TECHNISCHE UNIVERSITÄT DRESDEN

Dresden University of Technology
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

Completion-based computation of most specific concepts 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 10-03

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

http://lat.inf.tu-dresden.de

Visiting Address:
Nöthnitzer Str. 46
Dresden

Completion-based computation of most specific concepts with limited role-depth for $\mathcal{EL}$ and Prob-$\mathcal{EL}^{01}$

Abstract

In Description Logics the reasoning service most specific concept (msc) constructs a concept description that generalizes an ABox individual into a concept description. For the Description Logic $\mathcal{EL}$ the msc may not exist, if computed with respect to general $\mathcal{EL}$-TBoxes or cyclic ABoxes. However, it is still possible to find a concept description that is the msc up to a fixed role-depth, i.e. with respect to a maximal nesting of quantifiers. In this report we present a practical approach for computing the role-depth bounded msc, based on the polynomial-time completion algorithm for $\mathcal{EL}$. We extend these methods to Prob-$\mathcal{EL}_c^{01}$, which is a probabilistic variant of $\mathcal{EL}$. Together with the companion report [9] this report devises computation methods for the bottom-up construction of knowledge bases for $\mathcal{EL}$ and Prob-$\mathcal{EL}_c^{01}$.

1 Introduction

In Description Logics the inference *most specific concept* (msc) constructs a concept description that generalizes an individual into a concept description. For the Description Logic $\mathcal{EL}$ the msc needs not exist [2], if computed with respect to general $\mathcal{EL}$-TBoxes. However, it is still possible to find a concept description that is the msc up to a fixed role-depth. In this report we present a practical approach for computing the role-depth bounded msc, based on the polynomial-time completion algorithm for $\mathcal{EL}$. We extend this method to a simple probabilistic variant of $\mathcal{EL}$ that can express subjective probabilities and that was recently introduced in [8]. The probabilistic DL that we use, called Prob-$\mathcal{EL}_c^{01}$, allows only a fairly limited use of uncertainty. More precisely, it is only possible to express that a concept *may* hold ($P_{>0}C$), or that it holds *almost surely* ($P_{=1}C$). Despite its limited expressivity, this logic is interesting due to its nice algorithmic properties; as shown in [8], subsumption can be decided in polynomial time and instance checking can be performed in polynomial time as well.

Many practical applications that need to represent probabilistic information, such as medical applications or context-aware applications, need to characterize that observations only hold with certain probability. Furthermore, these applications face the problem that information from different sources does not coincide or that different diagnoses yield differing results. These applications need to "integrate" differing observations for the same state of affairs. A way to determine what the different information sources agree upon is to represent this information as ABox individuals and to find a common generalization of these individuals. A description of such a generalization of a group of ABox individuals can be obtained by applying the so-called *bottom-up approach* for constructing knowledge bases [5]. In this approach a set of individuals is generalized into a single concept description by first generating the msc of each concept and then apply the least common subsumer (lcs) to the set of obtained concept descriptions to extract their commonalities.

The second step, i.e., a computation procedure for the approximative lcs has been investigated for $\mathcal{EL}$ and Prob-$\mathcal{EL}_c^{01}$ in [10]. In this report we present a similar procedure for the msc. We devise a practical algorithm for computing the msc up to a certain role-depth for $\mathcal{EL}$ and Prob-$\mathcal{EL}_c^{01}$. The so-called *k-msc* obtained by the algorithm is still a generalization of the input, but not necessarily the least one – in this sense it is only an approximation of the msc. Moreover, our algorithms are based upon the completion algorithms for $\mathcal{EL}$ and Prob-$\mathcal{EL}_c^{01}$, and thus can be easily implemented on top of reasoners of these DLs.

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*, and a set N_I of *individual 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:*

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

The semantics of a concept description is defined in terms of an *interpretation* $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$. The domain Δ 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 $P^{\mathcal{I}} \subseteq \Delta$, each role name $r \in N_R$ to a binary relation $r^{\mathcal{I}} \subseteq \Delta \times \Delta$, and each individual name $a \in N_I$ to an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. This function is extended to arbitrary $\mathcal{EL}$-concept descriptions 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 $C_1 \sqsubseteq C_1$.

In the ABox individuals can be characterized by concepts and relations between individuals can be stated.

Definition 4 (assertion, ABox) *Let a, b be individual names, r a role name and C be a concept description, then*

- $C(a)$ *is a* concept assertion, *and*
- $r(a, b)$ *is a* role assertion.

An ABox $\mathcal{A}$ is finite set of assertions.

An interpretation $\mathcal{I}$ satisfies an assertion $C(a)$, iff $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and an assertion $r(a, b)$ is satisfied, iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. An interpretation is a model *of an ABox $\mathcal{A}$, if all assertions $\mathcal{A}$ contains are satisfied.*

If the concept assertions in the ABox $\mathcal{A}$ contain only $\mathcal{EL}$-concept descriptions, we call $\mathcal{A}$ an *$\mathcal{EL}$-ABox*. A *Knowledge Base* (KB) is a tuple $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consisting of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$.

