**LTCS–Report**

# Completion-based computation of least common subsumers with limited role-depth for $\mathcal{EL}$ and Prob-$\mathcal{EL}^{01}$

Rafael Peñaloza and Anni-Yasmin Turhan
Theoretical Computer Science, TU Dresden, Germany
*last name*@tcs.inf.tu-dresden.de

LTCS-Report 10-02

# Completion-based computation of least common subsumers with limited role-depth for $\mathcal{EL}$ and prob $\mathcal{EL}$

**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 report 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 Motivation

The *least common subsumer* (lcs) inference yields a concept description, that generalizes a collection of concept descriptions by extracting their commonalities. This inference was first introduced in [11]. The lcs is used in many applications, most prominently, it is used in the bottom-up construction of knowledge bases [6], where a new concept is to be introduced that has a collection of selected individuals as instances. This new concept is then a candidate for a new concept definition in the TBox. This can be achieved by first generalizing each selected individual into a concept description (by computing its *most specific concept* (msc)) and in a second step apply the lcs to these concept descriptions. The concept description returned by the lcs can then be (edited and) added to the TBox. Further applications of the lcs include similarity-based Information Retrieval [18, 1] or learning from examples [13, 12].

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 [4]. Thus, despite of its limited expressiveness, $\mathcal{EL}$ is used in many practical applications – most prominently in the medical ontology Snomed [19] – and is the basis for EL profile of the OWL 2.0 standard[1].

[1]http://www.w3.org/TR/owl2-profiles/

However, some practical applications such as medical or context-aware applications have 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. Whereas medical data may based on statistical analysis. This sort of information can be represented by the probabilistic DLs recently introduced [16]. Here probabilistic information is assigned to concepts (and roles) directly and not, as in other probabilistic DLs, to concept axioms [15, 14]. In particular in [16] the DL Prob-$\mathcal{EL}$ and its sublanguage Prob-$\mathcal{EL}^{01}$ was introduced. Latter allows to express limited probability values for $\mathcal{EL}$-concepts, and it was shown in [16] that instance checking can be done in polynomial time.

In applications where different information sources supply varying information for the same entity, the generalization of this information, gives evidence for 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 or even cyclic $\mathcal{EL}$-TBoxes does not need to exist (see [2]), due to cyclic definitions in the TBox.

In this report we present practical algorithms for computing the lcs up to a certain role-depth for general $\mathcal{EL}$- and Prob-$\mathcal{EL}^{01}$-TBoxes. The concept description obtained by the algorithm is still a generalization of the input concepts, but not necessarily the least one w.r.t. subsumption. We argue that this "good common subsumer" still is useful in practice. To the best of our knowledge this report gives the first computation algorithm for the lcs in a DL that captures (a limited form) probabilistic information. In [17] a probabilistic lcs for the DL $\mathcal{ALN}$ was investigated, with the idea to relax the notion of the lcs in order to avoid overfitting. This relaxed notion is helpful in cases where the concept obtained from the lcs is used as a search pattern.

Our algorithms to compute the lcs up to a certain role-depth for general $\mathcal{EL}$- and Prob-$\mathcal{EL}^{01}$-TBoxes are based upon the completion algorithms for computing classification in $\mathcal{EL}$ and Prob-$\mathcal{EL}^{01}$ and thus can be easily implemented on top of completion-based reasoners for these two logics.

# 2 Description Logics

In Description logics (DLs), *concept descriptions* are inductively defined with the help of a set of *concept constructors*, starting with a set $N_C$ of *concept names*, a set $N_R$ of *role names*. From elements of these sets complex concept descriptions can be obtained by concept constructors. In this report we are interested to reasoning with concept descriptions.

## 2.1 The DL $\mathcal{EL}$

The DL $\mathcal{EL}$ allows for two concept constructors: conjunction and existential restrictions. The DL $\mathcal{EL}$ also contains the *top-concept*, denoted $\top$.

**Definition 1 (Syntax of $\mathcal{EL}$-concept descriptions)** *Let $A$ denote a concept name, $r$ denotes a role and $C_1, C_2$ denote arbitrary $\mathcal{EL}$-concepts. An $\mathcal{EL}$-*concept description *$C$ can be obtained by the following rule:*

$$C \longrightarrow \top \mid A \mid C_1 \sqcap C_2 \mid \exists r.C_1$$

.

