Proc. 23rd Int. Workshop on Description Logics (DL2010), CEUR-WS 573, Waterloo, Canada, 2010.

Role-depth Bounded Least Common Subsumers by Completion for $\mathcal{EL}$- and Prob-$\mathcal{EL}$-TBoxes

Rafael Peñaloza and Anni-Yasmin Turhan

TU Dresden, Institute for Theoretical Computer Science

Abstract. The least common subsumer (lcs) w.r.t general $\mathcal{EL}$-TBoxes does not need to exists in general due to cyclic axioms. In this paper we present an algorithm for computing role-depth bounded $\mathcal{EL}$-lcs based on the completion algorithm for $\mathcal{EL}$. We extend this computation algorithm to a recently introduced probabilistic variant of $\mathcal{EL}$: Prob-$\mathcal{EL}^{01}$.

1 Introduction

The least common subsumer (lcs) inference yields a concept description, that generalizes a collection of concepts by extracting their commonalities. This inference was introduced in [8]. Most prominently the lcs is used in the bottom-up construction of knowledge bases [5], where a collection of individuals is selected for which a new concept definition is to be introduced in the TBox. This is can be achieved by first generalizing each selected individual into a concept description (by computing the most specific concept) and then applying the lcs to these concept descriptions. Further applications of the lcs include similarity-based Information Retrieval or learning from examples.

The lightweight Description Logic $\mathcal{EL}$ and many of its extensions enjoy the nice property that concept subsumption and classification of TBoxes can be computed in polynomial time [3]. Thus, despite of its limited expressiveness, $\mathcal{EL}$ is used in many practical applications – most prominently in the medical ontology SNOMED [15] – and is the basis for the EL profile of the OWL 2.0 standard.

However, some practical applications such as medical or context-aware applications need to represent information that holds only with a certain probability. For instance, context-aware applications may need to represent sensor data in their ontology, which is correct only with a certain probability. This sort of information can be represented by the probabilistic DLs recently introduced in [12], which allows to represent subjective probabilities. These DLs are based on Halpern's probabilistic FOL variant called Type-2 [9] and they allow to assign probabilistic information to concepts (and roles) and not, as in other probabilistic DLs, to concept axioms [11, 10]. In particular, in [12] the DL Prob-$\mathcal{EL}^{01}$ was introduced, which allows to express limited probability values for $\mathcal{EL}$-concepts, and it was shown that instance checking is in PTime.

If in applications different information sources supply varying information on the same topic, the generalization of this information by the lcs gives a description of what the sources agree upon. For both, $\mathcal{EL}$ and Prob-$\mathcal{EL}$, the computation of the lcs is a desirable task. Unfortunately, the lcs w.r.t. general TBoxes does not need to exist in this setting (see [1]), due to cyclic definitions in the TBox.

In this paper we present practical algorithms for computing the lcs up to a certain role-depth for $\mathcal{EL}$ and Prob-$\mathcal{EL}^{01}$. The concept obtained is still a generalization of the input concepts, but not necessarily the least one w.r.t. subsumption. Our computation algorithms are based on the completion algorithms for classification in $\mathcal{EL}$ and Prob-$\mathcal{EL}^{01}$ and thus can be implemented on top of reasoners for these two DLs. Due to space limitations most of the proofs can be found in [14].

2 $\mathcal{EL}$ and Prob-$\mathcal{EL}$

Starting from two disjoint sets N_C and N_R of *concept* and *role names*, respectively, *$\mathcal{EL}$-concept descriptions* are built using the concept *top* ($\top$) and the constructors conjunction ($\sqcap$), and existential restriction ($\exists r.C$). We will often call concept descriptions simply *concepts* for brevity. The semantics of $\mathcal{EL}$ is defined with the help 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 concepts.

A TBox is a set of *concept inclusion axioms* of the form $C \sqsubseteq D$, where C, D are concept descriptions. An interpretation $\mathcal{I}$ *satisfies* the concept inclusion $C \sqsubseteq D$, denoted as $\mathcal{I} \models C \sqsubseteq D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ if it satisfies all axioms in $\mathcal{T}$. A concept C is *subsumed by* a concept D w.r.t. $\mathcal{T}$ (denoted $C \sqsubseteq_{\mathcal{T}} D$) if, for every model $\mathcal{I}$ of $\mathcal{T}$ it holds that $\mathcal{I} \models C \sqsubseteq D$.

We now introduce Prob-$\mathcal{EL}^{01}$, a probabilistic logic that extends $\mathcal{EL}$ with the probabilistic constructors $P_{>0}$ and $P_{=1}$. Intuitively, the concepts $P_{>0}C$ and $P_{=1}C$ express that the probability of C being satisfied is greater than 0, and equal to 1, respectively. This logic was first introduced, along with more expressive probabilistic DLs in [12]. Formally, Prob-$\mathcal{EL}^{01}$ concepts are constructed as

$$C ::= \top \mid A \mid C \sqcap D \mid \exists r.C \mid P_{*}C,$$

where A is a concept name, r is a role name, and $*$ is one of $\{> 0, = 1\}$.