Based on the semantics of concept descriptions, TBoxes and ABoxes a number of inference problems for DLs have been defined. Some of the most relevant ones are the following:

- *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 $\mathcal{T}$.
- *ABox consistency.* An ABox $\mathcal{A}$ is *consistent* w.r.t. a TBox $\mathcal{T}$ if $\mathcal{A}$ and $\mathcal{T}$ have a common model.
- *The instance problem.* For a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ an individual name a is an *instance* of a concept C in an ABox $\mathcal{A}$ w.r.t. a TBox $\mathcal{T}$ (written $\mathcal{K} \models C(a)$) if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for every common model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{T}$.
- *ABox realization problem.* For an KB $\mathcal{K}$ with ABox $\mathcal{A}$ (and a TBox $\mathcal{T}$) the realization problem computes for each individual a in $\mathcal{A}$ the set of named concepts from $\mathcal{K}$ that have a as an instance and that is least (w.r.t. subsumption).

In this report we will use the instance problem. However, as we will see in Section 3, the completion algorithm computes in fact ABox realization. Also based on the instance relation the most specific concept is defined.

Definition 5 (most specific concept, msc) *Let $\mathcal{L}$ be a DL, $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a $\mathcal{L}$-KB. The* most specific concept *of an individual a from $\mathcal{A}$ is the $\mathcal{L}$-concept description C such that*

- $\mathcal{K} \models C(a)$, *and*
- *for each $\mathcal{L}$-concept description D $\mathcal{K} \models D(a)$ implies $C \sqsubseteq_{\mathcal{T}} D$.*

The msc depends on the DL in use. For the DL $\mathcal{FL}^{-}$ that only offers conjunction as concept constructor the msc always exists for cyclic ABoxes and TBoxes. For $\mathcal{EL}$ the msc does not need to exist if computed w.r.t. a cyclic ABox and an empty TBox as it was shown in [7].

2.2 Prob-$\mathcal{EL}_c^{01}$

We now introduce Prob-$\mathcal{EL}_c^{01}$, a probabilistic variant of $\mathcal{EL}$ that allows reasoning with limited uncertainty through probabilistic concepts. In this logic, it is only possible to express that a concept may hold (i.e., holds with probability greater than 0), or that it holds almost surely (that is, with probability 1). This probabilistic logic was first introduced in [8].

The probabilistic DL Prob-$\mathcal{EL}_c^{01}$ extends $\mathcal{EL}$ with the constructors $P_{>0}$ and $P_{=1}$. That is, Prob-$\mathcal{EL}$ concepts are constructed as

$$C ::= \top \mid A \mid C \sqcap D \mid \exists r.C \mid P_{>0}C \mid P_{=1}C,$$

where A is a concept name and r a role name. The intuition behind these last two expressions is that C holds with probability greater than 0 or equal to 1, respectively.

The semantics of Prob-$\mathcal{EL}_c^{01}$ generalizes 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*, 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 $\Delta^{\mathcal{I}}$. Additionally, it must hold that for every individual name $a \in N_I$ and every two worlds $w, w' \in W$: $a^{\mathcal{I}_w} = a^{\mathcal{I}_{w'}}$.

This last restriction expresses that named individuals must be interpreted alike in all worlds of a probabilistic interpretation. Thus, we can use the expression $a^{\mathcal{I}}$ without ambiguity.

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 constructors P_* is defined as

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

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 $\mathcal{T}$.

The relevant decision problems introduced previously for $\mathcal{EL}$ can also be defined in an analogous manner for Prob-$\mathcal{EL}_c^{01}$. Of particular interest in this report is the instance problem, that asks whether an individual name a must be interpreted as an instance of a concept C in every model of a given ABox and TBox.

This simple probabilistic logic retains the good complexity properties held by $\mathcal{EL}$. Moreover, as it will be shown in the following section, reasoning in Prob-$\mathcal{EL}_c^{01}$ can be performed through a variant of the completion algorithm for $\mathcal{EL}$.

3 Completion-based Instance Checking Algorithms

We briefly sketch the completion algorithms for instance checking in $\mathcal{EL}$ [3] and Prob-$\mathcal{EL}_c^{01}$ [8].

3.1 Instance checking in $\mathcal{EL}$

Assume we want to test for an $\mathcal{EL}$-KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ whether $\mathcal{K} \models D(a)$ holds. The completion algorithm first augments the knowledge base by introducing a concept name for the complex concept description D from the instance check. More precisely, it sets $\mathcal{K} = (\mathcal{T} \cup \{A_q \equiv D\}, \mathcal{A})$, where A_q is a new concept name not appearing in $\mathcal{K}$. The instance checking algorithm for $\mathcal{EL}$ works on normalized knowledge bases.

Normalization of the $\mathcal{EL}$ KB

Normalization is done two steps: first the ABox is transformed into a simple ABox.

An ABox is a *simple ABox*, if it only contains concept names in concept assertions. An $\mathcal{EL}$-ABox $\mathcal{A}$ can be transformed into a simple ABox by the following two steps:

1. replace each complex assertion $C(A)$ in $\mathcal{A}$ by $A(a)$ with a fresh name A
2. and introduce $A \equiv C$ in the TBox.

This normalization step of naming complex concept descriptions is performed first before completion. After this step the (now already augmented) TBox is normalized. By $\mathsf{BC}_\mathcal{T}$ we denote the set of *basic concept descriptions for an $\mathcal{EL}$-TBox $\mathcal{T}$*, i.e., the smallest set of concept descriptions which contains $\top$ and all concept names used in $\mathcal{T}$. Based on this, a normal form for TBoxes can be defined as follows.