The semantics of a concept description is defined in terms of an *interpretation* $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$. The domain $\Delta$ of $\mathcal{I}$ is a non-empty set of individuals and the interpretation function $\cdot^{\mathcal{I}}$ maps each concept name $A \in N_C$ to a set $P^{\mathcal{I}} \subseteq \Delta$ and each role name $r \in N_R$ to a binary relation $r^{\mathcal{I}} \subseteq \Delta \times \Delta$. Arbitrary $\mathcal{EL}$-concept descriptions are interpreted as follows:

**Definition 2 (Semantics of $\mathcal{EL}$-concept descriptions)** *The top-concept is interpreted as the domain ($\top^{\mathcal{I}} = \Delta$). The extension of $\cdot^{\mathcal{I}}$ to arbitrary $\mathcal{EL}$-concept descriptions is inductively defined, as follows:*

- $(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$, *and*
- $(\exists r.C)^{\mathcal{I}} = \{x \in \Delta \mid \exists y : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$ *for $r \in N_R$.*

Concept descriptions can be assigned a name or sub-concept super-concept relationships between arbitrary concept descriptions established in the TBox.

**Definition 3 (GCI, TBox)** *Let $C_1$, $C_2$, $D_1$ and $D_2$ be concept descriptions, then*

$$C_1 \sqsubseteq C_2$$

*is a* general concept inclusion axiom (GCI)*. The semantics of GCIs is given by the interpretation function. A GCI $C_1 \sqsubseteq C_2$ is satisfied for a TBox $\mathcal{T}$, iff $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$.*

*A TBox $\mathcal{T}$ is finite set of contains GCIs. An interpretation is a* model *of a TBox, if for all $C_1 \sqsubseteq C_2 \in \mathcal{T}$ and $D_1 \equiv D_2 \in \mathcal{T}$, it holds that $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$ and $D_1^{\mathcal{I}} = D_2^{\mathcal{I}}$.*

*If the concept axioms in the TBox contain only $\mathcal{EL}$-concept descriptions, we call it an $\mathcal{EL}$-*TBox.

It is easy to see that *concept equivalence* between two concept descriptions (written $C_1 \equiv C_2$) can be stated by two GCIs: $C_1 \equiv C_2$ and $C_1 \sqsubseteq C_1$. A special form of GCI are *primitive concept definitions.* Primitive concept definitions allow only for concept names on the left-hand side of the concept axiom.

Based on the semantics of concept descriptions and TBoxes a number of inference problems for DLs have been defined. The most relevant ones can be described as follows:

- *Concept satisfiability.* A concept $C$ is *satisfiable w.r.t. a TBox $\mathcal{T}$* if there exists a model $\mathcal{I}$ of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$.
- *Concept subsumption.* A concept $C$ *subsumes* a concept $D$ w.r.t. a TBox $\mathcal{T}$ (written $C \sqsubseteq_{\mathcal{T}} D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in every model $\mathcal{I}$ of $\mathcal{T}$.

Since the DL $\mathcal{EL}$ cannot express contradictions, every $\mathcal{EL}$-concept is trivially satisfiable. In the remainder of this report, we will thus concentrate on subsumption as the basic reasoning task.

## 2.2 Prob-$\mathcal{EL}$

We first introduce Prob-$\mathcal{EL}$, a probabilistic variant of $\mathcal{EL}$ that allows reasoning with probabilistic concepts. We later present the variant Prob-$\mathcal{EL}^{01}$ in which probabilistic expressions can only express that a concept is necessarily true, i. e. occurs with probability equal to 1, or that it is not impossible, i. e. occurs with probability greater than 0. These probabilistic logics were first introduced in [16].

The probabilistic DL Prob-$\mathcal{EL}$ extends $\mathcal{EL}$ with the constructor $P_*$. That is, Prob-$\mathcal{EL}$ concepts are constructed as

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

where $A$ is a concept name, $r$ a role name and $n$ a rational number from the interval $[0, 1]$. Intuitively, $P_* C$ expresses that the concept $C$ is true with probability at least $n$.

The semantics of Prob-$\mathcal{EL}$ generalize the interpretation-based semantics of $\mathcal{EL}$. 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*, $\mu$ 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 concept name $A$ is given by

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

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

$$(P_*C)^{\mathcal{I}_w} := \{d \in \Delta^{\mathcal{I}} \mid p_d^{\mathcal{I}}(C) \geq n\}.$$

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 $\mathcal{T}$.

The variant probabilistic DL Prob-$\mathcal{EL}^{01}$ extends $\mathcal{EL}$ with the probability constructors $P_{>0}$ and $P_{=1}$, with the straightforward semantics. Intuitively, this logic allows us to express the possibility or necessity of a concept, but does not allow a fine-grained description of the probability with which the concept is satisfied. This simple probabilistic variant retains the good complexity properties held by $\mathcal{EL}$. Moreover, as it will be shown in the following section, reasonig in Prob-$\mathcal{EL}^{01}$ can be performed through a variant of the completion algorithm for $\mathcal{EL}$.

## 3 Completion-based subsumption algorithms

In this section we recapitulate the completion algorithms for testing subsumption in $\mathcal{EL}$ from [8, 4] and for testing subsumption in prob-$\mathcal{EL}$ from [16].[2] Completion-based methods compute the subsumption relations between *all* concept names from the TBox simultaneously, i.e., they *classify* the Tbox.

