ARTICLE IN PRESS IJA:7704

International Journal of Approximate Reasoning ••• (••••) •••–•••

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Completion-based generalization inferences for the Description Logic $\mathcal{ELOR}$ with subjective probabilities

Andreas Ecke [a,1], Rafael Peñaloza [a,b,2], Anni-Yasmin Turhan [a,*,3]

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

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Description Logics
Probabilistic Description Logics
Generalizations
Non-standard inferences
Least common subsumer
Most specific concept

ABSTRACT

Description Logics (DLs) are a well-established family of knowledge representation formalisms. One of its members, the DL $\mathcal{ELOR}$ has been successfully used for representing knowledge from the bio-medical sciences, and is the basis for the OWL 2 EL profile of the standard ontology language for the Semantic Web. Reasoning in this DL can be performed in polynomial time through a completion-based algorithm.

In this paper we study the logic Prob-$\mathcal{ELOR}$, that extends $\mathcal{ELOR}$ with subjective probabilities, and present a completion-based algorithm for polynomial time reasoning in a restricted version, Prob-$\mathcal{ELOR}^{01}_{c}$, of Prob-$\mathcal{ELOR}$. We extend this algorithm to computation algorithms for approximations of (i) the most specific concept, which generalizes a given individual into a concept description, and (ii) the least common subsumer, which generalizes several concept descriptions into one. Thus, we also obtain methods for these inferences for the OWL 2 EL profile. These two generalization inferences are fundamental for building ontologies automatically from examples. The feasibility of our approach is demonstrated empirically by our prototype system Gel.

1. Introduction

Broadly speaking Description Logics (DLs) are a family of logical formalisms that allow to characterize categories from an application domain by so-called concept descriptions. These concept descriptions are the main building blocks for DL knowledge bases. When it comes to building or maintaining large knowledge bases the task of generalizing a collection of concept descriptions into a single one is a central task. For most real-world applications it is not enough to represent only crisp knowledge, instead probabilistic knowledge needs to be represented as well. Recently, a probabilistic variant of DLs that is based on subjective probabilities was introduced and classical reasoning services have been investigated for it in [1]. The main contribution of this paper is to lift our approach to compute generalizations [2] to the case of DLs with probabilities.

* Corresponding author.
E-mail addresses: ecke@tcs.inf.tu-dresden.de (A. Ecke), penaloza@tcs.inf.tu-dresden.de (R. Peñaloza), turhan@tcs.inf.tu-dresden.de (A.-Y. Turhan).
1 Supported by DFG Graduiertenkolleg 1763 (QuantLA) (grant GRK 1763).
2 Partially supported by the DFG within the Cluster of Excellence 'cfAED' (grant SFB 912).
3 Partially supported by the German Research Foundation (DFG) in the Collaborative Research Center 912 "Highly Adaptive Energy-Efficient Computing" (grant cfaed).

http://dx.doi.org/10.1016/j.ijar.2014.03.001
0888-613X/© 2014 Elsevier Inc. All rights reserved.

Description Logics are a family of knowledge representation formalisms with unambiguous semantics. They can be used to represent a knowledge domain by formalizing its vocabulary as concept descriptions, which are built from concept and role names using the constructors provided by the chosen DL [3]. One well-known DL is $\mathcal{EL}$, which offers the constructors conjunction ($C \sqcap D$), existential restrictions ($\exists r.C$) and the top concept ($\top$) and knowledge is represented through a set of axioms. For instance, using the concept names Person, Female and Mother and a role name has-child, one can describe in $\mathcal{EL}$ that mothers are female persons having at least one child using the axiom:

$$\mathsf{Mother} \equiv \mathsf{Person} \sqcap \mathsf{Female} \sqcap \exists \mathsf{has\text{-}child}.\top.$$

All axioms that define the terminology of the domain are collected in the so-called *TBox*. Besides describing terminological knowledge, DLs also allow for the representation of instances of concepts, so-called individuals. With the individual names mary and peter, we can describe that Mary is a woman and Peter is her son using three assertions: Woman(mary), Male(peter) and has-child(mary, peter). Assertions are collected in an *ABox*. Together, TBox and ABox form a *knowledge base* (KB).

DL reasoner systems offer a variety of reasoning services, that allow to deduce implicit knowledge from the axioms and assertions stated in a KB. Commonly provided standard reasoning services include *concept subsumption*, which determines subconcept relationships of two given concepts, and *instance checking*, which determines whether a given individual is an instance of a given concept. Indeed, using the axioms and assertions from above and the additional fact that women are exactly female persons ($\mathsf{Woman} \equiv \mathsf{Person} \sqcap \mathsf{Female}$), one can infer that mary is an instance of the concept Mother and that Mother is a subconcept of Woman. The process of computing *all* subsumption relationships of named concepts of a TBox is called *classification*.

Besides $\mathcal{EL}$, there is a variety of other Description Logics [3] for which reasoning services have been investigated. While most of these are more expressive than $\mathcal{EL}$, the additional constructors offered by these DLs (such as disjunction, negation and universal quantification in the DL $\mathcal{ALC}$) cause the important inference problems to become intractable. In $\mathcal{EL}$ and its extension $\mathcal{EL}^{++}$ subsumption, instance checking and many other reasoning tasks can be decided in polynomial time [4]. This can be done by *completion algorithms*, which compute the canonical model for a given KB, from which all subsumption relationships between named concepts and instance relationship between individuals and named concepts can be directly read off. The computation of the canonical models via completion serves as a foundation on which our algorithms for computing generalizations are built. Despite their relatively low expressive power, the $\mathcal{EL}$-family of DLs are used to define concepts in a number of large-scale bio-medical ontologies, such as SNOMED CT [5,6] and the Gene Ontology [7].

$\mathcal{EL}^{++}$ is a maximal subset of the most commonly used DL-features, for which standard inference problems still have polynomial complexity, which is the main reason that it has been standardized as OWL 2 EL profile of the Web Ontology Language [8] by the W3C. This in turn led to an increased use of $\mathcal{EL}^{++}$ in practical applications. Leaving the concept constructor of data-types and the bottom concept $\bot$ in $\mathcal{EL}^{++}$ aside, the resulting DL is called $\mathcal{ELOR}$. $\mathcal{ELOR}$ offers role inclusions (indicated by $\mathcal{R}$ in the name), which allows among others to express role hierarchies and transitive roles. Another tractable extension offers *nominals* as a concept constructor (indicated by $\mathcal{O}$ in the name). Nominals are always interpreted as singleton sets. For example

$$\mathsf{SpanishFlu} \sqsubseteq \mathsf{Flu} \sqcap \exists \mathsf{origin}.\{\mathsf{france}\}$$

expresses that the Spanish flu was a flu that originated in France. In this case, the treatment of {france} as a nominal is preferred over a named concept France, since it is clearly an instance of a country and not a general concept with many instances, and thus captures the intention that there is only one country France. The completion-based approach to compute generalizations has recently been extended to $\mathcal{ELOR}$ in [9].

Classical DLs like those mentioned above only allow to define crisp and definite knowledge. However, many application domains require to model uncertain knowledge. In the previous example, one might want to express that the case fatality rate of Spanish flu was >2.5%, i.e., a person infected with Spanish flu would die with a probability of more than 0.025, given no other knowledge about this patient. In the last years several approaches have been devised to capture uncertain information in DL knowledge bases (see, for example, [10–12]). In this paper we consider extensions of the probabilistic DLs introduced by Lutz and Schröder [1]. This family of DLs allows the modeling of uncertain knowledge by introducing probabilistic constructors. Prob-$\mathcal{EL}$ uses subjective (or Type-2 [13]) probabilities, which correspond to degrees of belief and are interpreted using a multiple-world semantics. For example, in Prob-$\mathcal{EL}$ one can express that obese people are likely to have high pressure, without requiring every obese person to be hypertense, using the axiom

$$\mathsf{Obese} \sqsubseteq P_{\geqslant 0.9} \exists \mathsf{hasCondition}.\mathsf{HighPressure}.$$

While most DLs studied in [1] are intractable or even undecidable for unrestricted probabilistic roles, a fragment Prob-$\mathcal{EL}_c^{01}$ extending $\mathcal{EL}$ was identified to still admit polynomial time reasoning. In this fragment, probabilistic concepts can be constructed using only the probabilities >0 and =1. A completion algorithm for classifying TBoxes in the language Prob-$\mathcal{EL}_c^{01}$ was described in [1]. However, the algorithm described by the authors is not complete—the corrected version is given in this paper, since it is needed in our algorithms for computing generalizations.

Beyond the standard reasoning services, there also exist a number of non-standard inferences like the generalization of different entities from DL knowledge bases. The *least common subsumer (lcs)* inference introduced in [14] generalizes a set of concept descriptions into a single new concept description that subsumes all the input concepts and that is least w.r.t. subsumption. Intuitively, the lcs captures all commonalities of the input concept descriptions. A second inference, the *most specific concept (msc)* [15], generalizes an individual into the most precise concept description that describes this individual.

Given the previous axioms that describes obese persons and mothers, assume that we have the additional knowledge that Mary is obese: $\mathsf{Obese}(\mathsf{mary})$. Then the msc of mary is the concept

$$\mathsf{Obese} \sqcap P_{\geqslant 0.9}\exists \mathsf{hasCondition}.\mathsf{HighPressure} \sqcap \mathsf{Mother}$$
$$\sqcap \mathsf{Female} \sqcap \mathsf{Person} \sqcap \mathsf{Woman} \sqcap \exists \mathsf{has\text{-}child}.\mathsf{Male},$$

which is incidentally equivalent to simply $\mathsf{Obese} \sqcap \mathsf{Mother} \sqcap \exists \mathsf{has\text{-}child}.\mathsf{Male}$. The lcs of this concept and $\mathsf{Person} \sqcap P_{\geqslant 0.6}\exists \mathsf{has\text{-}condition}.\mathsf{RadiusFracture}$ (which might occur if an x-ray only shows a vague line) is $\mathsf{Person} \sqcap P_{\geqslant 0.6}\exists \mathsf{has\text{-}condition}.\top$.