Definition 6 (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 $C_1, C_2 \in \mathsf{BC}_\mathcal{T}$*

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

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 $\mathcal{EL}$-TBox can be transformed into normal form by introducing new concept names and 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. This transformation can be done in linear time.

For a concept description C let $\mathsf{CN}(C)$ denote the set of all concept names and $\mathsf{RN}(C)$ denote the set of all role names that appear in C. The *signature of a concept description* C (denoted sig(C)) is $\mathsf{CN}(C) \cup \mathsf{RN}(C)$. Similarly, the set of concept names that appear in a TBox are denoted by $\mathsf{CN}(\mathcal{T})$ and role names by $\mathsf{RN}(\mathcal{T})$. The *signature of a TBox* $\mathcal{T}$ (denoted $\mathsf{sig}(\mathcal{T})$) is $\mathsf{CN}(\mathcal{T}) \cup \mathsf{RN}(\mathcal{T})$. The *signature of an ABox* $\mathcal{A}$ (denoted $\mathsf{sig}(\mathcal{A})$) is the set of concept names $\mathsf{CN}(\mathcal{A})$, role names $\mathsf{RN}(\mathcal{A})$ and individual names $\mathsf{IN}(\mathcal{A})$ that appear in $\mathcal{A}$. The signature of a KB $\mathcal{K}$ (denoted $\mathsf{sig}(\mathcal{K})$) is the set of concept and role names that appear in $\mathcal{T}$ and $\mathcal{A}$.

Clearly, for a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ the signature of $\mathcal{A}$ may be changed only during the first of the two normalization steps and the signature of $\mathcal{T}$ may be extended during both of them. However, since the first normalization step just introduces new names for complex concept descriptions appearing in concept assertions, it does not affect instance relations w.r.t. the signature of the original KB. The normalization of the TBox does not affect instance tests for $\mathcal{EL}$-concept descriptions formulated w.r.t. the signature of the original KB $\mathcal{K}$ as well.

The completion algorithm for instance checking is based on the one for classifying $\mathcal{EL}$-TBoxes introduced in [3]. The completion algorithm constructs a representation of the minimal model of $\mathcal{K}$. Let $\mathcal{K}$ =($\mathcal{T}$, $\mathcal{A}$) be an $\mathcal{EL}$-KB with a simple

CR1 If $C \in S(X)$, $C \sqsubseteq D \in \mathcal{T}$, and $D \notin S(X)$ then $S(X) := S(X) \cup \{D\}$

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

CR3 If $C \in S(X)$, $C \sqsubseteq \exists r.D \in \mathcal{T}$, and $D \notin S(X, r)$ then $S(X, r) := S(X, r) \cup \{D\}$

CR4 If $C \in S(X, r)$, $D \in S(C)$, $\exists r.C \sqsubseteq D \in \mathcal{T}$, and $D \notin S(X)$ then $S(X) := S(X) \cup \{D\}$

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

ABox $\mathcal{A}$ and $\mathcal{T}$ in normal form. The completion algorithm works on four kinds of *completion sets*: $S(a), S(a, r), S(C)$ and $S(C, r)$ for $a \in \mathsf{IN}(\mathcal{A})$, $C \in \mathsf{CN}(\mathcal{K})$, and $r \in \mathsf{RN}(\mathcal{K})$. These sets contain concept names from $\mathsf{CN}(\mathcal{K})$. Let D, C be concept names, r be a role name and a be an individual name. Intuitively, the completion rules make implicit subsumption and instance relationships explicit in the following sense:

- $D \in S(C)$ implies that $C \sqsubseteq_{\mathcal{T}} D$,
- $D \in S(C, r)$ implies that $C \sqsubseteq_{\mathcal{T}} \exists r.D$.
- $D \in S(a)$ implies that a is an instance of D w.r.t. $\mathcal{K}$,
- $D \in S(a, r)$ implies that a is an instance of $\exists r.D$ w.r.t. $\mathcal{K}$.

$\mathsf{S}_{\mathcal{K}}$ denotes the set of all completion sets of $\mathcal{K}$. The completion sets are initialized for each $a \in \mathsf{IN}(\mathcal{A})$ and each $C \in \mathsf{CN}(\mathcal{K})$ as follows:

- $S(C) := \{C, \top\}$ for each $C \in \mathsf{CN}(\mathcal{K})$,
- $S(C, r) := \emptyset$ for each $r \in \mathsf{RN}(\mathcal{K})$,
- $S(a) := \{C \in \mathsf{CN}(\mathcal{A}) \mid C(a) \text{ appears in } \mathcal{A}\} \cup \{\top\}$, and
- $S(a, r) := \{b \in \mathsf{IN}(\mathcal{A}) \mid r(a, b) \text{ appears in } \mathcal{A}\}$ for each $r \in \mathsf{RN}(\mathcal{K})$.

Then these sets are extended by applying the completion rules shown in Figure 2 until no more rule applies. In these rules X and Y can refer to concept or individual names.