In contrast to previously introduced probabilistic DLs, uncertainty in Prob-$\mathcal{EL}^{01}$ is expressed by assigning probabilities to concepts, instead of axioms. Thus, the semantics of Prob-$\mathcal{EL}^{01}$ generalize the interpretation-based semantics of $\mathcal{EL}$ towards the possible worlds semantic used by Halpern [9]. A *probabilistic interpretation* is of the form

$$\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu),$$

where $\Delta^{\mathcal{I}}$ is the (non-empty) *domain*, W is a set of *worlds*, μ is a discrete probability distribution on W, and for each world $w \in W$, $\mathcal{I}_w$ is a classical $\mathcal{EL}$ interpretation with domain $\Delta^{\mathcal{I}}$. The probability that a given element of the domain $d \in \Delta^{\mathcal{I}}$ belongs to the interpretation of a concept name A is given by

$$p_d^{\mathcal{I}}(A) := \mu(\{w \in W \mid d \in A^{\mathcal{I}_w}\}).$$

The functions $\mathcal{I}_w$ and $p_d^{\mathcal{I}}$ are extended to complex concepts in the usual way for the classical $\mathcal{EL}$ constructors, where the extension to the new constructors P_* is defined as

$$(P_{>0}C)^{\mathcal{I}_w} := \{d \in \Delta^{\mathcal{I}} \mid p_d^{\mathcal{I}}(C) > 0\},$$
$$(P_{=1}C)^{\mathcal{I}_w} := \{d \in \Delta^{\mathcal{I}} \mid p_d^{\mathcal{I}}(C) = 1\}.$$

A probabilistic interpretation $\mathcal{I}$ *satisfies* a concept inclusion $C \sqsubseteq D$, denoted as $\mathcal{I} \models C \sqsubseteq D$ if for every $w \in W$ it holds that $C^{\mathcal{I}_w} \subseteq D^{\mathcal{I}_w}$. It is a *model* of a TBox $\mathcal{T}$ if it satisfies all concept inclusions in $\mathcal{T}$. Let C, D be two Prob-$\mathcal{EL}^{01}$ concepts and $\mathcal{T}$ a TBox. We say that C is *subsumed by* D w.r.t. $\mathcal{T}$ (denoted as $C \sqsubseteq_{\mathcal{T}} D$) if for every model $\mathcal{I}$ of $\mathcal{T}$ it holds that $\mathcal{I} \models C \sqsubseteq D$.

Intuitively, the different worlds express the different possibilities for the domain elements to be interpreted (in the sense of crisp $\mathcal{EL}$ interpretations), and the probability of a concept C being satisfied by a given individual a is given by the probabilities of the different worlds in which a belongs to C.

An interesting property of this logic is that subsumption between concepts can be decided in polynomial time [12]. Moreover, as we will see in the following section, an algorithm for deciding subsumption can be obtained by extending the completion algorithm for (crisp) $\mathcal{EL}$.

3 Completion-based Subsumption Algorithms

We briefly sketch the completion algorithms for deciding subsumption in $\mathcal{EL}$ [3] and in Prob-$\mathcal{EL}^{01}$ [12]. Completion-based methods compute not only subsumption relations for a pair of concept names, but *classify* the whole TBox.

3.1 Completion-based Subsumption Algorithm for $\mathcal{EL}$

Given an $\mathcal{EL}$-TBox $\mathcal{T}$, we use $\mathsf{BC}_{\mathcal{T}}$ to denote the set of *basic concepts for $\mathcal{T}$*, i.e., the smallest set of concept descriptions which contains (1) $\top$ and (2) all concept names used in $\mathcal{T}$. A normal form for $\mathcal{EL}$-TBoxes can be defined as follows.

Definition 1 (Normal Form for $\mathcal{EL}$-TBoxes). *An $\mathcal{EL}$-TBox $\mathcal{T}$ is in* normal form *if all concept inclusions have one of the following forms, where $C_1, C_2, D \in \mathsf{BC}_{\mathcal{T}}$:*

$$C_1 \sqsubseteq D, \quad C_1 \sqcap C_2 \sqsubseteq D, \quad C_1 \sqsubseteq \exists r.C_2 \quad \text{or} \quad \exists r.C_1 \sqsubseteq D.$$

Any $\mathcal{EL}$-TBox $\mathcal{T}$ can be transformed into a normalized TBox $\mathcal{T}'$ by introducing new concept names. $\mathcal{EL}$-TBoxes can be transformed into normal form by applying the normalization rules displayed in Figure 1 exhaustively. These rules replace the GCI on the left-hand side of the rules with the set of GCIs on the right-hand side of the rule.