These generalization inferences have a variety of applications. In the bottom-up construction of knowledge bases new concept descriptions can be generated in an example-driven way from a set of individuals that a user selects [15,16]. Each of the selected individuals is first generalized into a concept description by the msc and then all of these concept descriptions are generalized into a single one by the lcs. This approach enables users of DL knowledge bases with little KR expertise to augment their ontologies with new concepts. Another application of generalization inferences are concept similarity measures [17,18]. These measures assess the similarity of two concepts and are the core of many ontology matching algorithms. Furthermore, in ontology-based information retrieval the msc and lcs are used to relax search concepts, which encode the information to be searched [19–21]. For more application of these generalization inferences see [16,2].

Neither the lcs nor the msc need to exist in $\mathcal{EL}$, if computed w.r.t. general or cyclic TBoxes [22] or cyclic ABoxes [23]. The reason is that the cyclic structure cannot be captured by a finite $\mathcal{EL}$-concept description. In [24] an extension of $\mathcal{EL}$ with greatest fixpoints was introduced, where the generalization concepts always exist. Earlier in [25] it was shown that under greatest fixed point semantics the lcs does exist. However, for both approaches the resulting DL may not be as easy to comprehend for a DL system user. Thus, we pursue a different approach here. Computation algorithms for approximative solutions for the lcs were devised in [2] and for the msc in [26]. These methods simply compute a generalization concept up to a certain size k, which is interpreted as a bound on the role-depth, i.e., the maximal nestings of quantifiers.

One way to compute the approximative generalizations k-lcs and k-msc is to use the canonical model constructed by the completion algorithm for $\mathcal{EL}$. This approach has been studied intensively and extended to $\mathcal{ELR}$ and $\mathcal{EL}$ with inverse roles [2,27,28]. Furthermore, completion-based classification algorithms become more widely used, both from a practical point of view in terms of reasoner implementations [29–31] as well as on the theoretical side with the recent extensions of $\mathcal{EL}$ with nominals [32], subjective probabilities [1] or even Horn variants of expressive DLs [33].

In cases where the lcs or msc exists and a large enough bound k was given, the methods for computing the role-depth bounded lcs and the role-depth bounded msc yield the exact solutions. However, to obtain the *least* common subsumer and the *most* specific concept by these methods in practice, a decision procedure for the existence of the lcs or msc, respectively, and a method for computing a sufficient k are needed. These have recently been supplied for $\mathcal{EL}$ in [34] and for $\mathcal{EL}$ extended by complex role inclusions in [35].

Although being a rather pragmatic approach, the role-depth bounded lcs and the role-depth bounded msc may yield approximations that are sufficient for most practical applications named above. Other applications require the notion of role-depth bounded generalizations. For example, [21] solves the problem of instance queries for concepts relaxed by similarity measures by computing a so-called mimic of the query concept w.r.t. a candidate individual a, which can be found by considering subconcepts of the role-depth bounded msc of a. Curé et al. [36] describe an application that evaluates user traces by making use of the probabilistic DLs as defined by Lutz and Schröder [1]. Interestingly, the authors need to compute the msc (and afterwards the lcs) for $k = 1$ in their application. They give an ad-hoc procedure to compute these inferences. Now, since their method for the 1-msc does not take the TBox information into account, their algorithm is not correct. In this paper we devise algorithms for computing the role-depth bounded generalization for Prob-$\mathcal{EL}_c^{01}$ and some of its extensions and we prove their correctness. In detail, the contributions of this paper are the following:

Classification algorithms. We give a uniform description of the completion-based classification procedures for the DLs $\mathcal{ELOR}$ and Prob-$\mathcal{ELO}_c^{01}$, i.e., Prob-$\mathcal{EL}_c^{01}$ extended by nominals. We also amend an error in the completion algorithm for Prob-$\mathcal{EL}_c^{01}$ presented in [1]. We show correctness of the extension of the amended algorithm to handle nominals.

Computation algorithms for the role-depth bounded lcs. The completion algorithms for classification are the basis, on which we develop algorithms to compute the role-depth bounded lcs in $\mathcal{ELOR}$ and Prob-$\mathcal{ELO}_c^{01}$. We also show correctness of our methods.

Computation algorithms for the role-depth bounded msc. Since the msc in the presence of nominals is trivial ($msc(a) = \{a\}$), another target DL should be considered in order to yield an informative version of the msc. Thus we consider $\mathcal{EL}$ and later Prob-$\mathcal{EL}_c^{01}$ as the target DL for the msc. Based on the completion algorithms for classification in $\mathcal{ELOR}$ and Prob-$\mathcal{ELO}_c^{01}$, we develop algorithms to compute the role-depth bounded msc w.r.t. KBs written in $\mathcal{ELOR}$ and Prob-$\mathcal{ELO}_c^{01}$ and show correctness of these methods.

Table 1
Concept constructors and TBox axioms for $\mathcal{ELOR}$.

	Syntax	Semantics
Named concept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
Top concept	$\top$	$\Delta^{\mathcal{I}}$
Nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
Conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
Existential restriction	$\exists r.C$	$\{d \in \Delta^{\mathcal{I}} \mid \exists e.(d,e) \in r^{\mathcal{I}} \wedge e \in C^{\mathcal{I}}\}$
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
RIA	$r_1 \circ \cdots \circ r_n \sqsubseteq s$	$(r_1 \circ \cdots \circ r_n)^{\mathcal{I}} \subseteq s^{\mathcal{I}}$

Implementation and optimizations for the classical DLs. To show that the obtained generalization algorithms can be practically utilized, we implemented those for $\mathcal{ELOR}$ in our system GEL. We describe several optimizations for the generalization inferences and evaluate our system on some bio-medical ontologies.

The paper is structured as follows: after introducing the basic notions of DLs and generalizations in Section 2, we discuss the completion algorithm and introduce the role-depth bounded lcs and msc algorithms for the classical DL $\mathcal{ELOR}$ in Section 3. Using this work as a prerequisite, Section 4 introduces the DL Prob-$\mathcal{ELO}_c^{01}$ with subjective probabilities and gives a correct completion algorithm. This completion algorithm then serves as a basis for algorithms to compute the role-depth bounded lcs and msc w.r.t. KBs formulated in this DL. All of the proofs can be found in Appendices A–B. Section 5 gives an overview of some optimizations for the generalization algorithms for $\mathcal{ELOR}$ and Prob-$\mathcal{ELO}_c^{01}$ and presents our implementation of the classical case GEL, which is implemented on top of the standard reasoner JCEL [37]. This system is used in an evaluation to show the practicability of our algorithms and optimizations in the context of knowledge bases from practical applications. Since our computation algorithms for the generalization inferences require a standard reasoner to compute the completion of the TBox and there is neither a reasoner for the probabilistic variants nor knowledge bases using this probabilistic DL available, we need to resort to an evaluation for $\mathcal{ELOR}$; this provides an approximate indicative of the performance of the generalization algorithms for the probabilistic case. We conclude the paper with an outline of possible future work.

2. Preliminaries

In this section we introduce the basic notions of classical Description Logics that will later be generalized to handle subjective probabilities. We start by defining concept descriptions for members of the $\mathcal{EL}$-family. Let N_C, N_R and N_I be mutually disjoint sets. N_C contains *concept names*, N_R contains *role names* and N_I contains *individual names*. From these sets *concept descriptions* (or *concepts* for short) are constructed inductively as follows. Let $A \in N_C$ be a concept name, $r \in N_R$ be a role name, and $a \in N_I$ be an individual name. $\mathcal{ELO}$*-concept descriptions* are built using the syntax rule

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

$\mathcal{EL}$*-concept descriptions* are $\mathcal{ELO}$-concept descriptions that do not contain *nominals*; i.e., concepts of the form $\{a\}$.

The semantics of $\mathcal{ELO}$ is defined by means of interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a non-empty *domain* $\Delta^{\mathcal{I}}$ and an *interpretation function* $\cdot^{\mathcal{I}}$ that assigns binary relations on $\Delta^{\mathcal{I}}$ to role names, subsets of $\Delta^{\mathcal{I}}$ to concept descriptions and elements of $\Delta^{\mathcal{I}}$ to individual names. For a more detailed description of the semantics, see [3]. The concept constructors, along with their syntax and semantics are displayed in the upper part of Table 1.

Let $\mathcal{L}$ be a Description Logic, (e.g., the DL $\mathcal{EL}$). *General concept inclusion axioms* (GCIs) are expressions of the form $C \sqsubseteq D$, where C and D are $\mathcal{L}$-concept descriptions. *Role inclusion axioms* (RIAs) are statements of the form $r_1 \circ \cdots \circ r_n \sqsubseteq s$ for $1 \leqslant n$, where $\{r_1, \ldots, r_n, s\} \subseteq N_R$. The interpretation of a *role chain* $r_1 \circ \cdots \circ r_n$ is

$$(r_1 \circ \cdots \circ r_n)^{\mathcal{I}} = \{(d_0, d_n) \mid \exists d_1, \ldots, d_{n-1}. \forall 1 \leqslant i \leqslant n. (d_{i-1}, d_i) \in r_i^{\mathcal{I}}\}.$$

An interpretation $\mathcal{I}$ *satisfies* a GCI $C \sqsubseteq D$, denoted as $\mathcal{I} \models C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$; it *satisfies* a RIA $r_1 \circ \cdots \circ r_n \sqsubseteq s$, denoted as $\mathcal{I} \models r_1 \circ \cdots \circ r_n \sqsubseteq s$, if $(r_1 \circ \cdots \circ r_n)^{\mathcal{I}} \subseteq s^{\mathcal{I}}$. These axioms are summarized in the lower part of Table 1. For the rest of this paper, we use $C \equiv D$ as an abbreviation for the two GCIs $C \sqsubseteq D$ and $D \sqsubseteq C$. RIAs allow to express sub-role relationships of the form $r \sqsubseteq s$ and thus a set of RIAs can be used to define *role hierarchies*. Additionally, RIAs can also express transitivity of roles by stating $r \circ r \sqsubseteq r$.

A *TBox* $\mathcal{T}$ is a finite set of axioms. An $\mathcal{ELO}$*-TBox* is a finite set of GCIs built from $\mathcal{ELO}$-concept descriptions. In addition, an $\mathcal{ELOR}$*-TBox* may also contain finitely many RIAs. An interpretation $\mathcal{I}$ is a *model of a TBox* $\mathcal{T}$ if it satisfies all the axioms contained in the TBox $\mathcal{T}$.