### 3.1 Completion-based subsumption algorithm for $\mathcal{EL}$

Given a TBox $\mathcal{T}$, we use $\mathsf{BC}_{\mathcal{T}}$ to denote the set of *basic concept descriptions for* $\mathcal{T}$ , i.e., the smallest set of concept descriptions which contains

- the top concept $\top$;
- all concept names used in $\mathcal{T}$;

Based on this, a normal form for TBoxes can be defined as follows.

**Definition 4 (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 $C_1, C_2 \in \mathsf{BC}_{\mathcal{T}}$*

[2]Actually, we prune the completion algorithm given for instance checking in prob-$\mathcal{EL}$ down to a method for testing subsumption in this DL.

$$
\begin{array}{lrcl}
\textsf{NF1} & C \sqcap \hat{D} \sqsubseteq E & \longrightarrow & \{\, \hat{D} \sqsubseteq A, C \sqcap A \sqsubseteq E \,\} \\
\textsf{NF2} & \exists r.\hat{C} \sqsubseteq D & \longrightarrow & \{\, \hat{C} \sqsubseteq A, \exists r.A \sqsubseteq D \,\} \\
\textsf{NF3} & \hat{C} \sqsubseteq \hat{D} & \longrightarrow & \{\, \hat{C} \sqsubseteq A, A \sqsubseteq \hat{D} \,\} \\
\textsf{NF4} & B \sqsubseteq \exists r.\hat{C} & \longrightarrow & \{\, B \sqsubseteq \exists r.A, A \sqsubseteq \hat{C} \,\} \\
\textsf{NF5} & B \sqsubseteq C \sqcap D & \longrightarrow & \{\, B \sqsubseteq C, B \sqsubseteq D \,\}
\end{array}
$$

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

Figure 1: $\mathcal{EL}$-normalization rules.

*and* $D \in \mathsf{BC}_{\mathcal{T}} \cup \{\bot\}$*:*

$$
\begin{aligned}
C_1 &\sqsubseteq D, \\
C_1 \sqcap C_2 &\sqsubseteq D, \\
C_1 &\sqsubseteq \exists r.C_2 \text{ or} \\
\exists r.C_1 &\sqsubseteq D.
\end{aligned}
$$

Any TBox $\mathcal{T}$ can be turned into a normalized TBox $\mathcal{T}'$ by introducing new concept and role 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.

The *signature of a concept description* $C$ (denoted $\mathsf{sig}(C)$) 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. However, this extension does not affect subsumption tests for $\mathcal{EL}$-concept descriptions formulated w.r.t. the signature of $\mathcal{T}$. To capture the relation between $\mathcal{T}$ and its normalized variant, we introduce the notion of a *conservative extension*.

**Definition 5 ($\mathsf{sig}(\mathcal{T})$-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*

**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 $S(C) := S(C) \cup \{E\}$

Figure 2: $\mathcal{EL}$ completion Rules

– *$\mathcal{T}_1$ and $\mathcal{T}_2$ are* $\mathsf{sig}(\mathcal{T}_1)$*-inseparable w.r.t. concept inclusion in $\mathcal{EL}$.*

The normalized TBox $\mathcal{T}$' is a *conservative extension* of $\mathcal{T}$ as already stated in [5], i.e., every model of $\mathcal{T}'$ is also a model of $\mathcal{T}$, and every model of $\mathcal{T}$ can be extended to a model of $\mathcal{T}'$ by appropriately adding the interpretations of the additional concept names introduced by normalization.

Let $\mathcal{T}$ be the TBox 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 $S(C, r)$, which 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 of 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 $r \in \mathsf{R}_\mathcal{T}$.

Then the sets $S(C)$ and $S(C, r)$ are extended by applying the completion rules shown in Table 2 until no more rule applies. After the completion has terminated, the subsumption relation between two named concepts $A$ and $B$ from $\mathcal{T}$ can be tested by checking whether $B \in S(A)$. Soundness and completeness of the $\mathcal{EL}$-completion algorithm has been shown in [9] as well as that it runs in polynomial time. This algorithm has recently been extended in [16] to a probabilistic variant of $\mathcal{EL}$, both of which we introduce in the next section.

## 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 6 (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*

$$C \sqsubseteq D, \quad C_1 \sqcap C_2 \sqsubseteq D, \quad C \sqsubseteq \exists r.A, \quad \exists r.A \sqsubseteq D,$$

*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. In the following, 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, occuring 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 $\mu$ 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 $S_\varepsilon(A, r, v)$.

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

- $S_0(A, 0) = \{\top, A\}$ 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 $A \sqsubseteq_\mathcal{T} B$ [16].