After the completion has terminated, the following relations hold:

- subsumption relation between two named concepts A and B from $\mathcal{K}$ holds iff $B \in S(A)$
- instance relation between an individual a and a named concept B from $\mathcal{K}$ holds iff $B \in S(a)$,

as shown in [3]. Hence, to decide the initial query $\mathcal{K} \models D(a)$, one has to test whether A_q appears in $S(a)$. In fact, instance queries for all individuals and all named concepts from the KB can be answered now; the completion algorithm does not only perform one instance checking, but ABox realization.

The completion algorithm runs in polynomial time in size of the knowledge base.

3.2 Completion Algorithms for Prob-$\mathcal{EL}_c^{01}$

To describe the completion algorithm for Prob-$\mathcal{EL}_c^{01}$, we need the notion of basic concepts. The set $\mathsf{BC}_\mathcal{T}$ of Prob-$\mathcal{EL}_c^{01}$ *basic concepts* for a KB $\mathcal{K}$ is the smallest set that contains (1) $\top$, (2) all concept names used in $\mathcal{K}$, and (3) all concepts of the form P_*A, where A is a concept name in $\mathcal{K}$. A Prob-$\mathcal{EL}_c^{01}$-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 concept name. The normalization rules in Figure 1 can also be used to transform a Prob-$\mathcal{EL}_c^{01}$-TBox into this extended notion of normal form. We further assume that for all assertions $C(a)$ in the ABox $\mathcal{A}$, C is a concept name. We denote as $\mathcal{P}_0^\mathcal{T}$, $\mathcal{P}_1^\mathcal{T}$ and $\mathcal{R}_0^\mathcal{T}$ the set of all concepts of the form $P_{>0}A, P_{=1}A$, and $P_{>0}r(a,b)$ respectively, occurring in a normalized knowledge base $\mathcal{K}$.

The completion algorithm for Prob-$\mathcal{EL}_c^{01}$ 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 $V := \{0, \varepsilon, 1\} \cup \mathcal{P}_0^\mathcal{T} \cup \mathcal{R}_0^\mathcal{T}$, 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 individual name a, concept name A, role name r and world v, we store the completion sets $S_0(A, v), S_\varepsilon(A, v), S_0(A, r, v)$, $S_\varepsilon(A, r, v)$, $S(a, v)$, and $S(a, r, v)$.

The algorithm initializes the sets as follows for every $A \in \mathsf{BC}_\mathcal{T}, r \in \mathsf{RN}(\mathcal{K})$, and $a \in \mathsf{IN}(\mathcal{A})$:

- $S_0(A, 0) = \{\top, A\}$ 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 $v \in V \setminus \{\varepsilon\}$,

PR1 If $C' \in S_*(X,v)$, $C' \sqsubseteq D \in \mathcal{T}$, and $D \notin S_*(X,v)$ then $S_*(X,v) := S_*(X,v) \cup \{D\}$

PR2 If $C_1, C_2 \in S_*(X,v)$, $C_1 \sqcap C_2 \sqsubseteq D \in \mathcal{T}$, and $D \notin S_*(X,v)$ then $S_*(X,v) := S_*(X,v) \cup \{D\}$

PR3 If $C' \in S_*(X,v)$, $C' \sqsubseteq \exists r.D \in \mathcal{T}$, and $D \notin S_*(X,r,v)$ then $S_*(X,r,v) := S_*(X,r,v) \cup \{D\}$

PR4 If $D \in S_*(X,r,v)$, $D' \in S_{\gamma(v)}(D,\gamma(v))$, $\exists r.D' \sqsubseteq E \in \mathcal{T}$, and $E \notin S_*(X,v)$ then $S_*(X,v) := S_*(X,v) \cup \{E\}$

PR5 If $P_{>0}A \in S_*(X,v)$, and $A \notin S_*(X, P_{>0}A)$ then $S_*(X, P_{>0}A) := S_*(X, P_{>0}A) \cup \{A\}$

PR6 If $P_{=1}A \in S_*(X,v)$, $v \neq 0$, and $A \notin S_*(X,v)$ then $S_*(X,v) := S_*(X,v) \cup \{A\}$

PR7 If $A \in S_*(X,v)$, $v \neq 0$, $P_{>0}A \in \mathcal{P}_0^{\mathcal{T}}$, and $P_{>0}A \notin S_*(X,v')$ then $S_*(X,v') := S_*(X,v') \cup \{P_{>0}A\}$

PR8 If $A \in S_*(X,1)$, $P_{=1}A \in \mathcal{P}_1^{\mathcal{T}}$, and $P_{=1}A \notin S_*(X,v)$ then $S_*(X,v) := S_*(X,v) \cup \{P_{=1}A\}$

PR9 If $r(a,b) \in \mathcal{A}, C \in S(b,0), \exists r.C \sqsubseteq D \in \mathcal{T}$, and $D \notin S(a,0)$ then $S(a,0) := S(a,0) \cup \{D\}$

PR10 If $P_{>0}r(a,b) \in \mathcal{A}$, $C \in S(b, P_{>0}r(a,b)), \exists r.C \sqsubseteq D \in \mathcal{T}$, and $D \notin S(a, P_{>0}r(a,b))$ then $S(a, P_{>0}r(a,b)) := S(a, P_{>0}r(a,b)) \cup \{D\}$

PR11 If $P_{=1}r(a,b) \in \mathcal{A}$, $C \in S(b,v)$ with $v \neq 0$, $\exists r.C \sqsubseteq D \in \mathcal{T}$ and $D \notin S(a,v)$ then $S(a,v) := S(a,v) \cup \{D\}$

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

- $S(a,0) = \{\top\} \cup \{A \mid A(a) \in \mathcal{A}\}$, $S(a,v) = \{\top\}$ for all $v \neq 0$,
- $S_0(A,r,v) = S_\varepsilon(A,r,v) = \emptyset$ for all $v \in V$, $S(a,r,v) = \emptyset$ for $v \neq 0$,
- $S(a,r,0) = \{b \in \mathsf{IN}(\mathcal{A}) \mid r(a,b) \in \mathcal{A}\}$.