Concept assertions are statements of the form $C(a)$, where C is a concept description and a is an individual name, while *role assertions* are statements of the form $r(a,b)$, where r is a role name, and a and b are individual names. We say that the interpretation $\mathcal{I}$ *satisfies* a concept assertion $C(a)$, denoted as $\mathcal{I} \models C(a)$, if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ and it satisfies a role assertion $r(a,b)$, denoted as $\mathcal{I} \models r(a,b)$, if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$. An *ABox* $\mathcal{A}$ is a set of concept or role assertions. An $\mathcal{L}$*-ABox* is a set of concept or role assertions, where only $\mathcal{L}$-concept descriptions are used in the concept assertions. An interpretation $\mathcal{I}$ is a *model of an ABox* $\mathcal{A}$ if it satisfies all concept and role assertions in $\mathcal{A}$.

A *knowledge base* (KB) $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ consists of a TBox $\mathcal{T}$ and an ABox $\mathcal{A}$. We call a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ an $\mathcal{L}$-knowledge base, if $\mathcal{T}$ is an $\mathcal{L}$-TBox and $\mathcal{A}$ an $\mathcal{L}$-ABox. An interpretation $\mathcal{I}$ is a *model of a knowledge base* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ if it is a model both, $\mathcal{T}$ and $\mathcal{A}$.

The formal semantics of the concept descriptions and the components of a knowledge base are used to define reasoning services. Since none of the DLs from Table 1 can express contradictory information, satisfiability (i.e., deciding whether a KB has a model) is trivial in these DLs. A concept description C is *subsumed* by D w.r.t. a TBox $\mathcal{T}$ (written $C \sqsubseteq_\mathcal{T} D$) iff for every model $\mathcal{I}$ of $\mathcal{T}$ it holds that $\mathcal{I} \models C \sqsubseteq D$. The concepts C and D are *equivalent* w.r.t. $\mathcal{T}$ (written $C \equiv_\mathcal{T} D$), if $C \sqsubseteq_\mathcal{T} D$ and $D \sqsubseteq_\mathcal{T} C$ hold. *Classification* of a TBox $\mathcal{T}$ is the computation of all subsumption relationships between concept names mentioned in $\mathcal{T}$.

An individual $a \in N_I$ is an *instance of* a concept C w.r.t. a KB $\mathcal{K}$ (denoted by $\mathcal{K} \models C(a)$) if $\mathcal{I} \models C(a)$ for all models $\mathcal{I}$ of $\mathcal{K}$. The *ABox realization problem* is to compute for each individual a in a given ABox $\mathcal{A}$ the set of those concept names from $\mathcal{K}$ that have a as an instance and that are least w.r.t. $\sqsubseteq_\mathcal{T}$.

Subsumption testing, instance checking, and even the more general problems of TBox classification and ABox realization can be done in polynomial time by a completion algorithm for $\mathcal{EL}$ [4] and the other DLs introduced above [38,32]. While the completion algorithm for extensions of $\mathcal{EL}$ by nominals introduced by Baader et al. [38] turned out to be incomplete, the method from Kazakov et al. [32] fixes this issue, yielding a sound and complete algorithm. We use this correct method as a basis for computing generalization inferences from a knowledge base, as described next.

When computing generalizations of either concept descriptions or individuals described in a KB, these concept descriptions and the knowledge base are written in a particular DL. On the other hand, the concept descriptions that capture the generalizations do not need to be written in the same DL; for instance, one may be interested in using less expressive constructors in the description of the generalizations. Thus, we distinguish between a *source DL* $\mathcal{L}_s$ for the input and a *target DL* $\mathcal{L}_t$ in which the generalization is formulated.

Definition 1 *(lcs, msc).* Let $\mathcal{L}_s$ and $\mathcal{L}_t$ be two DLs and $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{L}_s$-knowledge base. The *least common subsumer* of $\mathcal{L}_s$-concept descriptions $C_1, \ldots, C_n$ w.r.t. $\mathcal{T}$ (written: $lcs_\mathcal{T}(C_1, \ldots, C_n)$) is the $\mathcal{L}_t$-concept description D such that

1. $C_i \sqsubseteq_\mathcal{T} D$, for all $1 \leqslant i \leqslant n$ and
2. for each $\mathcal{L}_t$-concept description E holds: $C_i \sqsubseteq_\mathcal{T} E$ for all $1 \leqslant i \leqslant n$ implies $D \sqsubseteq_\mathcal{T} E$.

The *most specific concept* of an individual a from $\mathcal{K}$ (written: $(msc_\mathcal{K}(a))$) is the $\mathcal{L}_t$-concept description D such that

1. $\mathcal{K} \models D(a)$, and
2. for each $\mathcal{L}_t$-concept description E holds: $\mathcal{K} \models E(a)$ implies $D \sqsubseteq_\mathcal{T} E$.

If the target DL $\mathcal{L}_t$ is not clear from the context, we write $\mathcal{L}_t$-lcs or $\mathcal{L}_t$-msc throughout this paper. Typically, $\mathcal{L}_s = \mathcal{L}_t$ is considered; however, for DLs with disjunction the lcs is simply the disjunction of the input concept descriptions and thus not very informative. A more informative version can be obtained if the target language does not allow for disjunction [39,40]. For the $\mathcal{EL}$-family of DLs, the lcs and also the msc are, if they exist, unique up to equivalence (w.r.t. the underlying TBox or KB). Thus it is justified to speak of *the* lcs or *the* msc, respectively. Similarly, for target DLs $\mathcal{L}_t$ that offer nominals the msc is always trivial, since

$$msc(a) = \{a\}.$$

In order to obtain an informative msc for KBs written in a DL with nominals, we select a target DL that does not offer this kind of constructor.

In [22] it was shown that the $\mathcal{EL}$-lcs w.r.t. general $\mathcal{EL}$-TBoxes does not need to exist, when using the descriptive semantics, which is the standard semantics for DLs.[4] Likewise the msc in $\mathcal{EL}$ does not need to exist for cyclic ABoxes, as shown by Küsters and Molitor [26]. The reason for the non-existence is in both cases that cycles cannot be expressed by a finite $\mathcal{EL}$-concept description. In [24] $\mathcal{EL}$ was extended by fixed-points that can capture such cycles. Since we want to obtain a concept description for the lcs that is expressed in that DL in which the TBox is written (or a sublogic of it), we follow the idea from [2] and compute an approximative solution by limiting the maximal nesting of quantifiers in the generalizations. The *role depth* ($rd()$) of an $\mathcal{ELOR}$-concept description[5] is defined inductively as follows: Let $A \in N_C$ and $a \in N_I$, then

$$rd(A) := rd(\top) := rd\big(\{a\}\big) := 0,$$

[4] For greatest fixed point semantics, however, the lcs written in $\mathcal{EL}$ does always exist, see [22,25].

[5] Strictly speaking there are no $\mathcal{ELOR}$-concept descriptions. When referring to an $\mathcal{ELO}$-concept description that is defined w.r.t. an $\mathcal{ELOR}$-TBox we speak of an $\mathcal{ELOR}$-concept description in slight abuse of notation.

Please cite this article in press as: A. Ecke et al., Completion-based generalization inferences for the Description Logic $\mathcal{ELOR}$ with subjective probabilities, International Journal of Approximate Reasoning (2014), http://dx.doi.org/10.1016/j.ijar.2014.03.001

$$rd(C \sqcap D) := \max\{rd(C), rd(D)\},$$
$$rd(\exists r.C) := 1 + rd(C).$$

This leads to the following definition of a role-depth bounded lcs and a role-depth bounded msc, which is the most specific generalization up to the given role-depth bound.

Definition 2 *(Role-depth bounded lcs, role-depth bounded msc).* Let $\mathcal{L}_s, \mathcal{L}_t$ be DLs, $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{L}_s$-knowledge base and $k \in \mathbb{N}$. The *role-depth bounded least common subsumer* of $\mathcal{L}_s$-concept descriptions $C_1, \ldots, C_n$ w.r.t. $\mathcal{T}$ and k (written $k\text{-lcs}_{\mathcal{T}}(C_1, \ldots, C_n)$) is the $\mathcal{L}_t$-concept description D such that

1. $rd(D) \leqslant k$,
2. $C_i \sqsubseteq_{\mathcal{T}} D$, for all $1 \leqslant i \leqslant n$, and
3. for each $\mathcal{L}_t$-concept description E holds: $C_i \sqsubseteq_{\mathcal{T}} E$ for all $1 \leqslant i \leqslant n$ and $rd(E) \leqslant k$ imply $D \sqsubseteq_{\mathcal{T}} E$.

The *role-depth bounded most specific concept* of an individual a w.r.t. $\mathcal{K}$ and k (written $k\text{-msc}_{\mathcal{K}}(a)$) is the $\mathcal{L}_t$-concept description D such that

1. $rd(D) \leqslant k$,
2. $\mathcal{K} \models D(a)$, and
3. for each $\mathcal{L}_t$-concept description E holds: $\mathcal{K} \models E(a)$ and $rd(E) \leqslant k$ imply $D \sqsubseteq_{\mathcal{T}} E$.

Similarly to the lcs and msc, the k-lcs and k-msc are unique up to equivalence for a given k, thus we speak of *the* k-lcs and *the* k-msc. The reason for this uniqueness of the k-lcs is that there are only finitely many role-depth bounded, common subsumers of C and D modulo equivalence (and similarly, for the k-msc only finitely many role-depth bounded concepts with a as instance). Thus, the k-lcs and k-msc can always be written as the conjunction of all these subsumers. Again, we may write $\mathcal{L}_t$-k-lcs or $\mathcal{L}_t$-k-msc to specify the target DL explicitly.

If the exact lcs $L = \text{lcs}_{\mathcal{T}}(C, D)$ of two concepts C and D exists and has role-depth $rd(L) = k$, then the k-lcs of C and D will be equivalent to L, as they both subsume each other by Definitions 1 and 2. The same is true for the msc: If it exists, it will be found for a sufficiently high role-depth bound k. This implies the uniqueness also for the general lcs and msc. Also note that both the k-lcs and the k-msc can have exponential size in the role-depth bound k. This is easy to see for the TBox

$$\mathcal{T} = \{A \sqsubseteq \exists r.A \sqcap \exists s.A,\ B \sqsubseteq \exists r.B \sqcap \exists s.B\},$$

where the k-lcs of A and B takes the form of a full binary tree of depth k.