**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 $S_*(C, v) := S_*(C, v) \cup \{P_{=1}A\}$

Figure 3: Prob-$\mathcal{EL}^{01}$ completion rules

## 4 Computing Least Common Subsumers using Completion

The least common subsumer was first mentioned in [11] 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. [6]) and are not capable for handling general TBoxes. In case of $\mathcal{EL}$ the lcs has been investigated for more expressive TBoxes. It turned out in [3] that the lcs w.r.t. cyclic TBoxes the lcs does not need to exist w.r.t. descriptive semantics – 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. This approach was pursued in [2, 10], where greatest fixed-point semantics have been employed. A different approach was followed in [7], where the lcs was investigated for unfoldable TBoxes written in a "small" DL using concept names defined in a more expressive and 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 (and unfoldable TBoxes) [6] uses homomorphisms between so-called $\mathcal{EL}$-description trees, which are basically syntax trees of $\mathcal{EL}$-concept descriptions. The work on the lcs w.r.t. cyclic $\mathcal{EL}$-TBoxes [2, 3] uses (synchronized) simulations between $\mathcal{EL}$-description graphs to characterize subsumption. In this paper we use the completion algorithm from [4] as the underlying characterization of subsumption to obtain a role-depth bounded lcs in $\mathcal{EL}$.

Formally the least common subsumer inference is defined as follows.

**Definition 7 (Least common subsumer)** *Let $\mathcal{T}$ be a TBox and $C, D$ concept descriptions in the DL $\mathcal{L}$, then $L$ 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 $C \sqsubseteq_{\mathcal{T}} E$ implies $L \sqsubseteq_{\mathcal{T}} E$.*

Note, that the lcs is defined w.r.t. to a certain $\mathcal{L}$. In cases the lcs is computed for concept descriptions alone, we can using an empty TBox.

Due to the associativity of the lcs operator, the lcs can be defined as a $n$-ary operation. 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 exists 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 $\exists r.(C \sqcap \exists r.(C \sqcap \exists r.(C \sqcap \cdots$ 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. Intuitively, the role-depth is the maximal number of nested quantifiers of a concept description.

**Definition 8 (role-depth)** *Let $C$, $D$ be $\mathcal{EL}$-concept descriptions. 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 9 (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 $k\text{-}lcs_{\mathcal{T}}(C, D)$) 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* $L \sqsubseteq_{\mathcal{T}} E$.

Please note that in case the exact lcs has a role-depth less than $k$ the role-depth bounded lcs is the exact lcs.

### 4.1.1 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 [6].

The idea is that the normalization of the TBox introduces a new named concepts for each top-level conjunct and for each qualification of a top-level existential restriction that appears in the TBox. We call these new names *normalization names* in the following. During completion the normalization names and the concept names from the initial TBox are then collected in the completion sets of those concepts they subsume. Thus each normalization name in a completion set of, say $C$, provides a "pointer" to a complex concept description that is a subsumer of $C$. We compute the lcs by introducing new names for the input concepts of the lcs, do the normalization and completion and then collect the lcs concept from the completion sets of the input concepts. However, the returned lcs-concept description should only contain concept names that appear in the initial TBox, thus we need to "de-normalize" the concept descriptions obtained from the completion sets.

### 4.1.2 De-normalizing $\mathcal{EL}$-concept descriptions

The *signature of a concept description* $C$ (denoted $\mathsf{sig}(C)$) 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 $\mathcal{T}$. However, the extension of the signature does not affect subsumption tests for $\mathcal{EL}$-concept descriptions formulated w.r.t. the signature of $\mathcal{T}$.

**Lemma 10** *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 $\mathsf{sig}(C) \cup \mathsf{sig}(D) \subseteq \mathsf{sig}(\mathcal{T}')$ and $C'$ ($D'$) be the concept description obtained by removing all names $A \in \mathsf{sig}(T') \setminus \mathsf{sig}(\mathcal{T})$ from $C$ ($D$). 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 10 guarantees that subsumption relations w.r.t. the normalized TBox $\mathcal{T}'$ between arbitrary $\mathcal{EL}$-concept descriptions $C$ and $D$, that contain only names from $\mathsf{sig}(\mathcal{T}')$, also hold w.r.t. the original TBox $\mathcal{T}$ for $C$ and $D$ with the names from $\mathsf{sig}(T') \setminus \mathsf{sig}(\mathcal{T})$ removed.

### 4.1.3 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 and due to the simplicity of presentation 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.