Let $\mathcal{T}$ be a TBox in normal form to be classified and let $\mathsf{R}_{\mathcal{T}}$ denote the set of all role names appearing in $\mathcal{T}$. The completion algorithm works on two kinds on *completion sets*: $S(C)$ and $S(C, r)$ for each $C \in \mathsf{BC}_{\mathcal{T}}$ and $r \in \mathsf{R}_{\mathcal{T}}$, which

NF1 $C \sqcap \hat{D} \sqsubseteq E \longrightarrow \{ \hat{D} \sqsubseteq A, C \sqcap A \sqsubseteq E \}$

NF2 $\exists r.\hat{C} \sqsubseteq D \longrightarrow \{ \hat{C} \sqsubseteq A, \exists r.A \sqsubseteq D \}$

NF3 $\hat{C} \sqsubseteq \hat{D} \longrightarrow \{ \hat{C} \sqsubseteq A, A \sqsubseteq \hat{D} \}$

NF4 $B \sqsubseteq \exists r.\hat{C} \longrightarrow \{ B \sqsubseteq \exists r.A, A \sqsubseteq \hat{C} \}$

NF5 $B \sqsubseteq C \sqcap D \longrightarrow \{ B \sqsubseteq C, B \sqsubseteq D \}$

where $\hat{C}, \hat{D} \notin \mathsf{BC}_\mathcal{T}$ and A is a new concept name.

Fig. 1. $\mathcal{EL}$ normalization rules

CR1 If $C' \in S(C)$, $C' \sqsubseteq D \in \mathcal{T}$, and $D \notin S(C)$
then $S(C) := S(C) \cup \{D\}$

CR2 If $C_1, C_2 \in S(C)$, $C_1 \sqcap C_2 \sqsubseteq D \in \mathcal{T}$, and $D \notin S(C)$
then $S(C) := S(C) \cup \{D\}$

CR3 If $C' \in S(C)$, $C' \sqsubseteq \exists r.D \in \mathcal{T}$, and $D \notin S(C, r)$
then $S(C, r) := S(C, r) \cup \{D\}$

CR4 If $D \in S(C, r)$, $D' \in S(D)$, $\exists r.D' \sqsubseteq E \in \mathcal{T}$, and $E \notin S(C)$
then $S(C) := S(C) \cup \{E\}$

Fig. 2. $\mathcal{EL}$ completion rules

contain concept names from $\mathsf{BC}_\mathcal{T}$. The intuition is that the completion rules make implicit subsumption relationships explicit in the following sense:

- $D \in S(C)$ implies that $C \sqsubseteq_\mathcal{T} D$,
- $D \in S(C, r)$ implies that $C \sqsubseteq_\mathcal{T} \exists r.D$.

By $\mathsf{S}_\mathcal{T}$ we denote the set containing all completion sets of $\mathcal{T}$. In the algorithm, the completion sets are initialized as follows:

- $S(C) := \{C, \top\}$ for each $C \in \mathsf{BC}_\mathcal{T}$,
- $S(C, r) := \emptyset$ for each $r \in \mathsf{R}_\mathcal{T}$.

The sets $S(C)$ and $S(C, r)$ are extended by applying the completion rules shown in Figure 2 until no more rule applies. After the completion has terminated, the subsumption relation between two basic concepts A and B can be tested by checking whether $B \in S(A)$. Soundness and completeness of the $\mathcal{EL}$-completion algorithm has been shown in [4] as well as that it runs in polynomial time. This algorithm has recently been extended for a probabilistic variant of $\mathcal{EL}$, which we introduce next.

PCR1 If $C' \in S_*(C,v)$, $C' \sqsubseteq D \in \mathcal{T}$, and $D \notin S_*(C,v)$
then $S_*(C,v) := S_*(C,v) \cup \{D\}$

PCR2 If $C_1, C_2 \in S_*(C,v)$, $C_1 \sqcap C_2 \sqsubseteq D \in \mathcal{T}$, and $D \notin S_*(C,v)$
then $S_*(C,v) := S_*(C,v) \cup \{D\}$

PCR3 If $C' \in S_*(C,v)$, $C' \sqsubseteq \exists r.D \in \mathcal{T}$, and $D \notin S_*(C,r,v)$
then $S_*(C,r,v) := S_*(C,r,v) \cup \{D\}$

PCR4 If $D \in S_*(C,r,v)$, $D' \in S_{\gamma(v)}(D,\gamma(v))$, $\exists r.D' \sqsubseteq E \in \mathcal{T}$, and $E \notin S_*(C,v)$
then $S_*(C,v) := S_*(C,v) \cup \{E\}$

PCR5 If $P_{>0}A \in S_*(C,v)$, and $A \notin S_*(C,P_{>0}A)$
then $S_*(C,P_{>0}A) := S_*(C,P_{>0}A) \cup \{A\}$

PCR6 If $P_{=1}A \in S_*(C,v)$, $v \neq 0$, and $A \notin S_*(C,v)$
then $S_*(C,v) := S_*(C,v) \cup \{A\}$

PCR7 If $A \in S_*(C,v)$, $v \neq 0$, $P_{>0}A \in \mathcal{P}_0^{\mathcal{T}}$, and $P_{>0}A \notin S_*(C,v')$
then $S_*(C,v') := S_*(C,v') \cup \{P_{>0}A\}$

PCR8 If $A \in S_*(C,1)$, $P_{=1}A \in \mathcal{P}_1^{\mathcal{T}}$, and $P_{=1}A \notin S_*(C,v)$
then $S_*(C,v) := S_*(C,v) \cup \{P_{=1}A\}$

Fig. 3. Prob-$\mathcal{EL}^{01}$ completion rules

3.2 Completion-based Subsumption Algorithm for Prob-$\mathcal{EL}$

In Prob-$\mathcal{EL}^{01}$, basic concepts also include the probabilistic constructors; that is, the set $\mathsf{BC}_{\mathcal{T}}$ of Prob-$\mathcal{EL}^{01}$ basic concepts for $\mathcal{T}$ is the smallest set that contains (1) $\top$, (2) all concept names used in $\mathcal{T}$, and (3) all concepts of the form P_*A, where A is a concept name in $\mathcal{T}$.