3. Completion-based Inferences in $\mathcal{ELOR}$

To understand how the completion-based inferences for DLs with subjective probabilities work, a basic understanding of these algorithms for the classic description logics is very helpful. In this section we present and discuss the algorithms to compute classifications and generalizations in the DL $\mathcal{ELOR}$. These methods are all based on the completion method, which allows to classify TBoxes written in $\mathcal{EL}$ and several of its extensions in polynomial time [4,38,32]. All the missing proofs can be found in Appendix A.

We start by briefly describing the completion algorithm for classifying classical $\mathcal{ELOR}$-TBoxes, which is based on the consequence-based algorithm for $\mathcal{ELO}$ recently presented in [32]. Following the approach from [2] and [27], this completion-based classification method is the foundation for computing the role-depth bounded lcs in $\mathcal{ELOR}$ [9].

3.1. Classification in $\mathcal{ELOR}$

Completion algorithms for TBox classification and ABox realization in $\mathcal{EL}$ and its extensions typically proceed in three phases:

1. Normalize the knowledge base or TBox,
2. initialize the so-called completion sets and saturate them by applying completion rules, and
3. read-off the subsumption or instance relationships form the saturated sets.

The saturated completion sets represent canonical models of the TBox or KB.

For the description logic $\mathcal{EL}$, different algorithms are needed for TBox classification and ABox realization. However, this distinction disappears as soon as the DL is extended by nominals. Recall that a nominal is a concept whose interpretation is a singleton set (see Table 1). In other words, nominals are concepts that represent specific individuals of the knowledge domain. We have previously divided knowledge bases in two parts: the TBox, that represents the conceptual knowledge of

NF1	$C \sqcap \hat{D} \sqsubseteq E$	$\longrightarrow$	$\hat{D} \sqsubseteq A,\ C \sqcap A \sqsubseteq E$
NF2	$\exists r.\hat{C} \sqsubseteq D$	$\longrightarrow$	$\hat{C} \sqsubseteq A,\ \exists r.A \sqsubseteq D$
NF3	$\hat{C} \sqsubseteq \hat{D}$	$\longrightarrow$	$\hat{C} \sqsubseteq A,\ A \sqsubseteq \hat{D}$
NF4	$B \sqsubseteq \exists r.\hat{C}$	$\longrightarrow$	$B \sqsubseteq \exists r.A,\ A \sqsubseteq \hat{C}$
NF5	$B \sqsubseteq C \sqcap D$	$\longrightarrow$	$B \sqsubseteq C,\ B \sqsubseteq D$
NF6	$r_1 \circ r_2 \circ r_3 \sqsubseteq s$	$\longrightarrow$	$r_1 \circ r_2 \sqsubseteq t,\ t \circ r_3 \sqsubseteq s$

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

Fig. 1. $\mathcal{ELOR}$ normalization rules (from Baader et al. [4]).

the domain, and the ABox that states information about some named individuals. Using nominals, it is possible to simulate ABox assertions using GCIs as described by the following proposition.

Proposition 3. *Given the knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$. Let the TBox $\mathcal{T}'$ be as follows:*

$$\mathcal{T}' = \mathcal{T} \cup \big\{\{a\} \sqsubseteq C \bigm| C(a) \in \mathcal{A}\big\} \cup \big\{\{a\} \sqsubseteq \exists r.\{b\} \bigm| r(a,b) \in \mathcal{A}\big\}.$$

Then $\mathcal{K}$ and $\mathcal{T}'$ are equivalent, i.e., the models of $\mathcal{K}$ are exactly the models of $\mathcal{T}'$.

Proof. Let $\mathcal{I}$ be an interpretation. Then $\mathcal{I}$ satisfies the concept assertion $C(a)$ iff $a^\mathcal{I} \in C^\mathcal{I}$ iff $\{a\}^\mathcal{I} \subseteq C^\mathcal{I}$ iff $\mathcal{I}$ satisfies the GCI $\{a\} \sqsubseteq C$. Similarly, $\mathcal{I}$ satisfies the assertion $r(a,b)$ iff $(a^\mathcal{I}, b^\mathcal{I}) \in r^\mathcal{I}$ iff $a^\mathcal{I} \in \{d \in \Delta^\mathcal{I} \mid (d, b^\mathcal{I}) \in r^\mathcal{I}\}$ iff $\{a\}^\mathcal{I} \subseteq \{d \in \Delta^\mathcal{I} \mid \exists e \in \Delta^\mathcal{I}.(d,e) \in r^\mathcal{I} \wedge e \in \{b\}^\mathcal{I}\}$ iff $\mathcal{I}$ satisfies the GCI $\{a\} \sqsubseteq \exists r.\{b\}$. Thus, any model of $\mathcal{K}$ must be a model of $\mathcal{T}'$ and vice versa. □

This proposition shows that it suffices to consider TBox classification to obtain results for the ABox reasoning services for $\mathcal{ELOR}$. Whenever one wants to know whether an individual a is an instance of a concept C for a knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, one can simply check if $\{a\} \sqsubseteq_{\mathcal{T}'} C$ follows from the TBox $\mathcal{T}'$ as given in Proposition 3. For the rest of this section we therefore restrict our attention to reasoning w.r.t. TBoxes only.

We first present a completion-based classification algorithm for $\mathcal{ELOR}$, and then show how to use the computed completion sets for generalization inferences in this logic. Kazakov et al. [32] gave a complete reasoning algorithm for nominals, building upon the algorithms developed in [4], as the latter turned out to be incomplete in the presence of nominals. The completion algorithm presented next adapts the ideas of this consequence based classifier.

The first phase of the classification algorithm transforms the TBox into normal form. This normal form is based on the following auxiliary sets. Given an $\mathcal{ELOR}$-TBox $\mathcal{T}$, we use:

- $\mathsf{Sig}(\mathcal{T})$ to denote the set of concept names, role names, and individual names occurring in $\mathcal{T}$, and
- $\mathsf{BC}_\mathcal{T}$ to denote the set of *basic concepts for* $\mathcal{T}$, which contains $\top$, all concept names $A \in \mathsf{Sig}(\mathcal{T}) \cap N_C$, and all nominals $\{a\}$ for $a \in \mathsf{Sig}(\mathcal{T}) \cap N_I$.

Definition 4 *($\mathcal{ELOR}$-normal form).* An $\mathcal{ELOR}$-TBox $\mathcal{T}$ is in *normal form*, if all GCIs in $\mathcal{T}$ are of the form

$$A \sqsubseteq B, \qquad A_1 \sqcap A_2 \sqsubseteq B, \qquad A \sqsubseteq \exists r.B, \quad \text{or} \quad \exists r.A \sqsubseteq B;$$

and all role inclusion axioms are of the form

$$s \sqsubseteq r, \quad \text{or} \quad s \circ t \sqsubseteq r,$$

where $A, A_1, A_2, B \in \mathsf{BC}_\mathcal{T}$ and $\{r, s, t\} \subseteq N_R$.

All $\mathcal{ELOR}$-TBoxes can be transformed into normal form by applying a set of normalization rules given in [4] and depicted in Fig. 1. The main idea is to introduce new concept names for complex subconcepts and new role names to denote role chains as pairs of roles. The normalized TBox $\mathcal{T}'$ of an $\mathcal{ELOR}$-TBox $\mathcal{T}$ is then a conservative extension of $\mathcal{T}$ w.r.t. subsumption, that is, for all concepts C, D containing only names from $\mathsf{Sig}(\mathcal{T})$, we have $C \sqsubseteq_\mathcal{T} D$ iff $C \sqsubseteq_{\mathcal{T}'} D$ [4,2].

Before we describe the completion algorithm in detail, we introduce the reachability relation $\leadsto_R$, which plays a fundamental role in the correct treatment of nominals [4,32].

Definition 5 *($\leadsto_R$).* Let $\mathcal{T}$ be an $\mathcal{ELOR}$-TBox in normal form, $G \in N_C \cup \{\top\}$, and $D \in \mathsf{BC}_\mathcal{T}$. $G \leadsto_R D$ iff there exist roles $r_1, \ldots, r_n \in N_R$ and basic concepts $A_0, \ldots, A_n, B_0, \ldots, B_n \in \mathsf{BC}_\mathcal{T}$, $n \geqslant 0$, such that $A_i \sqsubseteq_\mathcal{T} B_i$ for all $0 \leqslant i \leqslant n$, $B_{i-1} \sqsubseteq \exists r_i.A_i \in \mathcal{T}$ for all $1 \leqslant i \leqslant n$, A_0 is either G or a nominal, and $B_n = D$.

Please cite this article in press as: A. Ecke et al., Completion-based generalization inferences for the Description Logic $\mathcal{ELOR}$ with subjective probabilities, International Journal of Approximate Reasoning (2014), http://dx.doi.org/10.1016/j.ijar.2014.03.001

OR1	If $A_1 \in S^G(A)$, $A_1 \sqsubseteq B \in \mathcal{T}$ and $B \notin S^G(A)$, then $S^G(A) := S^G(A) \cup \{B\}$
OR2	If $A_1, A_2 \in S^G(A)$, $A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{T}$ and $B \notin S^G(A)$, then $S^G(A) := S^G(A) \cup \{B\}$
OR3	If $A_1 \in S^G(A)$, $A_1 \sqsubseteq \exists r.B \in \mathcal{T}$ and $B \notin S^G(A, r)$, then $S^G(A, r) := S^G(A, r) \cup \{B\}$
OR4	If $B \in S^G(A, r)$, $B_1 \in S^G(B)$, $\exists r.B_1 \sqsubseteq C \in \mathcal{T}$ and $C \notin S^G(A)$, then $S^G(A) := S^G(A) \cup \{C\}$
OR5	If $B \in S^G(A, r)$, $r \sqsubseteq s \in \mathcal{T}$ and $B \notin S^G(A, s)$, then $S^G(A, s) := S^G(A, s) \cup \{B\}$
OR6	If $B \in S^G(A, r_1)$, $C \in S^G(B, r_2)$, $r_1 \circ r_2 \sqsubseteq s \in \mathcal{T}$ and $C \notin S^G(A, s)$, then $S^G(A, s) := S^G(A, s) \cup \{C\}$
OR7	If $\{a\} \in S^G(A_1) \cap S^G(A_2)$, $G \rightsquigarrow_R A_2$, and $A_2 \notin S^G(A_1)$, then $S^G(A_1) := S^G(A_1) \cup \{A_2\}$

Fig. 2. Completion rules for $\mathcal{ELOR}$.