**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: $k$-$lcs(C, D)$: 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 $(A, B, \mathsf{S}, k)$
Input: $A, B$: concept names; $\mathsf{S}$: set of completion sets; $k$: natural number
Output: $k$-$lcs(A, B)$: 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 $\bigsqcap_{P \in \mathsf{common-names}} P$
6: else return $\bigsqcap_{P \in \mathsf{common-names}} P \sqcap$
        $\bigsqcap_{r \in \mathsf{R}_{\mathcal{T}}} \Big( \bigsqcap_{(E,F) \in S(A,r) \times S(B,r)} \exists r.$ k-lcs-r $(E, F, \mathsf{S}, k-1)\Big)$
7: end if
```

The procedure k-lcs first adds concept definitions for the input concept descriptions to (a copy of) the TBox and transforms this TBox into a TBox in normalization form $\mathcal{T}$'. 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 completion sets in in $\mathsf{S}$. Then it calls the function k-lcs-r with the concept names $A$ and $B$ for the input concepts, the set of completions sets $\mathsf{S}$, and the role-depth $k$. 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 intersected (line 4) and stored in common-names.[3] In

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

case $k = 0$, i.e. the role-depth bound is reached, 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.

For $L = \mathsf{k\text{-}lcs}(C, D, \mathcal{T}, k)$ it holds by construction that $rd(L) \leq k$.[4] We now show that the result of the function k-lcs is a common subsumer of the input concept descriptions.

**Lemma 11** *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 $D \sqsubseteq_\mathcal{T} L$.*

**Proof.** We show the claim by induction on $k$. By $\mathcal{T}'$ we denote the TBox obtained from $\mathcal{T}$ by applying the normalization rules.

*Case: $k = 0$.*
Thus $rd(L) = 0$ and $L = A_1 \sqcap A_2 \ldots \sqcap A_m$, with $A_i \in \mathsf{BC}_\mathcal{T}$. Now, either

- one of the input concept descriptions $C, D$ subsumes the other one and thus appears in the subsumer set of the (name of) the other concept. Then its name is returned from k-lcs-r (lines 1 and 2) and then the initial concept description is returned by k-lcs (lines 4 and 5) and thus the claim holds.

- or, $L$ is obtained from: $L =$remove-normalization-names$(S(A) \cap S(B))$ and thus $\sqcap_{C' \in S(A)} C' \sqsubseteq_{\mathcal{T}'} L$. Since $C \equiv_{\mathcal{T}'} A$, it holds that $C \sqsubseteq_{\mathcal{T}'} \sqcap_{C' \in S(A)} C'$. Thus $C \sqsubseteq_{\mathcal{T}'} \sqcap_{C' \in S(A)} C' \sqsubseteq_{\mathcal{T}'} L$ and by application of Lemma 10 we obtain $C \sqsubseteq_\mathcal{T} L$. Argument for $D \sqsubseteq_\mathcal{T} L$ is analogous.

*Case: $k = n$, with $n \geq 0$.*
Assume that the result holds for some $k \geq 0$, and we show it for $k+1$. Since $k \geq 0$, $L =$ remove-normalization-names$(\bigsqcap_{A_i \in S(A) \cap S(B)} A_i \sqcap \bigsqcap_{r \in \mathsf{R}_\mathcal{T}} \big(\bigsqcap_{(E,F) \ \in \ S(A,r) \times S(B,r)} \exists r.$ k-lcs-r $(E, F, \mathsf{S}, k - 1))$. It follows for the conjunction of concept names in $L$ $\bigsqcap_{A_i \in S(A) \cap S(B)} A_i$ that $F \sqsubseteq_\mathcal{T}$remove-normalization-names$(S(A) \cap S(B))$ holds for all $F \in \{C, D\}$ by the same argument as in the base case. It remains to show that the conjunction of existential restrictions is a subsumer of $C$ and $D$. Due to the assumption that result holds for $k-1$, we know that for all $r \in \mathsf{R}_\mathcal{T}$ and for all $E \in S(A, r)$ it holds that $A \sqsubseteq_{\mathcal{T}'} \exists r.E$, and hence $A \sqsubseteq_{\mathcal{T}'} \bigsqcap_{(E,F) \in S(A,r) \times S(B,r)} \exists r.$k-lcs$(E, F, \mathcal{T}', k)$. The argument for $D \sqsubseteq_\mathcal{T} L$ is analogous. ❏

[4]Recall our assumption that the role-depth of each input concept is less or equal to $k$.

Next, we show that the result of the function k-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.

**Lemma 12** *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 $L \sqsubseteq_{\mathcal{T}} E$.*

**Proof.** We prove the claim by induction on $rd(E)$.

*Case: $rd(E) = 0$.*