Definition 2 (Normal Form for Prob-$\mathcal{EL}^{01}$-TBoxes). *A Prob-$\mathcal{EL}^{01}$-TBox $\mathcal{T}$ is in* normal form *if all its axioms are of one of the following forms*

$$C \sqsubseteq D, \quad C_1 \sqcap C_2 \sqsubseteq D, \quad C \sqsubseteq \exists r.A, \quad \text{or} \quad \exists r.A \sqsubseteq D,$$

where $C, C_1, C_2, D \in \mathsf{BC}_{\mathcal{T}}$ and A is a new concept name.

The normalization rules in Figure 1 can also be used to transform a Prob-$\mathcal{EL}^{01}$-TBox into this extended notion of normal form. We denote as $\mathcal{P}_0^{\mathcal{T}}$ and $\mathcal{P}_1^{\mathcal{T}}$ the set of all concepts of the form $P_{>0}A$ and $P_{=1}A$, respectively, occurring in a normalized TBox $\mathcal{T}$.

The completion algorithm for Prob-$\mathcal{EL}^{01}$ follows the same idea as the algorithm for $\mathcal{EL}$, but uses several completion sets to deal with the information of what needs to be satisfied in the different worlds of a model. We define the set of worlds $V := \{0, \varepsilon, 1\} \cup \mathcal{P}_0^{\mathcal{T}}$, where the probability distribution μ assigns a probability of 0 to the world 0, and the uniform probability $1/(|V|-1)$ to all other worlds. For each concept name A, role name r and world v, we store the completion sets $S_0(A,v), S_\varepsilon(A,v), S_0(A,r,v)$, and $S_\varepsilon(A,r,v)$. These completion sets are simple generalizations of the completion sets for crisp $\mathcal{EL}$. Intuitively, $D \in S_0(C,v)$ implies $C \sqsubseteq D$ if $v=0$, $C \sqsubseteq P_{=1}D$ if $v=1$, and $C \sqsubseteq P_{>0}D$, otherwise. Likewise, $D \in S_\varepsilon(C,v)$ implies $P_{>0}C \sqsubseteq D$ if $v=0$, $P_{>0}C \sqsubseteq P_{=1}D$ if $v=1$, and $P_{>0}C \sqsubseteq P_{>0}D$, otherwise.

The algorithm initializes the sets as follows for every $A \in \mathsf{BC}_\mathcal{T}$ and $r \in \mathsf{R}_\mathcal{T}$:

- $S_0(A,0) = \{\top, A\}$ and $S_0(A,v) = \{\top\}$ for all $v \in V \setminus \{0\}$,
- $S_\varepsilon(A,\varepsilon) = \{\top, A\}$ and $S_\varepsilon(A,v) = \{\top\}$ for all $v \in V \setminus \{\varepsilon\}$,
- $S_0(A,r,v) = S_\varepsilon(A,r,v) = \emptyset$ for all $v \in V$.

These sets are then extended by exhaustively applying the rules shown in Figure 3, where $* \in \{0,\varepsilon\}$ and $\gamma : V \to \{0,\varepsilon\}$ is defined by $\gamma(0)=0$, and $\gamma(v)=\varepsilon$ for all $v \in V \setminus \{0\}$.

The first four rules are simple adaptations of the completion rules for $\mathcal{EL}$, while the last four rules deal with probabilistic concepts. This algorithm terminates in polynomial time. After termination it holds that, for every pair of concept names A, B, $B \in S_0(A,0)$ if and only if $A \sqsubseteq_\mathcal{T} B$ [12].

4 Computing Least Common Subsumers using Completion

The least common subsumer was first mentioned in [8] and has since been investigated for several DLs. However, most lcs computation algorithms were devised for concept descriptions only or for unfoldable TBoxes (see e.g. [5]) and are not capable of handling general TBoxes. In case of $\mathcal{EL}$ the lcs has been investigated for cyclic TBoxes: the lcs does not need to exist w.r.t. descriptive semantics [2], which is the usual semantics for DLs. One approach to compute the lcs even in the presence of GCIs is to use different semantics for the underlying DL, e.g., greatest fixed-point semantics have been employed in [1, 7]. A different approach was followed in [6, 16], where the lcs was investigated for unfoldable TBoxes written in a “small” DL using concepts from an expressive general background TBox.