These sets are then extended by exhaustively applying the rules shown in Figure 3, where X ranges over $\mathsf{BC}_{\mathcal{T}} \cup \mathsf{IN}(\mathcal{A})$, $S_*(X,v)$ stands for $S(X,v)$ if X is an individual and for $S_0(X,v), S_\varepsilon(X,v)$ if $X \in \mathsf{BC}_{\mathcal{T}}$, and $\gamma : V \to \{0, \varepsilon\}$ is defined by $\gamma(0) = 0$, and $\gamma(v) = \varepsilon$ for all $v \in V \setminus \{0\}$.

This algorithm terminates in polynomial time. After termination, the completion sets store all the information necessary to decide subsumption of concept names, as well as checking whether an individual is an instance of a given concept name [8]. For the former decision, it holds that for every pair A, B of concept names: $B \in S_0(A,0)$ iff $A \sqsubseteq_{\mathcal{K}} B$. In the case of instance checking, we have that $\mathcal{K} \models A(a)$ iff $A \in S(a,0)$.

4 Computing the k-MSC using Completion

The msc was first investigated for $\mathcal{EL}$-concept descriptions and w.r.t. unfoldable TBoxes and possibly cyclic ABoxes in [7]. It was shown that the msc does not need to exist for cyclic ABoxes. Consider the ABox $\mathcal{A} = \{r(a,a), C(a)\}$. The msc of a 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. For cyclic TBoxes it has been shown in [2] that the msc does not need to exists even if the ABox is acyclic.

To avoid infinite nestings in presence of cyclic ABoxes it was proposed in [7] to limit the role-depth of the concept description to be computed. This limitation yields an approximation of the msc, which is still a concept description with the input individual as an instance, but it does not need to be the least one (w.r.t. subsumption) with this property. We follow this idea to compute approximative msc also in presence of general TBoxes.

The *role-depth* of a concept description C (denoted $rd(C)$) is the maximal number of nested quantifiers of C. Now we can define the msc with limited role-depth for $\mathcal{EL}$.

Definition 7 (role-depth bounded $\mathcal{EL}$-msc) *Let $\mathcal{K}$ =($\mathcal{T}$, $\mathcal{A}$) be an $\mathcal{EL}$-KB and a an individual in $\mathcal{A}$ and $k \in \mathbb{N}$. Then the $\mathcal{EL}$-concept description C is the* role-depth bounded $\mathcal{EL}$-most specific concept *of a w.r.t. $\mathcal{K}$ and role-depth k (written k-$msc_{\mathcal{K}}(a)$) iff*

1. *$rd(C) \leq k$,*
2. *$\mathcal{K} \models C(a)$, and*
3. *for all $\mathcal{EL}$-concept descriptions E with $rd(E) \leq k$ holds: $\mathcal{K} \models E(a)$ implies $C \sqsubseteq_{\mathcal{T}} E$.*

Notice that in case the exact msc has a role-depth less than k the role-depth bounded msc is the exact msc.

4.1 Computing the k-*msc* in $\mathcal{EL}$ by completion

The computation of the msc relies on a characterization of the instance relation. While in earlier works this was given by homomorphism [7] or simulations [2] between graph representations of the knowledge base and the concept in question, we use the completion algorithm as such a characterization. Furthermore, we construct the msc by traversing the completion sets to "collect" the msc. More precisely, the set of completion sets encodes a graph structure, where the sets

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

Procedure $\mathsf{k\text{-}msc}$ $(a, \mathcal{K}, k)$
Input: a: individual from $\mathcal{K}$; $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ an $\mathcal{EL}$-KB; $k \in \mathbb{N}$
Output: role-depth bounded $\mathcal{EL}$-msc of a w.r.t. $\mathcal{K}$ and k.

$(\mathcal{T}', \mathcal{A}') := \mathsf{simplify\text{-}ABox}(\mathcal{T}, \mathcal{A})$
$\mathcal{K}'' := (\mathsf{normalize}(\mathcal{T}'), \mathcal{A}')$
$\mathsf{S}_{\mathcal{K}} := \mathsf{apply\text{-}completion\text{-}rules}(\mathcal{K})$
return $\mathsf{Remove\text{-}normalization\text{-}names}$ ($\mathsf{traversal\text{-}concept\text{-}i}\,(a, \mathsf{S}_{\mathcal{K}}, k)$)

Procedure $\mathsf{traversal\text{-}concept\text{-}i}$ (a, S, k)
Input: a: individual name from $\mathcal{K}$; S: set of completion sets; $k \in \mathbb{N}$
Output: role-depth traversal concept (w.r.t. $\mathcal{K}$) and k.