The concept $E$ is the concept name $A$ or $B$ or a conjunction of concept names. In the first case we obtain the result $L = C$ from k-lcs and thus $E \equiv_{\mathcal{T}} A \equiv_{\mathcal{T}} C \equiv_{\mathcal{T}} L$. In the second case we obtain $L = D$ from k-lcs-r and thus $E \equiv_{\mathcal{T}} B \equiv_{\mathcal{T}} D \equiv_{\mathcal{T}} L$. In the third case $E$ is a conjunction of concept names: $E = A_1 \sqcap \ldots \sqcap A_m$ for $A_i \in N_C$. Since $C \sqsubseteq_{\mathcal{T}} E$ and $D \sqsubseteq_{\mathcal{T}} E$ holds, we have $C \sqsubseteq_{\mathcal{T}} A_i$ and $D \sqsubseteq_{\mathcal{T}} A_i$ for all $A_i$ with $1 \leq i \leq m$. From this follows $A_i \in S(A) \cap S(B)$ for all $i$, which implies that $L \sqsubseteq_{\mathcal{T}} \bigsqcap_{C' \in S(A) \cap S(B)} C' \sqsubseteq_{\mathcal{T}} E$.

*Case: $rd(E) = n$, with $n \geq 0$.*
$E$ can contain two kinds of conjuncts: (possibly zero) concept names and at least one existential restriction. The concept names in $E$ must appear in $L$ as well by an argument analogous to the base case.

Let $\exists r.E'$ be a (top-level) conjunct of $E$. Since $C \sqsubseteq_{\mathcal{T}'} E$ and $D \sqsubseteq_{\mathcal{T}'} E$, there must exist $A' \in S(A, r)$ and $B' \in S(B, r)$ such that $A' \sqsubseteq_{\mathcal{T}} E'$ and $B' \sqsubseteq_{\mathcal{T}} E'$. By induction hypothesis, we then have that $\exists r.\mathsf{k\text{-}lcs}(A', B', \mathcal{T}, k-1) \sqsubseteq_{\mathcal{T}} E'$. By $L \sqsubseteq_{\mathcal{T}} \exists r.\mathsf{k\text{-}lcs}(A', B', \mathcal{T}, k-1)$, it is implied that $L \sqsubseteq_{\mathcal{T}} E$. ❏

We obtain together with Lemma 11 and Lemma 12 that all conditions of Definition 9 are fulfilled for $\mathsf{k\text{-}lcs}(C, D, \mathcal{T}, k)$.

**Theorem 13** *Let $C$ and $D$ be $\mathcal{EL}$-concept descriptions, $\mathcal{T}$ an $\mathcal{EL}$-TBox, $k \in \mathbb{N}$, then $\mathsf{k\text{-}lcs}(C, D, \mathcal{T}, k) \equiv_{\mathcal{T}} k\text{-}lcs_{\mathcal{T}}(C, D)$.*

Thus, if the lcs exists and has a role-depth of less than $k$, the algorithm k-lcs yields the exact lcs.

The complexity of the overall method is polynomial for the binary lcs and exponential for the $n$-ary lcs due to the cross-product computation–just as in the case for the lcs for $\mathcal{EL}$-concept descriptions, see [6]. However, in contrast to the lcs algorithm for $\mathcal{EL}$-concept descriptions, the algorithm k-lcs does not need to perform unfolding, i.e. copying of concepts, but proceeds by structure sharing if a concept from the TBox needs to be examined. Thus it is even advantageous for unfoldable $\mathcal{EL}$-TBoxes such as Snomed [19].

## 4.2 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 4.1.1. First, it adds concept definitions for the input concepts to the TBox and normalizes it. It then applies the completion rules exhaustively to produce the set of completion sets $\mathsf{S}$. It then calls a variation of the function $\mathsf{k\text{-}lcs\text{-}r}$ that can deal with probabilistic concepts. The new function $\mathsf{k\text{-}lcs\text{-}r}$ 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 $\mathsf{common\text{-}names} := S_0(A,0)\cap S_0(B,0)$ and $S_0^{>0}(A,r) := \bigcup_{v\in V\setminus\{0\}} S_0(A,r,v)$.

This new computation generalizes the idea for obtaining the lcs for $\mathcal{EL}$, taking also into consideration the probabilistic concepts. Basically, if there is a concept name $E \in S_0(A,r,v)$ for some world $v \in V \setminus \{0\}$, then it holds that $A \sqsubseteq P_{>0}(\exists r.E)$. Thus, the last two conjunctions express the existential restrictions that subsume both concepts $A$ and $B$ with probability greater than 0 and 1, respectively.

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.