Informally, the concept name D is reachable from G if there is a chain of existential restrictions leading to D that starts either with G or with a nominal. Notice that if an interpretation $\mathcal{I}$ satisfying the axiom $A \sqsubseteq \exists r.B$ is such that $A^{\mathcal{I}} \neq \emptyset$, then there must be an element of $\Delta^{\mathcal{I}}$ that belongs to A, and hence must have an r-successor that belongs to the concept B. In particular, this implies that $B^{\mathcal{I}} \neq \emptyset$. Thus, the reachability relation $G \rightsquigarrow_R D$ intuitively states that, under the assumption that G is not empty, D cannot be empty either. This information will be used to identify concept names that must be interpreted as a given nominal, as described next.

The completion algorithm for $\mathcal{ELOR}$ keeps a set of completion sets of the form $S^G(A)$ and $S^G(A, r)$ for every $G \in (\mathsf{Sig}(\mathcal{T}) \cap N_C) \cup \{\top\}$, every basic concept A and every role name r. Intuitively, these sets are used to make implicit subsumption relationships explicit; for instance, $B \in S^A(A)$ expresses that A is subsumed by B in any model of the TBox, and $B \in S^A(A, r)$ expresses that A is subsumed by $\exists r.B$. However, notice that this subsumption would still hold if A was interpreted as the empty set. Nominals, on the other hand, are a special kind of concept that can never have an empty interpretation, since they are always interpreted as singleton sets. This also implies that no subsumer of a nominal may obtain an empty interpretation, as it must contain at least the nominal individual. Since the non-emptiness of concepts may influence the subsumption relations, we need to be able to express it in some way. Thus, e.g. the completion set $S^G(A)$ stores all the subsumers of A under the assumption that the interpretation of G is non-empty. We use $G : A \sqsubseteq B$ to denote the conditional subsumption $A \sqsubseteq B$, given that G is not empty.

The completion sets are initialized for every $G \in (\mathsf{Sig}(\mathcal{T}) \cap N_C) \cup \{\top\}$, every basic concept A, and every role name r as follows:

$$S^G(A) = \{A, \top\},$$
$$S^G(A, r) = \emptyset.$$

These completion sets are then extended using the completion rules depicted in Fig. 2 exhaustively. It can be shown that the algorithm terminates after polynomial time, and is sound and complete for classifying the TBox; that is, for deciding subsumptions between concept names appearing in $\mathcal{T}$ [32]. In particular, once the completion sets are saturated, i.e., no completion rule is applicable, the completion sets have the following properties.

Proposition 6. *Let $\mathcal{T}$ be an $\mathcal{ELOR}$-TBox in normal form, $A, B \in \mathsf{BC}_{\mathcal{T}}$ be two basic concepts, $r \in \mathsf{Sig}(\mathcal{T}) \cap N_R$, and $G = A$ or $G \rightsquigarrow_R A$ if $A \in N_C$, and $G \in N_C \cup \{\top\}$ otherwise. Then, the following properties hold:*

$$A \sqsubseteq_{\mathcal{T}} B \quad \text{iff} \quad B \in S^G(A), \text{ and}$$
$$A \sqsubseteq_{\mathcal{T}} \exists r.B \quad \text{iff} \quad \text{there exists } E \in \mathsf{BC}_{\mathcal{T}} \text{ with } E \in S^G(A, r) \text{ and } B \in S^G(E).$$

A consequence of this proposition is that if we want to decide whether the TBox $\mathcal{T}$ entails the subsumption $C \sqsubseteq D$, where $C \in \mathsf{Sig}(\mathcal{T}) \cap N_C$ and D is a basic concept appearing in $\mathcal{T}$, it suffices to test only whether $D \in S^C(C)$. Analogously, if C is either $\top$ or a nominal, is suffices to test whether $D \in S^{\top}(C)$. To reduce the overhead introduced by the use of nominals, i.e. the saturation of a separate set of completion sets for each concept name occurring in the TBox and $\top$, it is possible to implement a two-phase approach that first applies the rules **OR1** to **OR6**, propagating this information to all completion sets, and only afterwards the derivations that depend on the presence of nominals (starting from rule **OR7**) are computed. For details on the benefits of this strategy and how to implement it see [32].

We now show how to use these completion sets for computing generalization inferences in $\mathcal{ELOR}$. First we describe the computation of the role-depth bounded $\mathcal{ELOR}$-lcs. Afterwards, we describe the computation of the corresponding msc. As described before, in the presence of nominals, the computation of the most specific concept describing an individual is a trivial task and the result may not be informative. Hence, we describe its approximation in the less expressive target DL $\mathcal{EL}$.

Please cite this article in press as: A. Ecke et al., Completion-based generalization inferences for the Description Logic $\mathcal{ELOR}$ with subjective probabilities, International Journal of Approximate Reasoning (2014), http://dx.doi.org/10.1016/j.ijar.2014.03.001

Procedure k-lcs$(C, D, \mathcal{T}, k)$
Input: C, D: $\mathcal{ELOR}$-concept descriptions; $\mathcal{T}$: $\mathcal{ELOR}$-TBox; $k \in \mathbb{N}$
Output: role-depth bounded $\mathcal{ELOR}$-lcs of C, D w.r.t. $\mathcal{T}$ and k

$\mathcal{T}' :=$ normalize$(\mathcal{T} \cup \{A \equiv C, B \equiv D\})$
$\mathsf{S}_\mathcal{T} :=$ apply-completion-rules$(\mathcal{T}')$
return k-lcs-r$(A, B, \mathsf{S}_\mathcal{T}, k, \mathsf{Sig}(\mathcal{T}))$

Procedure k-lcs-r$(X, Y, \mathsf{S}_\mathcal{T}, k, \mathsf{Sig}(\mathcal{T}))$
Input: $X, Y \in \mathsf{BC}_\mathcal{T}$; $k \in \mathbb{N}$;
$\mathsf{S}_\mathcal{T}$: set of saturated completion sets; signature of original TBox $\mathcal{T}$
Output: role-depth bounded $\mathcal{ELOR}$-lcs of X, Y w.r.t. $\mathcal{T}$ and k

common-names $:= S^X(X) \cap S^Y(Y) \cap \mathsf{BC}_\mathcal{T}$
if $k = 0$ **then**
  return $\displaystyle\bigsqcap_{P \in \text{common-names}} P$
else
  return $\displaystyle\bigsqcap_{P \in \text{common-names}} P \sqcap \bigsqcap_{r \in \mathsf{Sig}(\mathcal{T}) \cap N_R} \; \bigsqcap_{E \in S^X(X,r),\, F \in S^Y(Y,r)} \exists r.\text{k-lcs-r}(E, F, \mathsf{S}_\mathcal{T}, k-1, \mathsf{Sig}(\mathcal{T}))$

Fig. 3. Computation algorithm for role-depth bounded $\mathcal{ELOR}$-lcs.

3.2. Computing the role-depth bounded $\mathcal{ELOR}$-lcs

In order to compute the role-depth bounded lcs of two $\mathcal{ELOR}$-concepts, we take advantage of the properties of the completion sets computed by the completion algorithm, as described by Proposition 6. Essentially, we first accumulate the direct subsumers, stored in the completion sets, and then recursively improve the approximation by adding role successors until the exact lcs is found or the role-depth bound is reached. In the presence of nominals, special care needs to be taken in choosing the right completion sets since the non-emptiness of some of the concepts may produce new subsumption relations, but not all of these sets are relevant.

An algorithm that computes the role-depth bounded $\mathcal{ELOR}$-lcs using the completion sets can be found in Fig. 3. In the first step, two new concept names A and B are introduced as abbreviations for the concepts C and D, and the TBox is normalized. The completion algorithm from Fig. 2 is then applied on the extended and normalized TBox to obtain all the completion sets.

In the recursive procedure k-lcs-r, we first obtain all the basic concepts that subsume both A and B by intersecting the sets $S^A(A)$ and $S^B(B)$. Clearly, the conjunction of all these basic concepts is still a subsumer for A and B w.r.t. $\mathcal{T}'$, and hence also for the two input concepts. Next, for every role name r and every basic concept C in $S^A(A, r)$, we know that $\exists r.C$ is a subsumer of A, and similarly for $D \in S^B(B, r)$. Thus, we can recursively compute the least common subsumer, for a role-depth of $k-1$, for all pairs (C, D) in $S^A(A, r) \times S^B(B, r)$.

The concept computed in this way may be highly redundant. For instance, consider the example TBox from Section 2 again, this time with role inclusions:

$$\mathcal{T} = \{A \sqsubseteq \exists r.A \sqcap \exists s.A,\ B \sqsubseteq \exists r.B \sqcap \exists s.B,\ r \sqsubseteq t,\ s \sqsubseteq u\}.$$

Now, a naive implementation of the above algorithm would return a concept description corresponding to the full 4-ary tree of depth k as the k-lcs of A and B w.r.t. $\mathcal{T}$, where the four edges of each node are labelled with r, s, t and u. This problem can be solved using the optimization techniques introduced in Section 5.

Proposition 7. *Let $\mathcal{T}$ be an $\mathcal{ELOR}$-TBox, C and D be $\mathcal{ELOR}$-concept description and k be a natural number. Then $L =$ k-lcs$(C, D, \mathcal{T}, k)$ is the $\mathcal{ELOR}$-lcs of C and D w.r.t. $\mathcal{T}$ and the role-depth bound k.*

3.3. Computing the role-depth bounded $\mathcal{EL}$-msc w.r.t. $\mathcal{ELOR}$-KBs

We now turn our attention to the other generalization inference: the computation of the most specific concept representing a given individual. Recall that, since $\mathcal{ELOR}$ allows the use of nominals, computing the (exact) $\mathcal{ELOR}$ msc for a given individual is a trivial task: the most specific $\mathcal{ELOR}$-concept describing an individual $a \in N_I$ is always the nominal $\{a\}$. However, it may be of interest to compute the msc w.r.t. a less expressive target language that does not allow for nominals. Therefore, we now describe how to compute the role-depth bounded $\mathcal{EL}$-msc of an individual w.r.t. an $\mathcal{ELOR}$-KB.