All approaches for proving the (non-)existence of the lcs or devising computation algorithms for the lcs are built on a characterization of subsumption for the respective DL and for the underlying TBox formalism. For instance, the lcs algorithm for $\mathcal{EL}$-concept descriptions [5] uses homomorphisms between so-called $\mathcal{EL}$-description trees. The work on the lcs w.r.t. cyclic $\mathcal{EL}$-TBoxes [1, 2] uses (synchronized) simulations between $\mathcal{EL}$-description graphs to characterize subsumption. In this paper we use the completion algorithm from [3] as the underlying characterization of subsumption to obtain a role-depth bounded lcs in $\mathcal{EL}$.

Formally the lcs is defined as follows. Let $\mathcal{T}$ be a TBox and C, D concept descriptions in the DL $\mathcal{L}$, then the $\mathcal{L}$-concept description L is the *least common subsumer (lcs)* of C, D w.r.t. $\mathcal{T}$ (written $lcs_\mathcal{T}(C,D)$) iff

1. $C \sqsubseteq_{\mathcal{T}} L$ and $D \sqsubseteq_{\mathcal{T}} L$, and
2. for all $\mathcal{L}$-concept descriptions E it holds that,
   $C \sqsubseteq_{\mathcal{T}} E$ and $D \sqsubseteq_{\mathcal{T}} E$ implies $L \sqsubseteq_{\mathcal{T}} E$.

Note, that the lcs is defined w.r.t. to a certain DL $\mathcal{L}$. In cases where the lcs is computed for concept descriptions, we can simply use an empty TBox $\mathcal{T}$. Due to the associativity of the lcs operator, the lcs can be defined as a n-ary operation. However, we stick to its binary version for simplicity of the presentation.

4.1 Role-depth bounded lcs in $\mathcal{EL}$

As mentioned, the lcs does not need to exist due to cycles in the TBox. Consider the TBox $\mathcal{T} = \{A \sqsubseteq \exists r.A \sqcap C,\ B \sqsubseteq \exists r.B \sqcap C\}$. The lcs of A and B is then $C \sqcap \exists r.(C \sqcap \exists r.(C \sqcap \exists r.(C \sqcap \cdots$ and cannot be expressed by a finite concept description. To avoid such infinite nestings, we limit the role-depth of the concept description to be computed. Let C, D be $\mathcal{EL}$-concept descriptions, then the *role-depth* of a concept description C (denoted $rd(C)$) is:

- 0 for concept names and $\top$
- $max(rd(C), rd(D))$ for a conjunction $C \sqcap D$, and
- $1 + rd(C)$ for existential restrictions of the form $\exists r.C$.

Now we can define the lcs with limited role-depth for $\mathcal{EL}$.

Definition 3 (Role-depth bounded $\mathcal{EL}$-lcs). *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and C, D $\mathcal{EL}$-concept descriptions and $k \in \mathbb{N}$. Then the $\mathcal{EL}$-concept description L is the* role-depth bounded $\mathcal{EL}$-least common subsumer *of C, D w.r.t. $\mathcal{T}$ and role-depth k (written k-lcs(C, D)) iff*

1. *$rd(L) \leq k$,*
2. *$C \sqsubseteq_{\mathcal{T}} L$ and $D \sqsubseteq_{\mathcal{T}} L$, and*
3. *for all $\mathcal{EL}$-concept descriptions E with $rd(E) \leq k$ it holds that,*
   $C \sqsubseteq_{\mathcal{T}} E$ and $C \sqsubseteq_{\mathcal{T}} E$ implies $L \sqsubseteq_{\mathcal{T}} E$.

4.2 Computing the Role-depth Bounded $\mathcal{EL}$- lcs

The computation algorithm for the role-depth bounded lcs w.r.t. general $\mathcal{EL}$-TBoxes, constructs the concept description from the set of completion sets. More precisely, it combines and intersects the completion sets in the same fashion as in the cross-product computation in the lcs algorithm for $\mathcal{EL}$-concept descriptions from [5].

However, the completion sets may contain concept names that were introduced during normalization. The returned lcs-concept description should only contain concept names that appear in the initial TBox, thus we need to "denormalize" the concept descriptions obtained from the completion sets.

De-normalizing $\mathcal{EL}$-concept Descriptions. The *signature of a concept description* C (denoted $\mathsf{sig}(\mathrm{C})$) is the set of concept names and role names that appear in C. The *signature of a TBox* $\mathcal{T}$ (denoted $\mathsf{sig}(\mathcal{T})$) is the set of concept names and role names that appear in $\mathcal{T}$.

Clearly, the signature of $\mathcal{T}$ may be extended during normalization. To capture the relation between $\mathcal{T}$ and its normalized variant, we introduce the notion of a *conservative extension* as in [13].