**Lemma 14** *Let $C$ and $D$ be Prob-$\mathcal{EL}^{01}$-concept descriptions, $\mathcal{T}$ a Prob-$\mathcal{EL}^{01}$-TBox, $k \in \mathbb{N}$ and $L = \mathsf{k\text{-}lcs}(C,D,\mathcal{T},k)$. Then $C \sqsubseteq_\mathcal{T} L$ and $D \sqsubseteq_\mathcal{T} L$.*

**Proof.** We show the result by induction on $k$. First, if $k = 0$, then $rd(L) = 0$. If one of the concepts $C, D$ subsumes the other, then the algorithm returns it. Otherwise, we have that $L = \bigsqcap_{E\in S_0(A,0)\cap S_0(B,0)} E$ and thus, since $C \equiv_\mathcal{T} A$, $C \sqsubseteq_\mathcal{T} \bigsqcap_{E\in S_0(A,0)} E \sqsubseteq_\mathcal{T} L$. An analogous argument shows that $D \sqsubseteq_\mathcal{T} L$.

Assume now that the result holds for some $k \geq 0$, and we show it for $k+1$. First, using the same argument for the base case, it is easy to see that $C \sqsubseteq_\mathcal{T} \bigsqcap_{E\in S_0(A,0)\cap S_0(B,0)} E$. Let now $r \in \mathsf{R}_\mathcal{T}$. Then, for all $E \in S_0(A,r,0)$ it holds that $A \sqsubseteq_\mathcal{T} \exists r.E$, and hence $A \sqsubseteq_\mathcal{T} \bigsqcap_{(E,F)\in S_0(A,r,0)\times S_0(B,r,0)} \exists r.\mathsf{k\text{-}lcs}(E,F,\mathcal{T},k)$. Additionally, for all $E \in S_0(A,r,v)$ with $v \in V \setminus \{0\}$ it holds that $A \sqsubseteq_\mathcal{T} P_{>0}(\exists r.E)$, and for all $E \in S_0(A,r,1)$ it holds $A \sqsubseteq_\mathcal{T} P_{=1}(\exists r.E)$. Hence,

$$A \sqsubseteq_\mathcal{T} \bigsqcap_{(E,F)\in S_0(A,r,0)\times S_0(B,r,0)} P_{>0}(\exists r.\mathsf{k\text{-}lcs}(E,F,\mathcal{T},k))$$

and

$$A \sqsubseteq_{\mathcal{T}} \bigsqcap_{(E,F)\in S_0(A,r,0)\times S_0(B,r,0)} P_{=1}(\exists r.\mathsf{k\text{-}lcs}(E,F,\mathcal{T},k)).$$

And thus, $C \sqsubseteq_{\mathcal{T}} L$. An analogous argument yields the result for $D \sqsubseteq_{\mathcal{T}} L$. ❑

**Lemma 15** *Let $C$ and $D$ be Prob-$\mathcal{EL}^{01}$-concept descriptions, $\mathcal{T}$ an Prob-$\mathcal{EL}^{01}$-TBox, $k \in \mathbb{N}$ and $L = \mathsf{k\text{-}lcs}(C,D,\mathcal{T},k)$ and $E$ an Prob-$\mathcal{EL}^{01}$-concept description with $rd(E) \leq k$. If $C \sqsubseteq_{\mathcal{T}} E$ and $D \sqsubseteq_{\mathcal{T}} E$, then $L \sqsubseteq_{\mathcal{T}} E$.*

**Proof.** By induction on $rd(E)$. First, if $rd(E) = 0$, then $E = A_1 \sqcap \ldots \sqcap A_m$ for $A_i \in N_C$. Since $C \sqsubseteq_{\mathcal{T}} E$ and $D \sqsubseteq_{\mathcal{T}} E$, then $C \sqsubseteq A_i$ and $D \sqsubseteq A_i$ for all $i, 1 \leq i \leq m$. But then $A_i \in S_0(A,0) \cap S_0(B,0)$ for all $i$. This implies that $L \sqsubseteq_{\mathcal{T}} \bigsqcap_{C' \in S_0(A,0)\cap S_0(B,0)} C' \sqsubseteq_{\mathcal{T}} E$.

Let now $rd(E) > 0$ and let $\exists r.E'$ be a (top level) conjunct of $E$, where $E'$ is not of the form $P_*F$. Then, as $C \sqsubseteq_{\mathcal{T}} E$ and $D \sqsubseteq_{\mathcal{T}} E$, there must exist $A' \in S_0(A,r,0)$ and $B' \in S_0(B,r,0)$ such that $A' \sqsubseteq_{\mathcal{T}} E'$ and $B' \sqsubseteq_{\mathcal{T}} E'$. By induction hypothesis, we then have that $\exists r.\mathsf{k\text{-}lcs}(A',B',\mathcal{T},k-1) \sqsubseteq_{\mathcal{T}} E'$. Let now $P_{>0}(\exists r.E')$ be a conjunct in $E$. Then there must exist $A' \in S_0^{>0}(A,r)$ and $B' \in S_0^{>0}(B,r)$ such that $A' \sqsubseteq_{\mathcal{T}} E'$ and $B' \sqsubseteq_{\mathcal{T}} E'$. But then, we have that $P_{>0}(\exists r.\mathsf{k\text{-}lcs}(A',B',\mathcal{T},k-1)) \sqsubseteq_{\mathcal{T}} P_{>0}(\exists r.E')$. Analogously, we can prove that for every conjunct $P_{=1}(\exists r.E')$ in $E$, there exist $A' \in S_0(A,r,1)$ and $B' \in S_0(B,r,1)$ such that $A' \sqsubseteq_{\mathcal{T}} E'$ and $B' \sqsubseteq_{\mathcal{T}} E'$, and hence $P_{=1}(\exists r.\mathsf{k\text{-}lcs}(A',B',\mathcal{T},k-1)) \sqsubseteq_{\mathcal{T}} P_{=1}(\exists r.E')$. All this together implies that $L \sqsubseteq_{\mathcal{T}} E$. ❑

The following theorem is then a trivial consequence of Lemmas 14 and 15.

**Theorem 16** *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 $\mathsf{k\text{-}lcs}(C,D,\mathcal{T},k) \equiv_{\mathcal{T}}$ k-lcs$(C,D)$.*

Similar to the completion algorithm, the extension of the algorithm to compute the role-bounded lcs in $\mathcal{EL}$ to the "light-weight" probabilistic DL Prob-$\mathcal{EL}^{01}$ is fairly straight-forward.

## 5 Summary and Outlook

In this report we presented a practical approach for computing least common subsumers of limited role-depth in the DL $\mathcal{EL}$ and the moderately probabilistic DL Prob-$\mathcal{EL}^{01}$. It turns out that the complexity of computing the role-depth limited lcs is the same for concept descriptions.

Althought simply limiting the role-depth of the result concept description is not elegant from the theoretical perspective, we argue that it will turn out to be sufficient for many practical applications.