An algorithm for computing the $\mathcal{EL}$-k-msc w.r.t. an $\mathcal{ELOR}$-KB is described in Fig. 4. Once again, its correctness is a consequence of the invariants described by Proposition 6. The set $S^\top(\{a\})$ contains all the basic concepts that subsume the nominal $\{a\}$; that is, all concepts whose interpretation must contain the individual $a^\mathcal{I}$. Likewise, $S^\top(\{a\}, r)$ contains all

Please cite this article in press as: A. Ecke et al., Completion-based generalization inferences for the Description Logic $\mathcal{ELOR}$ with subjective probabilities, International Journal of Approximate Reasoning (2014), http://dx.doi.org/10.1016/j.ijar.2014.03.001

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

$\mathcal{T} :=$ absorb-ABox$(\mathcal{K})$
$\mathcal{T}' :=$ normalize$(\mathcal{T})$
$\mathsf{S}_\mathcal{K} :=$ apply-completion-rules$(\mathcal{T}')$
return traversal-concept$(\{a\}, \mathsf{S}_\mathcal{K}, k, \mathsf{Sig}(\mathcal{K}))$

Procedure traversal-concept$(A, \mathsf{S}_\mathcal{K}, k, \mathsf{Sig}(\mathcal{K}))$
Input: A: basic concept from $\mathcal{T}'$; $\mathsf{S}_\mathcal{K}$: set of completion sets; $k \in \mathbb{N}$; signature of original KB $\mathcal{K}$
Output: role-depth bounded traversal concept of A w.r.t. $\mathcal{K}$ and k.

if $k = 0$ **then**
 return $\bigsqcap_{B \in S^A(A) \cap (\mathsf{Sig}(\mathcal{K}) \cap N_C \cup \{\top\})} B \sqcap \top$
else
 return $\bigsqcap_{B \in S^A(A) \cap (\mathsf{Sig}(\mathcal{K}) \cap N_C \cup \{\top\})} B \sqcap \bigsqcap_{r \in \mathsf{Sig}(\mathcal{K}) \cap N_R} \bigsqcap_{B \in S^A(A,r)} \exists r.\text{traversal-concept}(B, \mathsf{S}_\mathcal{K}, k-1, \mathsf{Sig}(\mathcal{K}))$

Fig. 4. Computation algorithm for the role-depth bounded $\mathcal{EL}$-msc w.r.t. $\mathcal{ELOR}$-KBs.

the existential restrictions subsuming $\{a\}$. Thus, a recursive conjunction of all these subsumers provides the most specific representation of the individual a.

Since the target language is $\mathcal{EL}$, all nominals are removed from the output. However, the recursion includes also the $\mathcal{EL}$-msc of the removed nominals, hence indeed providing the most specific $\mathcal{EL}$ representation of the input individual.

Proposition 8. *Let $\mathcal{K}$ be an $\mathcal{ELOR}$-KB, a be an individual occurring in $\mathcal{K}$ and k be a natural number. Then $M =$ k-msc$(a, \mathcal{K}, k)$ is the $\mathcal{ELR}$-msc of a w.r.t. $\mathcal{K}$ and the role-depth bound k.*

In this section we have shown how to compute generalization inferences with a bounded role-depth w.r.t. KBs written in the DL $\mathcal{ELOR}$. With the exception of datatypes and the bottom concept, this covers all major features of the OWL 2 EL profile of the standard ontology language OWL 2 [8]. Given its status as W3C standard, it is likely that more and bigger ontologies are built using this profile, thus the generalization inferences investigated in this paper and their computation algorithms for approximation will become more useful to ontology engineers. In fact, there already exist ontologies that use nominals in their representation. For example, the FMA ontology currently contains 85 nominals.

While many large ontologies have been built using the expressivity of languages from the $\mathcal{EL}$-family, and several tools have been developed for reasoning in them, the study of these logics and their inferences is usually restricted to the classical setting, where knowledge is certain and assertable. Unfortunately, in many domains it is unavoidable to deal with uncertain knowledge. For that reason, we are interested in studying the probabilistic variants of these logics and their inferences. In the following section we take a look at the probabilistic DL Prob-$\mathcal{ELOR}$.

4. Completion-based inferences in Prob-$\mathcal{ELOR}$

So far, we have focused our attention on representing and reasoning with knowledge that is certain. For example, the concept Father speaks of all those individuals that are *known* to be fathers. Likewise, we can use the assertion Father(bob) and the GCI Father $\sqsubseteq$ Parent to express the fact that Bob is a father, and every father is also a parent, respectively. However, when trying to represent knowledge it is not uncommon to encounter situations where a degree of uncertainty is unavoidable. This is often the case in the medical and biological domains, where knowledge is obtained through clinical testing, and there might exist hidden, or not completely understood, factors affecting the outcome. For instance, we would like to be able to express that obese people are *likely* to have high pressure, without asserting that every obese person *must* have high pressure.

A common method for dealing with uncertainty is through the use of probabilities, which associates every event to a degree, or weight, measuring the likelihood that the event will take place. Thus, a natural approach to represent uncertain knowledge is to try to extend description logics to handle probabilistic semantics. Several approaches for probabilistic extensions of description logics have been proposed over the years. We describe here one that deals with subjective, as opposed to statistical, probabilities.

The probabilistic logic Prob-$\mathcal{EL}$ was introduced by Lutz and Schröder [1] as an extension of $\mathcal{EL}$ that allows for probabilistic concepts and roles. Here, we extend these ideas to cover also the additional constructors found in $\mathcal{ELOR}$. Formally, Prob-$\mathcal{ELOR}$-concepts are built through the following syntactic rule

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

Please cite this article in press as: A. Ecke et al., Completion-based generalization inferences for the Description Logic $\mathcal{ELOR}$ with subjective probabilities, International Journal of Approximate Reasoning (2014), http://dx.doi.org/10.1016/j.ijar.2014.03.001

where $A \in N_C$, $r \in N_R$, $o \in N_I$, $\triangleright \in \{>, <, \geqslant, \leqslant, =\}$, and $q \in [0, 1]$. Intuitively, a concept of the form $P_{\triangleright q}C$ denotes the class of all objects that belong to C with a probability $\triangleright q$. For example, we can use the concept $P_{\geqslant 0.9}\exists\mathsf{hasCondition.HighPressure}$ to represent the class of all individuals that are likely to have high pressure.

The semantics of this logic generalizes the semantics of classical $\mathcal{ELOR}$ by considering a set of possible worlds, corresponding to a formalization of subjective (or Type 2) probabilities [13]. Formally, the semantics of Prob-$\mathcal{ELOR}$ are based on *probabilistic interpretations*.

Definition 9 *(Probabilistic interpretation).* A *probabilistic interpretation* is a tuple of the form $\mathcal{I} = (\Delta^{\mathcal{I}}, W, (\mathcal{I}_w)_{w \in W}, \mu)$, where $\Delta^{\mathcal{I}}$ is a (non-empty) domain, W is a non-empty set of (possible) *worlds*, μ is a discrete probability distribution over W, and for every $w \in W$, $\mathcal{I}_w$ is a classical $\mathcal{ELOR}$ interpretation with domain $\Delta^{\mathcal{I}}$. Additionally, for every $a \in N_I$ and every two worlds $w, w' \in W$, it holds that $a^{\mathcal{I}_w} = a^{\mathcal{I}_{w'}}$.

From a probabilistic interpretation, we can compute the probability that a given element of the domain $d \in \Delta^{\mathcal{I}}$ belongs to the interpretation of a concept name A, and respectively, the probability that a pair of individuals is related via a role r as follows:

$$p_d^{\mathcal{I}}(A) := \mu\big(\big\{w \in W \bigm| d \in A^{\mathcal{I}_w}\big\}\big),$$
$$p_{d,e}^{\mathcal{I}}(r) := \mu\big(\big\{w \in W \bigm| (d, e) \in r^{\mathcal{I}_w}\big\}\big).$$

The functions $\mathcal{I}_w$ and $p_d^{\mathcal{I}}$ are extended to complex concept descriptions through the following mutual recursion.

$$\begin{aligned}
\top^{\mathcal{I}_w} &= \Delta^{\mathcal{I}},\\
(C \sqcap D)^{\mathcal{I}_w} &= C^{\mathcal{I}_w} \cap D^{\mathcal{I}_w},\\
(\exists r.C)^{\mathcal{I}_w} &= \big\{d \in \Delta^{\mathcal{I}} \bigm| \exists e \in C^{\mathcal{I}_w}.(d, e) \in r^{\mathcal{I}_w}\big\},\\
\big(\{o\}\big)^{\mathcal{I}_w} &= \big\{o^{\mathcal{I}_w}\big\},\\
(P_{\triangleright q}C)^{\mathcal{I}_w} &= \big\{d \in \Delta^{\mathcal{I}} \bigm| p_d^{\mathcal{I}}(C) \triangleright q\big\},\\
(\exists P_{\triangleright q}r.C)^{\mathcal{I}_w} &= \big\{d \in \Delta^{\mathcal{I}} \bigm| \exists e \in C^{\mathcal{I}_w}.p_{d,e}^{\mathcal{I}}(r) \triangleright q\big\},\\
p_d^{\mathcal{I}}(C) &= \mu\big(\big\{w \in W \bigm| d \in C^{\mathcal{I}_w}\big\}\big).
\end{aligned}$$

It should be noted that the semantics of the probabilistic concepts $P_{\triangleright q}C$ does not depend on any specific world. Indeed, $(P_{\triangleright q}C)^{\mathcal{I}_w} = (P_{\triangleright q}C)^{\mathcal{I}_{w'}}$ holds for every $w, w' \in W$. The intuition behind this fact is that the probabilistic constructor accumulates the information across all possible worlds.

A Prob-$\mathcal{ELOR}$ TBox is a finite set of GCIs of the form $C \sqsubseteq D$, where C, D are Prob-$\mathcal{ELOR}$-concepts, and role inclusions of the form $r_1 \circ \cdots \circ r_n \sqsubseteq s$, where $r_1, \ldots, r_n, s \in N_R$. We say that the probabilistic interpretation $\mathcal{I}$ *satisfies* the GCI $C \sqsubseteq D$ if for every world $w \in W$ it holds that $C^{\mathcal{I}_w} \subseteq D^{\mathcal{I}_w}$. $\mathcal{I}$ satisfies the role inclusion $r_1 \circ \cdots \circ r_n \sqsubseteq s$ if $(r_1 \circ \cdots \circ r_n)^{\mathcal{I}_w} \subseteq s^{\mathcal{I}_w}$ holds for every $w \in W$. $\mathcal{I}$ is a *model* of the TBox $\mathcal{T}$ if $\mathcal{I}$ satisfies all the GCIs and all the role inclusions in $\mathcal{T}$.