Definition 4 ($\mathsf{sig}(\mathcal{T})$-inseparable, conservative extension). *Let $\mathcal{T}_1, \mathcal{T}_2$ be $\mathcal{EL}$-TBoxes.*

- $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *are* $\mathsf{sig}(\mathcal{T}_1)$-inseparable w.r.t. concept inclusion in $\mathcal{EL}$, *iff for all $\mathcal{EL}$-concept descriptions C, D with $\mathsf{sig}(C) \cup \mathsf{sig}(D) \subseteq \mathsf{sig}(\mathcal{T}_1)$, we have $\mathcal{T}_1 \models C \sqsubseteq D$ iff $\mathcal{T}_2 \models C \sqsubseteq D$.*
- $\mathcal{T}_2$ *is a* conservative extension of $\mathcal{T}_1$ w.r.t. concept inclusion in $\mathcal{EL}$, *if*
  - $\mathcal{T}_1 \subseteq \mathcal{T}_2$*, and*
  - $\mathcal{T}_1$ *and* $\mathcal{T}_2$ *are* $\mathsf{sig}(\mathcal{T}_1)$*-inseparable w.r.t. concept inclusion in* $\mathcal{EL}$*.*

However, the extension of the signature by normalization according to the normalization rules from Figure 1 does not affect subsumption tests for $\mathcal{EL}$-concept descriptions formulated w.r.t. the initial signature of $\mathcal{T}$.

Lemma 1. *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and $\mathcal{T}'$ the TBox obtained from $\mathcal{T}$ by applying the $\mathcal{EL}$ normalization rules, C, D be $\mathcal{EL}$-concept descriptions with $\mathsf{sig}(C) \subseteq \mathsf{sig}(\mathcal{T})$ and $\mathsf{sig}(D) \subseteq \mathsf{sig}(\mathcal{T}')$ and D' be the concept description obtained by replacing all names $A \in \mathsf{sig}(\mathcal{T}') \setminus \mathsf{sig}(\mathcal{T})$ from D with $\top$. Then $C \sqsubseteq_{\mathcal{T}'} D$ iff $C' \sqsubseteq_{\mathcal{T}} D'$.*

Proof. Since $\mathcal{T}'$ is a conservative extension of $\mathcal{T}$ w.r.t. concept inclusion in $\mathcal{EL}$, it is implied that $\mathcal{T}$ and $\mathcal{T}'$ are $\mathsf{sig}(\mathcal{T})$*-inseparable w.r.t. concept inclusion in* $\mathcal{EL}$*.* Thus the claim follows directly. □

Lemma 1 guarantees that subsumption relations w.r.t. the normalized TBox $\mathcal{T}'$ between C and D, also hold w.r.t. the original TBox $\mathcal{T}$ for C and D', which is basically obtained from D by removing the names introduced by normalization, i.e., concept names from $\mathsf{sig}(\mathcal{T}') \setminus \mathsf{sig}(\mathcal{T})$.

A Computation Algorithm for *k-lcs*. We assume that the role-depth of each input concept of the lcs has a role-depth less or equal to k. This assumption is motivated by the applications of the lcs on the one hand and on the other by the simplicity of presentation, rather than a technical necessity. The algorithm for computing the role-depth bounded lcs of two $\mathcal{EL}$-concept descriptions is depicted in Algorithm 1. It consists of the procedure k-lcs, which calls the recursive procedure k-lcs-r.

The procedure k-lcs first adds concept definitions for the input concept descriptions to (a copy of) the TBox and transforms this TBox into the normalized TBox $\mathcal{T}'$. Next, it calls the procedure apply-completion-rules, which applies the $\mathcal{EL}$ completion rules exhaustively to the TBox $\mathcal{T}'$, and stores the obtained set of

Algorithm 1 Computation of a role-depth bounded $\mathcal{EL}$-lcs.

Procedure k-lcs $(C, D, \mathcal{T}, k)$
Input: C, D: $\mathcal{EL}$-concept descriptions; $\mathcal{T}$: $\mathcal{EL}$-TBox; k: natural number
Output: $k\text{-}lcs(C, D)$: role-depth bounded $\mathcal{EL}$-lcs of C and D w.r.t $\mathcal{T}$ and k.

1: $\mathcal{T}' :=$ normalize$(\mathcal{T} \cup \{A \equiv C, B \equiv D\})$
2: $\mathsf{S}_{\mathcal{T}'} :=$ apply-completion-rules$(\mathcal{T}')$
3: $L :=$ k-lcs-r $(A, B, \mathsf{S}_{\mathcal{T}'}, k)$
4: **if** $L = A$ **then return** C
5: **else if** $L = B$ **then return** D
6: **else return** remove-normalization-names(L)
7: **end if**

Procedure k-lcs-r (A, B, S, k)
Input: A, B: concept names; S: set of completion sets; k: natural number
Output: $k\text{-}lcs(A, B)$: role-depth bounded $\mathcal{EL}$-lcs of A and B w.r.t $\mathcal{T}$ and k.