The computation algorithm k-lcs can be easily be implemented on top of the completion algorithm. In case a *k-lcs* turns out to be 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. It remains an interesting question whether the completion sets obtained from the classification of the initial TBox can simply be extended, when $C \equiv A$ and $D \equiv B$ are added.

In our future work we will try to extend the computation algorithm for the role-depth limited lcs to full Prob-$\mathcal{EL}$.

## References

[1] M.-A. Aufaure and H. Hajji. Semantic structuration of image annotations: A data mining approach. In *Proc of the International Workshop on Multimedia Information Systems (MIS'02)*, pages 38–47, 2002.

[2] F. Baader. Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In G. Gottlob and T. Walsh, editors, *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI-03)*, pages 325–330. Morgan Kaufmann, 2003.

[3] F. Baader. Terminological cycles in a description logic with existential restrictions. In G. Gottlob and T. Walsh, editors, *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI-03)*, pages 319–324. Morgan Kaufmann, 2003.

[4] 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)*, Edinburgh, UK, 2005. Morgan-Kaufmann Publishers.

[5] 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.

[6] F. Baader, R. Küsters, and R. Molitor. Computing least common subsumer in description logics with existential restrictions. In T. Dean, editor, *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99)*, pages 96–101, Stockholm, Sweden, 1999. Morgan Kaufmann, Los Altos.

[7] F. Baader, B. Sertkaya, and A.-Y. Turhan. Computing the least common subsumer w.r.t. a background terminology. *Journal of Applied Logics*, 2007.

[8] S. Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In R. López de Mantáras and L. Saitta, editors, *Proc. of the 16th European Conf. on Artificial Intelligence (ECAI-04)*, pages 298–302. IOS Press, 2004.

[9] S. Brandt. Reasoning in $\mathcal{ELH}$ w.r.t. general concept inclusion axioms. LTCS-Report LTCS-04-03, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2004. See http://lat.inf.tu-dresden.de/research/reports.html.

[10] S. Brandt. *Standard and Non-standard Reasoning in Description Logics.* PhD thesis, Institute for theoretical computer science, TU Dresden, January 2006.

[11] W. W. Cohen, A. Borgida, and H. Hirsh. Computing least common subsumers in description logics. In W. Swartout, editor, *Proc. of the 10th Nat. Conf. on Artificial Intelligence (AAAI-92)*, pages 754–760, San Jose, CA, 1992. AAAI Press/The MIT Press.

[12] W. W. Cohen and H. Hirsh. Learnability of description logics with equality constraints. *ML*, 17(2/3):169–200, 1994.

[13] W. W. Cohen and H. Hirsh. Learning the CLASSIC description logics: Theoretical and experimental results. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proc. of the 4th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-94)*, pages 121–133, Bonn, 1994. Morgan Kaufmann, Los Altos.

[14] P. Klinov, B. Parsia, and U. Sattler. On correspondences between probabilistic first-order and description logics. In B. Cuenca Grau, I. Horrocks, B. Motik, and U. Sattler, editors, *Proc. of the 2008 Description Logic Workshop (DL 2009)*, volume 477 of *CEUR Workshop*, 2009.

[15] T. Lukasiewicz. Expressive probabilistic description logics. *Artificial Intelligence*, 172(6-7):852–883, 2008.

[16] C. Lutz and L. Schröder. Probabilistic description logics for subjective probabilities. In F. Lin and U. Sattler, editors, *Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-10)*, 2010. To appear.

[17] Th. Mantay, R. Möller, and A. Kaplunova. Computing probabilistic least common subsumers in description logics. In *Proceedings KI-99, 23. Deutsche Jahrestagung für Künstliche Intelligenz*, pages 89–100. Springer-Verlag, 1999.

[18] R. Möller, V. Haarslev, and B. Neumann. Semantics-based information retrieval. In *Proceedings IT&KNOWS-98: International Conference on Information Technology and Knowledge Systems*, pages 49–6, Vienna, Budapest, 1998.

[19] 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.