if $k = 0$ **then return** $\bigsqcap_{A \in S(a)} A$
else return $\bigsqcap_{A \in \mathsf{S}(a)} A \sqcap$
$$\bigsqcap_{r \in \mathsf{RN}(\mathcal{K}'')} \; \bigsqcap_{A \in \mathsf{CN}(\mathcal{K}'') \cap S(a,r)} \exists r.\ \mathsf{traversal\text{-}concept\text{-}c}\ (A, \mathsf{S}, k-1) \ \sqcap$$
$$\bigsqcap_{r \in \mathsf{RN}(\mathcal{K}'')} \; \bigsqcap_{b \in \mathsf{IN}(\mathcal{K}'') \cap S(a,r)} \exists r.\ \mathsf{traversal\text{-}concept\text{-}i}\ (b, \mathsf{S}, k-1)$$
end if

Procedure $\mathsf{traversal\text{-}concept\text{-}c}$ (A, S, k)
Input: A: concept name from $\mathcal{K}''$; S: set of completion sets; $k \in \mathbb{N}$
Output: role-depth bounded traversal concept.

if $k = 0$ **then return** $\bigsqcap_{B \in S(A)} B$
else return $\bigsqcap_{B \in S(A)} B \sqcap \bigsqcap_{r \in \mathsf{RN}(\mathcal{K}'')} \bigsqcap_{B \in S(A,r)} \exists r.\mathsf{traversal\text{-}concept\text{-}c}\ (B, \mathsf{S}, k-1)$
end if

$S(X)$ are the nodes and the sets $S(X, r)$ encode the edges. Traversing this graph structure, one can construct an $\mathcal{EL}$-concept. To obtain a finite concept in the presence of cyclic ABoxes or TBoxes one has to limit the role-depth of the concept to be obtained.

Definition 8 (traversal concept) *Let $\mathcal{K}$ be an $\mathcal{EL}$-KB, $\mathcal{K}''$ be its normalized form, $\mathsf{S}_{\mathcal{K}}$ the completion set obtained from $\mathcal{K}$ and $k \in \mathbb{N}$. Then the* traversal concept of a named concept *A (denoted k-$\mathsf{C}_{\mathsf{S}_{\mathcal{K}}}(A)$) with $\mathsf{sig}(A) \subseteq \mathsf{sig}(\mathcal{K}'')$ is the concept obtained from executing the procedure call* $\mathsf{traversal\text{-}concept\text{-}c}$*$(A, \mathsf{S}_{\mathcal{K}}, k)$ shown in Algorithm 1.*

The traversal concept of an individual *a (denoted k-$\mathsf{C}_{\mathsf{S}_{\mathcal{K}}}(a)$) with $a \subseteq \mathsf{sig}(\mathcal{K})$ is the concept description obtained from executing the procedure call* $\mathsf{traversal\text{-}concept\text{-}i}$*$(a, \mathsf{S}_{\mathcal{K}}, k)$ shown in Algorithm 1.*

The idea is that the traversal concept of an individual yields its msc. However, the traversal concept contains names from $\mathsf{sig}(\mathcal{K}'') \setminus \mathsf{sig}(\mathcal{K})$, i.e., concept names that were introduced during normalization – we call this kind of concept names *normalization names* in the following. The returned msc should be formulated w.r.t. the signature of the original KB, thus the normalization names need to be removed or replaced.

Lemma 9 *Let $\mathcal{K}$ be an $\mathcal{EL}$-KB, $\mathcal{K}''$ its normalized version, $\mathsf{S}_\mathcal{K}$ be the set of completion sets obtained for $\mathcal{K}$, $k \in \mathbb{N}$ a natural number and $a \in \mathsf{IN}(\mathcal{K})$. Furthermore let $C = k\text{-}\mathsf{C}_{\mathsf{S}_\mathcal{K}}(a)$ and $\widehat{C}$ be obtained from C by removing the normalization names. Then*

$$\mathcal{K}'' \models C(a) \text{ iff } \mathcal{K} \models \widehat{C}(a).$$

Proof. The *only-if* direction is trivial since C is constructed by a conjunction of concepts. Removing some conjuncts will only produce a more general concept; that is, $C \sqsubseteq \widehat{C}$ holds.

For the other direction, assume that $\mathcal{K}'' \not\models C(a)$. Then, there must be a concept of the form $D := \exists r_1.\exists r_2.\ldots\exists r_m.B$ with B a concept name from $\mathcal{K}''$ such that $\mathcal{K}'' \not\models D(a)$. If B is not a normalization name, then it follows directly that $\mathcal{K}'' \not\models C(a)$. Otherwise, by construction, there must be a concept D' such that, either $D' := \exists r_1.\exists r_2.\ldots\exists r_m.B'$ and $B \in S(B')$ or $D' := \exists r_1.\exists r_2.\ldots\exists r_{m-1}.B'$ and $B \in S(B', r_m)$. This approach can be repeated until we find such a D' where B is not a normalization name, or we reach an individual name. In any of both cases, it follows that $\mathcal{K}'' \not\models C(a)$. Since the normalization rules preserve instances w.r.t. the original concept names, we can then deduce that $\mathcal{K} \not\models C(a)$. ∎

This lemma guarantees that removing the normalization names from the traversal concept preserves the instance relationships.