1: **if** $B \in S(A)$ **then return** B
2: **else if** $A \in S(B)$ **then return** A
3: **end if**
4: common-names $:= S(A) \cap S(B)$
5: **if** $k = 0$ **then return** $\bigsqcap_{P \in \text{common–names}} P$
6: **else return** $\bigsqcap_{P \in \text{common–names}} P \sqcap \bigsqcap_{r \in \mathsf{R}_{\mathcal{T}}} \Big(\bigsqcap_{(E,F) \in S(A,r) \times S(B,r)} \exists r.\ \text{k-lcs-r}\ (E, F, \mathsf{S}, k-1) \Big)$
7: **end if**

completion sets in S. Then it calls the function k-lcs-r with the concept names A and B for the input concepts, the set of completion sets S, and the role-depth limit k. The result is then de-normalized and returned (lines 4 to 6). More precisely, in case a complex concept description is returned from k-lcs-r, the procedure remove-normalization-names removes concept names that were added during the normalization of the TBox.

The function k-lcs-r gets a pair of concept names, a set of completion sets and a natural number as inputs. First, it tests whether one of the input concepts subsumes the other w.r.t. $\mathcal{T}'$. In that case the name of the subsuming concept is returned. Otherwise the set of concept names that appear in the completion sets of both input concepts is stored in common-names (line 4).[1] In case the role-depth bound is reached ($k = 0$), the conjunction of the elements in common-names is returned. Otherwise, the elements in common-names are conjoined with a conjunction over all roles $r \in \mathsf{R}_{\mathcal{T}}$, where for each r and each element of the cross-product over the r-successors of the current A and B a recursive call to k-lcs-r is made with the role-depth bound reduced by 1 (line 6). This conjunction is then returned to k-lcs.

[1] Note, that the intersection $S(A) \cap S(B)$ is never empty, since both sets contain $\top$.

For $L = \mathsf{k\text{-}lcs}(C, D, \mathcal{T}, k)$ it holds by construction that $rd(L) \leq k$.[2] We now show that the result of the function $\mathsf{k\text{-}lcs}$ is a common subsumer of the input concept descriptions.

Lemma 2. *Let C and D be $\mathcal{EL}$-concept descriptions, $\mathcal{T}$ an $\mathcal{EL}$-TBox, $k \in \mathbb{N}$ and $L = \mathsf{k\text{-}lcs}(C, D, \mathcal{T}, k)$. Then $C \sqsubseteq_{\mathcal{T}} L$ and $D \sqsubseteq_{\mathcal{T}} L$.*

Lemma 1 justifies to replace "normalization names" in the concept description constructed from the normalization sets in the fashion described earlier and still preserve the subsumption relationships. Lemma 2 can be shown by induction on k. For the full proof see [14].

Next, we show that the result of the function $\mathsf{k\text{-}lcs}$ obtained for $\mathcal{EL}$-concept descriptions C and D is the least (w.r.t. subsumption) concept description of role-depth up to k that subsumes the input concepts, see [14].

Lemma 3. *Let C and D be $\mathcal{EL}$-concept descriptions, $\mathcal{T}$ an $\mathcal{EL}$-TBox, $k \in \mathbb{N}$ and $L = \mathsf{k\text{-}lcs}(C, D, \mathcal{T}, k)$ and E an $\mathcal{EL}$-concept description with $rd(E) \leq k$. If $C \sqsubseteq_{\mathcal{T}} E$ and $D \sqsubseteq_{\mathcal{T}} E$, then $L \sqsubseteq_{\mathcal{T}} E$.*

We obtain together with Lemma 2 and Lemma 3 that all conditions of Definition 3 are fulfilled for $\mathsf{k\text{-}lcs}(C, D, \mathcal{T}, k)$.

Theorem 1. *Let C and D be $\mathcal{EL}$-concept descriptions, $\mathcal{T}$ an $\mathcal{EL}$-TBox, $k \in \mathbb{N}$, then $\mathsf{k\text{-}lcs}(C, D, \mathcal{T}, k) \equiv k\text{-}lcs(C, D)$.*

For cases where $\mathsf{k\text{-}lcs}$ returns a concept description with role-depth of less than k we conjecture that it is the exact lcs.

The complexity of the overall method is exponential. However, if a compact representation of the lcs with structure sharing is used, the lcs-concept descriptions can be represented polynomially. In contrast to the lcs algorithm for $\mathcal{EL}$-concept descriptions, the algorithm $\mathsf{k\text{-}lcs}$ does not need to copy concepts[3] that are referenced several times, but proceeds by structure sharing by re-using the completion sets. Thus completion-based algorithm is even advantageous for unfoldable $\mathcal{EL}$-TBoxes such as Snomed.

Moreover, if a *k-lcs* is too general and a bigger role depth of the *k-lcs* is desired, the completion of the TBox does not have to be redone for a second computation. The completion sets can simply be "traversed" further.