Note that GCIs in Prob-$\mathcal{ELOR}$ are always crisp: In any model the elements must satisfy all GCIs in all possible worlds, but elements may be instances of the left-hand and right-hand sides of the GCI with a certain probability. This is different to the usual modelling of statistical probabilities (e.g. in [12]), where the given GCI must be satisfied to a certain degree over the whole domain, but any element of the domain either satisfies the GCI or not. One can simulate a probabilistic GCI of the form $C \sqsubseteq_{\triangleright p} D$ in Prob-$\mathcal{ELOR}$ using $C \sqsubseteq P_{\triangleright p}D$. In this case, any instance of C is always also an instance of $P_{\triangleright p}D$, which again is an instance of D in worlds with combined probability $\triangleright p$.

A Prob-$\mathcal{ELOR}$ ABox is a finite set of assertions of the form $C(a)$, $r(a, b)$, and $P_{\triangleright q}r(a, b)$, where C is a Prob-$\mathcal{ELOR}$-concept, r is a role name and a, b are individual names. We say that the probabilistic interpretation $\mathcal{I}$ *satisfies* the assertion $C(a)$, denoted as $\mathcal{I} \models C(a)$, if for every world $w \in W$ it holds that $a^{\mathcal{I}_w} \in C^{\mathcal{I}_w}$; similarly, we have $\mathcal{I} \models r(a, b)$ if $(a^{\mathcal{I}_w}, b^{\mathcal{I}_w}) \in r^{\mathcal{I}_w}$ for all $w \in W$, and $\mathcal{I} \models P_{\triangleright q}r(a, b)$ if $p_{a^{\mathcal{I}}, b^{\mathcal{I}}}^{\mathcal{I}}(r) \triangleright q$. $\mathcal{I}$ is a *model* of the ABox $\mathcal{A}$ if $\mathcal{I}$ satisfies all the assertions in $\mathcal{A}$.

Returning to our example, we can express that obese people are likely to have high pressure by including the GCI

$$\mathsf{Obese} \sqsubseteq P_{\geqslant 0.9}\exists\mathsf{hasCondition.HighPressure}$$

into the TBox.

As for classical $\mathcal{ELOR}$, an important decision problem in Prob-$\mathcal{ELOR}$ is the subsumption between concepts w.r.t. a given TBox. In this paper, we are mainly interested in studying the cases where this problem can be solved in polynomial time. Unfortunately, the probabilistic constructors increase the complexity of reasoning, and deciding subsumption becomes intractable in general. In fact, as shown in [41], the problem is ExpTime-complete, even if only one constructor of the form $P_{\triangleright q}$ with $q \in (0, 1)$ is allowed. Moreover, the problem becomes PSpace-hard when probabilistic existential restrictions of the form $\exists P_{>0}r$ or $\exists P_{=1}r$ are used.

To regain tractability, we restrict to the logic Prob-$\mathcal{ELOR}_c^{01}$, in which probabilistic concepts can only be of the form $P_{>0}C$ or $P_{=1}C$. This is without doubt a fairly inexpressive logic. However, it strictly increases the expressivity of $\mathcal{ELOR}$. For example, in this logic it is possible to express that obese people are *almost certainly*, but not necessarily, cardiovascular patients using the axiom $\mathsf{Obese} \sqsubseteq P_{=1}\mathsf{CardiovascularPatient}$. This extended expressivity provides the possibility of reasoning under a limited notion of uncertainty in Description Logics.

Intuitively, the concept $P_{>0}C$ expresses the class of individuals that could *possibly* belong to C, while $P_{=1}C$ contains the individuals that *almost certainly* belong to C. It is important to notice that the constructor $P_{=1}$ does add expressivity to the logic. Indeed, for an axiom $C \sqsubseteq D$ to be satisfied, an interpretation must interpret C as a subset of D in every world, independent to their likelihood provided by the probability distribution μ. On the other hand, a model $\mathcal{I}$ of $C \sqsubseteq P_{=1}D$ may have worlds w where $C^{\mathcal{I}_w} \not\subseteq D^{\mathcal{I}_w}$, as long as these worlds are almost impossible, i.e. have an associated probability $\mu(w) = 0$.

Using probabilistic concepts of the form $P_{>0}D$ allows to explicitly reason over possible implications. If the TBox contains GCIs that expresses that certain gene defects can possibly lead to obesity, and obesity can possible cause ischemic heart diseases, then one consequence is that these gene defects can possibly cause ischemic heart diseases. This reasoning would not be possible or meaningful in crisp $\mathcal{ELOR}$.

The concept constructor $P_{=1}C$ however seems less useful in practice. By the exact semantics, one could for example express that primes are odd with a probability of 1 ($\mathsf{Prime} \sqsubseteq P_{=1}\mathsf{Odd}$). However, we argue that in practice one can use this constructor to express almost certain or high probabilities in general, even though it may not conform to the exact semantics. For example, one can read the GCI $\mathsf{Influenza} \sqsubseteq P_{=1}\exists\mathsf{causes}.\mathsf{HighFever}$ as follows: influenza causes high fever most of the time, though there are exceptions to this. But influenza does *not* logically imply a fever. Then, Prob-$\mathcal{ELOR}_c^{01}$ allows practical reasoning about both possible and almost certain subsumers (and of course logical implications), which can be quite helpful in domains that need such a distinction.

It was previously shown that for the sublogic Prob-$\mathcal{EL}_c^{01}$ of Prob-$\mathcal{ELOR}_c^{01}$ that does neither allow nominals nor role inclusion axioms, subsumption can be decided in polynomial time using a completion-based algorithm [1]. We extend this result by showing that the polynomial upper bound still holds in the presence of nominals and role inclusions. To this aim, we extend the completion algorithm following the ideas presented by Kazakov et al. [32]. The completion algorithm we obtain this way will then be used to compute the role-depth bounded lcs and msc, in a way akin to the method presented in the previous section.

Notice that there is no probabilistic constructor applied to roles. Thus, we do not need to modify any of the rules that deal with role inclusion axioms in the algorithm, as role inclusions and subjective probabilities are completely orthogonal. For this reason, we restrict the description of our algorithm and its applications to generalization inferences to the sublogic Prob-$\mathcal{ELO}_c^{01}$ of Prob-$\mathcal{ELOR}_c^{01}$. This restriction should avoid unnecessary overhead of notation and rules, while increasing readability. The missing rules to deal with role inclusion axioms are shown in Appendix B; they are straightforward adaptations of the rules **OR5** and **OR6** to the probabilistic setting.

The completion rules for Prob-$\mathcal{ELO}_c^{01}$ are based on the completion algorithm for Prob-$\mathcal{EL}_c^{01}$ presented in [1]. The basic idea of this completion algorithm is the same as in the classical case: to construct a canonical model of the given knowledge base. However, since probabilistic interpretations contain a set of worlds, the completion algorithm has to work on sets of completion sets: one for each probabilistic concept. Basically, the completion algorithm uses a world with positive probability for each probabilistic concept $P_{>0}A$ occurring the TBox $\mathcal{T}$; this way, for each GCI $B \sqsubseteq P_{>0}A \in \mathcal{T}$ the world $v = P_{>0}A$ serves as witness for this subsumption.

Before introducing the completion algorithm for Prob-$\mathcal{ELO}_c^{01}$, we show that, similar to the non-probabilistic case, any knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ can be reduced to an equivalent TBox $\mathcal{T}'$ by introducing new concept inclusions for all assertions in $\mathcal{A}$. For concept assertions and classical role assertions, this can be done the same way as for classical $\mathcal{ELOR}$. For probabilistic role assertions $P_{>0}r(a, b)$ (and $P_{=1}r(a, b)$), we cannot use the translation to $\{a\} \sqsubseteq \exists P_{>0}r.\{b\}$, since Prob-$\mathcal{ELO}_c^{01}$ does not allow probabilistic roles in the TBox; however, we can move the probability constructor from the role to the front of the existential restriction to get $\{a\} \sqsubseteq P_{>0}\exists r.\{b\}$ (and analogously for $P_{=1}r(a, b)$). While this is not possible in general, since probabilistic roles have a different semantics that probabilistic existential restrictions, it works for our case, as individuals, and hence nominals, have the same interpretation in each world.

Lemma 10. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base. Then $\mathcal{K}$ can be reduced to a single, equivalent TBox $\mathcal{T}'$:*

$$\begin{aligned}\mathcal{T}' = \mathcal{T} \cup {} & \big\{\{a\} \sqsubseteq C \bigm| C(a) \in \mathcal{A}\big\} \cup \\ & \big\{\{a\} \sqsubseteq \exists r.\{b\} \bigm| r(a, b) \in \mathcal{A}\big\} \cup \\ & \big\{\{a\} \sqsubseteq P_{>0}\exists r.\{b\} \bigm| P_{>0}r(a, b) \in \mathcal{A}\big\} \cup \\ & \big\{\{a\} \sqsubseteq P_{=1}\exists r.\{b\} \bigm| P_{=1}r(a, b) \in \mathcal{A}\big\}\end{aligned}$$

In particular, instance checks $\mathcal{K} \models C(a)$ can be reduced to subsumption checks $\mathcal{T}' \models \{a\} \sqsubseteq C$.

Proof. We show that $\mathcal{K}$ and $\mathcal{T}'$ have the same models:

- $\mathcal{I}, w \models C(a) \in \mathcal{A}$ iff $a^{\mathcal{I}} \in C^{\mathcal{I},w}$ iff $\{a\}^{\mathcal{I}} \subseteq C^{\mathcal{I},w}$ iff $\mathcal{I}, w \models \{a\} \sqsubseteq C$,
- $\mathcal{I}, w \models r(a,b) \in \mathcal{A}$ iff $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I},w}$ iff $a^{\mathcal{I}} \in \{d \mid (d, b^{\mathcal{I}}) \in r^{\mathcal{I},w}\}$ iff $\mathcal{I}, w \models \{a\} \sqsubseteq \exists r.\{b\}$,
- $\mathcal{I}, w \models P_{>0}r(a,b) \in \mathcal{A}$ iff $\exists v \in W\colon\ \mu(v) > 0 \wedge (a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I},v}$ iff $a^{\mathcal{I}} \in \{d \mid \exists v \in W\colon\ \mu(v) > 0 \wedge (d, b^{\mathcal{I}}) \in r^{\mathcal{I},v}\}$ iff $\mathcal{I}, w \models \{a\} \sqsubseteq P_{>0}\exists r.\{b\}$,
- $\mathcal{I}, w \models P_{=1}r(a,b) \in \mathcal{A}$ iff $\forall v \in W\colon\ \mu(v) > 0 \Rightarrow (a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I},v}$ iff $a^{\mathcal{I}} \in \{d \mid \forall v \in W\colon\ \mu(v) > 0 \Rightarrow (d, b^{\mathcal{I}}) \in r^{\mathcal{I},v}\}$ iff $\mathcal{I}, w \models \{a\} \sqsubseteq P_{=1}\exists r.\{b\}$. □

In the following, we will therefore only consider Prob-$\mathcal{ELO}_c^{01}$-TBoxes instead of Prob-$\mathcal{ELO}_c^{01}$-knowledge bases.