We now ready to describe a computation algorithm for the role-depth bounded msc. This is the procedure k-msc as displayed in Algorithm 1.

The procedure k-msc has an individual a from a knowledge base $\mathcal{K}$, the knowledge base $\mathcal{K}$ itself and number k for the role depth-bound as parameter. It first performs the two normalization steps on $\mathcal{K}$, then applies the completion rules from Figure 2 to the normalized KB $\mathcal{K}''$ and stores the set of completion sets in $\mathsf{S}_\mathcal{K}$. Afterwards it computes the traversal-concept of a from $\mathsf{S}_\mathcal{K}$ w.r.t. role-depth bound k. In a post-processing step it applies Remove-normalization-names to the traversal concept.

Obviously, the concept description returned from the procedure k-msc has a role-depth less or equal to k. Thus the first condition of Definition 7 is fulfilled. We prove next that the concept description obtained from k-msc fulfills the second condition from Definition 7.

Lemma 10 *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{EL}$-KB and a an individual in $\mathcal{A}$ and $k \in \mathbb{N}$. If $C = \mathsf{k\text{-}msc}(a, \mathcal{K}, k)$, then $\mathcal{K} \models C(a)$.*

Proof. The proof is an easy induction on k. For the base case, when $k = 0$, C is the conjunction of all concept names in $S(a)$. By the properties of the completion algorithm, we know that for every $A \in S(a)$ it holds that $\mathcal{K} \models A(a)$. Thus, $\mathcal{K} \models \bigsqcap_{A \in S(a)} A(a) = C(a)$.

For the induction step, the algorithm simply adds the elements of a completion set either from an individual (if there is an assertion of the form $r(a, b)$) or from a concept, by means of the completion set $S(a, r)$. Again, by the properties of the completion algorithm, it holds that for every $A \in S(a, r)$, $\mathcal{K} \models \exists r.A(a)$. Thus, it follows that $\mathcal{K} \models C(a)$. ■

We have then shown that the concept obtained by the procedure $\mathsf{k\text{-}msc}(a, \mathcal{K}, k)$ contains a as an instance, and has role depth at most k. It remains only to show that it is the least concept with these two properties.

Lemma 11 *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{EL}$-KB and a an individual in $\mathcal{A}$ and $k \in \mathbb{N}$. If $C = \mathsf{k\text{-}msc}(a, \mathcal{K}, k)$, then for all $\mathcal{EL}$-concept descriptions E with $rd(E) \leq k$ holds: $\mathcal{K} \models E(a)$ implies $C \sqsubseteq_{\mathcal{T}} E$.*

Proof. The proof is by induction on the role depth k. Assume first that $k = 0$. Then, any $\mathcal{EL}$-concept description E must be of the form $B_1 \sqcap \ldots \sqcap B_m$, where $m \geq 0$ and each B_i is a concept name. Likewise, C is the conjunction of all concept names that have a as an instance. Suppose that $C \not\sqsubseteq_{\mathcal{T}} E$. Then, there must exist a conjunct B_i from E that does not appear as a conjunct in C. But then, by the properties of the completion algorithm, it follows that a is not an instance of B_i; i.e. $\mathcal{K} \not\models B_i(a)$. But this implies that $\mathcal{K} \not\models E(a)$.

Let now $k > 0$. Then E is of the form

$$B_1 \sqcap \ldots B_m \sqcap \exists r_1.D_1 \sqcap \ldots \exists r_{m'}.D_{m'},$$

where B_i is a concept name and $rd(D_i) < k$ for all i. Suppose that $C \not\sqsubseteq_{\mathcal{T}} E$. Then, there must be a conjunct of E that has no conjunct of C as a subconcept. We assume w.l.o.g. that this conjunct is of the form $\exists r_i.D_i$ (otherwise, we can treat it as in the base case). Let b be an individual such that $r_i(a, b) \in \mathcal{K}$. Then, by induction hypothesis we know that $\mathcal{K} \not\models D_i(b)$. Additionally, for any concept name $A' \in S(a, r_i)$, D_i cannot subsume $C' :=$traversal-concept-c $(A', \mathsf{S}, k-1)$ (see Lemma 9). But then, this implies that there is a conjunct in D_i that does not subsume any of the conjuncts of C'. Thus, putting all these arguments together and using once again the properties of the completion algorithm, we can conclude that a is not an instance of $\exists r_i.D_i$, and hence $\mathcal{K} \not\models E(a)$. ■

The two lemmas yield the correctness of the overall procedure.

Theorem 12 *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{EL}$-KB and a an individual in $\mathcal{A}$ and $k \in \mathbb{N}$. Then $\mathsf{k\text{-}msc}(a, \mathcal{K}, k) \equiv k\text{-}msc_{\mathcal{K}}(a)$.*

Notice that the *k-msc* can grow exponential in the size of the knowledge base. However, using structure sharing, one can represent this concept using only polynomial space.

4.2 Most specific concept in Prob-$\mathcal{EL}_c^{01}$

In order to compute the msc in Prob-$\mathcal{EL}_c^{01}$, we follow the same idea used for $\mathcal{EL}$: we simply accumulate all concepts to which the individual a belongs, given the information in the completion sets. This process needs to be done recursively in order to account for both, the successors of a explicitly encoded in the ABox, and the nesting of existential restrictions masked by normalization names.