4.3 Computing the Role-depth Bounded Prob-$\mathcal{EL}^{01}$-lcs

The computation of the role-depth bounded Prob-$\mathcal{EL}^{01}$-lcs follows the same steps as in Section 4.2. First, it adds concept definitions for the input concepts to the TBox and normalizes it. It then applies the completion rules from Figure 3 exhaustively to produce the set of completion sets S. It then calls a variation of the function $\mathsf{k\text{-}lcs\text{-}r}$ that can deal with probabilistic concepts. The new function

[2] Recall our assumption: the role-depth of each input concept is less or equal to k.
[3] as typically done during unfolding

$\mathsf{k\text{-}lcs\text{-}r}$ is identical to the one presented in Algorithm 1, except that in line 6 it now returns:

$$\begin{aligned}\bigsqcap_{P \in \mathsf{common-names}} P \sqcap \bigsqcap_{r \in \mathsf{R}_{\mathcal{T}}} \Big(& \bigsqcap_{(E,F) \in S_0(A,r,0) \times S_0(B,r,0)} \exists r.\mathsf{k\text{-}lcs\text{-}r}(E, F, \mathsf{S}, k-1) \sqcap \\ & \bigsqcap_{(E,F) \in S_0^{>0}(A,r) \times S_0^{>0}(B,r)} P_{>0}(\exists r.\mathsf{k\text{-}lcs\text{-}r}(E, F, \mathsf{S}, k-1)) \sqcap \\ & \bigsqcap_{(E,F) \in S_0(A,r,1) \times S_0(B,r,1)} P_{=1}(\exists r.\mathsf{k\text{-}lcs\text{-}r}(E, F, \mathsf{S}, k-1))\Big),\end{aligned}$$

where $S_0^{>0}(A, r) := \bigcup_{v \in V \setminus \{0\}} S_0(A, r, v)$. The result is then de-normalized by removing all concept names that were introduced during the normalization phase. The correctness of this procedure can be shown in a similar way as it was done for $\mathcal{EL}$ before.

Theorem 2. *Let C and D be Prob-$\mathcal{EL}^{01}$-concept descriptions, $\mathcal{T}$ a Prob-$\mathcal{EL}^{01}$-TBox, and $k \in \mathbb{N}$; then* $\mathsf{k\text{-}lcs}(C, D, \mathcal{T}, k) \equiv$ *k-lcs*(C, D).

Again, the proof is given in [14].

5 Conclusions

In this paper we have presented a practical method for computing the role-depth bounded lcs of $\mathcal{EL}$-concepts w.r.t. a general TBox. Our approach is based on the completion sets that are computed during classification of a TBox. Thus, any of the available implementation of the $\mathcal{EL}$ completion algorithm can be easily extended to an implementation of the lcs computation algorithm. We also showed that the same idea can be adapted for the computation of the lcs in the probabilistic DL Prob-$\mathcal{EL}^{01}$.

As future work, we want to investigate the computation of the most specific Prob-$\mathcal{EL}^{01}$ concept (msc) that describes a given individual in an ABox. We also plan to investigate the bottom-up constructions (i. e. lcs and msc computations) in more expressive probabilistic DLs. One possible extension is by studying Prob-$\mathcal{ALE}$. A second approach is to weaken the restrictions imposed in Prob-$\mathcal{EL}^{01}$, allowing for different probabilistic constructors.

References

1. F. Baader. Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI-03)*, 2003.
2. F. Baader. Terminological cycles in a description logic with existential restrictions. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI-03)*, 2003.
3. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05)*, 2005.

4. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. LTCS-Report LTCS-05-01, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2005. See http://lat.inf.tu-dresden.de/research/reports.html.
5. F. Baader, R. Küsters, and R. Molitor. Computing least common subsumer in description logics with existential restrictions. In *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99)*, 1999.
6. F. Baader, B. Sertkaya, and A.-Y. Turhan. Computing the least common subsumer w.r.t. a background terminology. *Journal of Applied Logics*, 2007.
7. S. Brandt. *Standard and Non-standard Reasoning in Description Logics.* PhD thesis, Institute for Theoretical Computer Science, TU Dresden, 2006.
8. W. W. Cohen, A. Borgida, and H. Hirsh. Computing least common subsumers in description logics. In *Proc. of the 10th Nat. Conf. on Artificial Intelligence (AAAI-92)*, 1992.
9. J. Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46:311–350, 1990.
10. P. Klinov, B. Parsia, and U. Sattler. On correspondences between probabilistic first-order and description logics. In *Proc. of the 2008 Description Logic Workshop (DL 2009)*, volume 477 of *CEUR*, 2009.
11. T. Lukasiewicz. Expressive probabilistic description logics. *Artificial Intelligence*, 172(6-7):852–883, 2008.
12. C. Lutz and L. Schröder. Probabilistic description logics for subjective probabilities. In *Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-10)*, 2010. To appear.
13. C. Lutz and F. Wolter. Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. *Journal of Symbolic Computation*, 45(2):194 – 228, 2010.
14. R. Peñaloza and A.-Y. Turhan. Completion-based computation of least common subsumers with limited role-depth for $\mathcal{EL}$ and prob-$\mathcal{EL}^{01}$. LTCS-Report LTCS-10-02, Chair f. Automata Theory, Inst. for Theoretical Computer Science, TU Dresden, Germany, 2010. See http://lat.inf.tu-dresden.de/research/reports.html.
15. K. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with snomed-rt. *Journal of the American Medical Informatics Assoc.*, 2000. Fall Symposium Special Issue.
16. A.-Y. Turhan. *On the Computation of Common Subsumers in Description Logics.* PhD thesis, TU Dresden, Institute for Theoretical Computer Science, 2007.