept $E \neq \top$ with $C \sqsubseteq_{\mathcal{T}} E$ and $D \sqsubseteq_{\mathcal{T}} E$. Then Theorem 2.6 and Lemma Appendix A.1 yield $E \in \mathfrak{C}(p) \cap \mathfrak{C}(q)$ for $p = (\mathcal{I}'_{C,\mathcal{T}}, d_C)$ and $q = (\mathcal{I}'_{D,\mathcal{T}}, d_D)$. Therefore boundedness of $\sim_i$ implies $C \sim_c D = p \sim_i q > 0$.
3. dissimilar closed: Assume that for two $\mathcal{EL}$-concept $C, D \neq \top$, there is no concept $E \neq \top$ with $C \sqsubseteq_{\mathcal{T}} E$ and $D \sqsubseteq_{\mathcal{T}} E$. Then Theorem 2.6 and Lemma Appendix A.1 imply that $\mathfrak{C}(p) \cap \mathfrak{C}(q) = \{\top\}$ for $p = (\mathcal{I}'_{C,\mathcal{T}}, d_C)$ and $q = (\mathcal{I}'_{D,\mathcal{T}}, d_D)$, and thus, since we assume that $\sim_i$ is dissimilar closed, $C \sim_c D = p \sim_i q = 0$.
4. equivalence invariant: Assume that $C \equiv_{\mathcal{T}} D$. Then by Theorem 2.6 and Lemma Appendix A.1 we have $(\mathcal{I}'_{C,\mathcal{T}}, d_C) \simeq (\mathcal{I}'_{D,\mathcal{T}}, d_D)$ and thus equisimulation invariance of $\sim_i$ implies $(\mathcal{I}'_{C,\mathcal{T}}, d_C) \sim_i (\mathcal{J}, e) = (\mathcal{I}'_{D,\mathcal{T}}, d_D) \sim_i (\mathcal{J}, e)$ for any pointed interpretation $(\mathcal{J}, e)$, in particular pointed interpretations of the form $(\mathcal{I}'_{E,\mathcal{T}}, d_E)$. This then yields $C \sim_c E = D \sim_c E$ for any $\mathcal{EL}$-concept E.

5. equivalence closed: Assume that $C \equiv_{\mathcal{T}} D$. Then by Theorem 2.6 and Lemma Appendix A.1 we have $(\mathcal{I}'_{C,\mathcal{T}}, d_C) \simeq (\mathcal{I}'_{D,\mathcal{T}}, d_D)$ and thus $(\mathcal{I}'_{C,\mathcal{T}}, d_C) \sim_i (\mathcal{I}'_{D,\mathcal{T}}, d_D) = 1$ since $\sim_i$ is equisimulation closed. But then we also have $C \sim_c D = 1$.

   Similarly, assume that $C \sim_c D = (\mathcal{I}'_{C,\mathcal{T}}, d_C) \sim_i (\mathcal{I}'_{D,\mathcal{T}}, d_D) = 1$. Then $(\mathcal{I}'_{C,\mathcal{T}}, d_C) \simeq (\mathcal{I}'_{D,\mathcal{T}}, d_D)$ since $\sim_i$ is equisimulation closed, and thus Theorem 2.6 yields $C \equiv_{\mathcal{T}} D$. □

Appendix A.3. Correctness of Algorithm relaxed-instances *from Figure 2*

Theorem 5.10. *Let $\sim_c$ be the CSM derived from $\sim_i(\sim_{\text{prim}}, g, w)$ with the primitive measure $\sim_{\text{prim}}$, the weighting function g, and the discounting factor w. Then the algorithm* relaxed-instances *is sound and complete:*

1. Soundness*: If $a \in$ relaxed-instances$(Q, \mathcal{K}, t, \sim_{\text{prim}}, g, w)$ for a number n of iterations, then $a \in \text{Relax}_t^{\sim_c}(Q)$.*
2. Completeness*: If $a \in \text{Relax}_t^{\sim_c}(Q)$, then there exists an $i \in \mathbb{N}$ such that for all $n \geq i$ iterations $a \in$ relaxed-instances$(Q, \mathcal{K}, t, \sim_{\text{prim}}, g, w)$.*

Proof. First, we show that the fixed-point of msim_i for $(\mathcal{I}'_{Q,\mathcal{T}}, d_Q)$ and $(\mathcal{I}'_{\mathcal{K}}, d_a)$ with $i \to \infty$ corresponds to the maximal similarity between Q and all concepts D that have a as an instance. This is due to the fact that all concepts D that have a as an instance must be equivalent to generalized concepts of the (possibly infinite) fully expanded $\text{msc}_{\mathcal{K}}(a)$ (see [19]) and that the tree unraveling of $(\mathcal{I}_{\mathcal{K}}, d_a)$ yields exactly the fully expanded $\text{msc}_{\mathcal{K}}(a)$ [26, 35]. By choosing the subsets $S_{\text{CN}} \subseteq \text{CN}(q)$ and $S_{\text{SC}} \subseteq \text{SC}(q)$ for each pair of pointed interpretations $p = (\mathcal{I}_{Q,\mathcal{T}}, d)$ and $q = (\mathcal{I}_{\mathcal{K}}, e)$, the algorithm maximizes the similarity over those generalized concepts, and thus, always computes the maximal similarity between Q and all concepts D that have a as an instance.

1. Soundness: relaxed-instances computes the similarities between all $d \in \mathcal{I}_{Q,\mathcal{T}}$ and $e \in \mathcal{I}_{\mathcal{K}}$ iteratively. Again, this mapping from old to new msim values done in each iteration (line 3–5) is a contraction mapping, and therefore we can apply the Banach fixed-point theorem. This yields that the similarity values computed by relaxed-instances converge to the solutions of $\sim_{\text{imax}}$ and thus for the pair (d_Q, d_a) to the maximal similarity between Q and all concepts D that have a as an instance. Furthermore, all factors used in updating the similarity values are positive, thus the mapping is monotone, and since relaxed-instances starts with similarity value 0 for all pairs of elements, the values for (d_Q, d_a) converges to the solution from below. This means that whenever relaxed-instances finds a value $\text{msim}_i(d_Q, d_a) > t$ for some i, we know that also $(\mathcal{I}_{Q,\mathcal{T}}, d_Q) \sim_{\text{imax}} (\mathcal{I}_{\mathcal{K}}, d_a) > t$ and thus $a \in \text{Relax}_t^{\sim_c}(Q)$. The claim follows.
2. Completeness: Let a be a relaxed instance of Q w.r.t. $\sim_c$, $\mathcal{K}$ and t, i.e. $(\mathcal{I}_{Q,\mathcal{T}}, d_Q) \sim_i (\mathcal{I}_{\mathcal{K}}, d_a) - t = \delta > 0$. The convergence of the similarities computed during relaxed-instances by the Banach fixed-point theorem means that there is an $n \in \mathbb{N}$ such that the error in the value $\text{msim}_i(d_Q, d_a)$ for all iterations $i \geq n$ is less than δ; and thus greater than t, which yields that $a \in$ relaxed-instances$(Q, \mathcal{K}, t, \sim_{\text{prim}}, g, w)$ when run for $i \geq n$ iterations. □