In the following we use the abbreviation $S^{>0}(a,r) := \bigcup_{v \in V \setminus \{0\}} S(a,r,v)$. We then define $\mathsf{traversal\text{-}concept\text{-}i}(a, \mathsf{S}, k)$ as

$$\begin{aligned}\bigsqcap_{B \in S(a,0)} B \sqcap \bigsqcap_{r \in \mathsf{RN}(\mathcal{K}'')} \Big(\bigsqcap_{r(a,b) \in \mathcal{K}''} & \exists r.\mathsf{traversal\text{-}concept\text{-}i}(b, \mathsf{S}, k-1) \sqcap \\ \bigsqcap_{B \in \mathsf{CN}(\mathcal{K}'') \cap S(a,r,0)} & \exists r.\mathsf{traversal\text{-}concept\text{-}c}(B, \mathsf{S}, k-1) \sqcap \\ \bigsqcap_{B \in \mathsf{CN}(\mathcal{K}'') \cap S(a,r,1)} & P_{=1}(\exists r.\mathsf{traversal\text{-}concept\text{-}c}(B, \mathsf{S}, k-1)) \sqcap \\ \bigsqcap_{B \in \mathsf{CN}(\mathcal{K}'') \cap S^{>0}(a,r)} & P_{>0}(\exists r.\mathsf{traversal\text{-}concept\text{-}c}(B, \mathsf{S}, k-1))\Big),\end{aligned}$$

where $\mathsf{traversal\text{-}concept\text{-}c}(B, \mathsf{S}, k+1)$ is

$$\begin{aligned}\bigsqcap_{C \in S_0(B,0)} B \sqcap \bigsqcap_{r \in \mathsf{RN}} \Big(\bigsqcap_{C \in S_0(B,r,0)} & \exists r.\mathsf{traversal\text{-}concept\text{-}c}(C, \mathsf{S}, k) \sqcap \\ \bigsqcap_{C \in S_0(B,r,1)} & P_{=1}(\exists r.\mathsf{traversal\text{-}concept\text{-}c}(C, \mathsf{S}, k)) \sqcap \\ \bigsqcap_{C \in S_0^{>0}(B,r)} & P_{>0}(\exists r.\mathsf{traversal\text{-}concept\text{-}c}(C, \mathsf{S}, k))\Big)\end{aligned}$$

and $\mathsf{traversal\text{-}concept\text{-}c}(B, \mathsf{S}, 0) = \bigsqcap_{C \in S_0(B,0)} B$.

Once the traversal concept has been computed, it is possible to remove all normalization names preserving the instance relation, which gives us the msc in the original signature of $\mathcal{K}$. The proof is analogous to the one for $\mathcal{EL}$, only treating the probabilistic constructors in a similar way as done for the existential restrictions.

Theorem 13 *Let $\mathcal{K}$ a Prob-$\mathcal{EL}_c^{01}$-knowledge base, $a \in \mathsf{IN}(\mathcal{A})$, and $k \in \mathbb{N}$; then* $\mathsf{Remove\text{-}normalization\text{-}names}(\mathsf{traversal\text{-}concept\text{-}i}(a, \mathsf{S}, k)) \equiv k\text{-}msc_{\mathcal{K}}(a)$.

5 Conclusions

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

Together with the completion-based computation of role-depth limited (least) common subsumers given in [10] these results complete the bottom-up approach for general $\mathcal{EL}$- and Prob-$\mathcal{EL}_c^{01}$-KBs. This approach yields a practical method to compute commonalities for differing observations regarding individuals. To the best of our knowledge this has not been investigated for DLs that can express uncertainty.

References

[1] F. Baader. Least common subsumers, most specific concepts, and role-value-maps in a description logic with existential restrictions and terminological cycles. LTCS-Report LTCS-02-07, Chair f. Automata Theory, Inst. for Theoretical Computer Science, TU Dresden, Germany, 2002. See `http://lat.inf.tu-dresden.de/research/reports.html`.

[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, 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.

[4] F. Baader and R. Küsters. Computing the least common subsumer and the most specific concept in the presence of cyclic $\mathcal{ALN}$-concept descriptions. In O. Herzog and A. Günter, editors, *Proc. of the 22th German Annual Conf. on Artificial Intelligence (KI'98)*, volume 1504 of *Lecture Notes in Computer Science*, pages 129–140, Bremen, Germany, 1998. Springer.

[5] 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.

[6] R. Küsters and R. Molitor. Approximating most specifc concepts in description logics with existential restrictions. In F. Baader, G. Brewka, and T. Eiter, editors, *Proc. of the 24th German Annual Conf. on Artificial Intelligence (KI'01)*, volume 2174 of *Lecture Notes In Artificial Intelligence*, pages 33–47. Springer, 2001.

[7] R. Küsters and R. Molitor. Approximating most specific concepts in description logics with existential restrictions. *AI Communications*, 15(1):47–59, 2002.

[8] 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.

[9] 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.

[10] R. Peñaloza and A.-Y. Turhan. Role-depth bounded least common subsumers by completion for $\mathcal{EL}$- and Prob-$\mathcal{EL}$-TBoxes. In V. Haarslev, D. Toman, and G. Weddell, editors, *Proc. of the 2010 Description Logic Workshop (DL'10)*, 2010.