4.1. Classification algorithm for Prob-$\mathcal{ELO}_c^{01}$

As in the non-probabilistic case, the completion algorithm for classifying a Prob-$\mathcal{ELO}_c^{01}$-TBox $\mathcal{T}$ works in three steps:

1. normalize the TBox $\mathcal{T}$,
2. construct the completion sets from the concept names in the normalized TBox, and
3. saturate the completion sets by exhaustively applying completion rules.

The normal form for Prob-$\mathcal{ELO}_c^{01}$-TBoxes is the same as for $\mathcal{ELOR}$ (excluding role inclusion axioms), i.e., a Prob-$\mathcal{ELO}_c^{01}$-TBox $\mathcal{T}$ is in *normal form*, if it contains only axioms of the form

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

where $C, C_1, C_2, D \in \mathsf{BC}_{\mathcal{T}}$ and $A \in \mathsf{Sig}(\mathcal{T}) \cap N_C$. However, for Prob-$\mathcal{ELO}_c^{01}$ the set of basic concepts is different: probabilistic concepts are considered basic concepts as well. The set $\mathsf{BC}_{\mathcal{T}}$ of Prob-$\mathcal{ELO}_c^{01}$ *basic concepts* contains $\top$, all nominals $\{a\} \in \mathcal{T}$, all concept names $\mathsf{Sig}(\mathcal{T}) \cap N_C$, and for each concept name A in $\mathsf{Sig}(\mathcal{T}) \cap N_C$ the concepts $P_{>0}A$ and $P_{=1}A$.

To transform a Prob-$\mathcal{ELO}_c^{01}$-TBox $\mathcal{T}$ into normal form, the same rules can be used as in the non-probabilistic case. These rules are shown in Fig. 1. Clearly, since Prob-$\mathcal{ELO}_c^{01}$ does not contain role inclusion axioms, rule **NF6** will not be applicable.

The set of worlds on which the completion algorithm works, is defined as $V := \{0, 1, \varepsilon\} \cup \mathcal{P}_0^{\mathcal{T}}$, where $\mathcal{P}_0^{\mathcal{T}}$ is the set of concepts of the form $P_{>0}A$ that occur in $\mathcal{T}$. The probability distribution μ for V assigns probability 0 to the world 0 and uniform probability $\frac{1}{|V \setminus \{0\}|}$ to all other worlds. In the following, we will use P_v as an abbreviation for $P_0X = X$, $P_1X = P_{=1}X$ and $P_vX = P_{>0}X$ for all $v \in V \setminus \{0, 1\}$.

To classify Prob-$\mathcal{ELO}_c^{01}$-TBoxes, we essentially combine the completion algorithm for $\mathcal{ELOR}$ (excluding the rules for role inclusions) with the completion algorithm for Prob-$\mathcal{EL}_c^{01}$ given in [1]. On the one hand the presence of nominals causes conditional subsumption relationships, i.e., the non-emptiness of a concept can yield subsumptions that would not be true if the concept were empty. On the other hand Prob-$\mathcal{EL}_c^{01}$ introduces a set of different worlds on which we have to infer subsumptions. To account for both of this in Prob-$\mathcal{ELO}_c^{01}$, we need a separate set of completion sets for each of the different worlds and for each concept name G. The completion sets hence now have the following form: $S_*^G(X, v)$ and $S_*^G(X, r, v)$ where X is a concept name, $\top$, or a nominal; $v \in V$ is a world; G is a concept name or $\top$; $* \in \{0, \varepsilon\}$ and r is a role name. The completion sets contain again basic concepts. The intuition behind them is similar to the crisp case: Whenever we have $B \in S_0^G(A, v)$, then this means that $A \sqsubseteq_{\mathcal{T}} P_vB$ and whenever $B \in S_\varepsilon^G(A, v)$, then $P_{>0}A \sqsubseteq_{\mathcal{T}} P_vB$, both under the condition that G has a non-empty interpretation. The completion sets $S_*^G(X, r, v)$ are interpreted analogously.

The reachability relation for Prob-$\mathcal{ELO}_c^{01}$-concepts is similar to the one for $\mathcal{ELOR}$, but it must distinguish between concept names A and probabilistic concepts $P_{>0}A$. For example, non-emptiness of G does not imply non-emptiness of $P_{>0}A$, even if $G \leadsto_R A$, e.g. in worlds with probability 0. Similarly, non-emptiness of G does not imply non-emptiness of A for $G \leadsto_R P_{>0}A$. Therefore we introduce two kinds of reachability relation, $G \overset{0}{\leadsto}_R A$ for $G \leadsto_R A$ and $G \overset{\varepsilon}{\leadsto}_R A$ for $G \leadsto_R P_{>0}A$:

Definition 11. Let $\mathcal{T}$ be a Prob-$\mathcal{ELO}_c^{01}$-TBox in normal form, G be a concept name or $\top$ and D be a concept name or a nominal. Then $G \overset{0}{\leadsto}_R D$ iff there are roles $r_1, \ldots, r_n$ and concept names $A_0, \ldots, A_n$ with $A_i \in S_0^G(A_{i-1}, r_i, 0)$ for all $1 \leqslant i \leqslant n$ such that $A_n = D$ and A_0 is either G or a nominal. Similarly $G \overset{\varepsilon}{\leadsto}_R D$ iff $G \overset{0}{\leadsto}_R X$, $P_{>0}Y \in S_0^G(X, 0)$ and there are roles $r_1, \ldots, r_n$ and concept names $A_0, \ldots, A_n$ with $A_i \in S_\varepsilon^G(A_{i-1}, r_i, \varepsilon)$ for all $1 \leqslant i \leqslant n$ such that $A_n = D$ and $A_0 = Y$.

Nominals interact with the set of possible worlds in a different way than normal concepts. In particular, the concepts $P_{>0}\{a\}$ and $P_{=1}\{a\}$ are actually equivalent to $\{a\}$, since $\{a\}$ is interpreted as the singleton domain element $a^{\mathcal{I}}$ in each world. This means that also $X \sqsubseteq P_{>0}\{a\}$, $X \sqsubseteq P_{=1}\{a\}$ and $X \sqsubseteq \{a\}$ are equivalent, or in terms of completion sets: Whenever $\{a\} \in S_*(X, v)$, then $\{a\}$ must be in $S_*(X, w)$ for all $w \in V$.

PR1	If $C' \in S_*^G(X,v)$ and $C' \sqsubseteq D \in \mathcal{T}$ then $S_*^G(X,v) := S_*^G(X,v) \cup \{D\}$
PR2	If $C_1, C_2 \in S_*^G(X,v)$ and $C_1 \sqcap C_2 \sqsubseteq D \in \mathcal{T}$ then $S_*^G(X,v) := S_*^G(X,v) \cup \{D\}$
PR3	If $C' \in S_*^G(X,v)$ and $C' \sqsubseteq \exists r.D \in \mathcal{T}$ then $S_*^G(X,r,v) := S_*^G(X,r,v) \cup \{D\}$
PR4	If $D \in S_*^G(X,r,v)$, $D' \in S_{\gamma(v)}^G(D,\gamma(v))$ and $\exists r.D' \sqsubseteq E \in \mathcal{T}$ then $S_*^G(X,v) := S_*^G(X,v) \cup \{E\}$, where $\gamma(0)=0$ and $\gamma(v)=\varepsilon$ for all $v \in V \setminus \{0\}$
PR5	If $\{a\} \in S_{*_1}^G(X,*_1) \cap S_{*_2}^G(D,*_2)$ and $G \overset{*_2}{\rightsquigarrow}_R D$ then $S_{*_1}^G(X,*_1) := S_{*_1}^G(X,*_1) \cup \{P_{*_2}D\}$
PR6	If $P_{>0}A \in S_*^G(X,v)$ then $S_*^G(X,P_{>0}A) := S_*^G(X,P_{>0}A) \cup \{A\}$
PR7	If $P_{=1}A \in S_*^G(X,0)$ then $S_*^G(X,1) := S_*^G(X,1) \cup \{A\}$
PR8	If $P_{=1}A \in S_*^G(X,v)$ and $v \neq 0$ then $S_*^G(X,v) := S_*^G(X,v) \cup \{A\}$
PR9	If $A \in S_*^G(X,v)$ and $v \neq 0$, $P_{>0}A \in \mathcal{P}_0^{\mathcal{T}}$ then $S_*^G(X,v') := S_*^G(X,v') \cup \{P_{>0}A\}$
PR10	If $A \in S_*^G(X,1)$ and $P_{=1}A \in \mathcal{P}_1^{\mathcal{T}}$ then $S_*^G(X,v) := S_*^G(X,v) \cup \{P_{=1}A\}$
PR11	If $\{a\} \in S_*^G(X,v)$ then $S_*^G(X,v') := S_*^G(X,v') \cup \{\{a\}\}$

Fig. 5. Completion rules for Prob-$\mathcal{ELO}_c^{01}$.

The completion sets $S_*^G(X,v)$ and $S_*^G(X,r,v)$ are initialized as follows:

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

These completion sets are then extended by applying the completion rules in Fig. 5 exhaustively to compute all subsumption relations. The completion rules can be divided into three groups: the first group of completion rules **PR1** to **PR5** are basically the same as rules **OR1** to **OR4** and **OR7** for $\mathcal{ELOR}$; they compute the subsumption between basic concepts inside each world. The rule **PR4** differs, depending on whether the world v is 0 or any other world in V; i.e., whether the world has positive probability or not. To describe this, we use a function $\gamma : V \to \{0, \varepsilon\}$, where $\gamma(0) = 0$ and $\gamma(v) = \varepsilon$ for all $v \in V \setminus \{0\}$. The next five rules **PR6** to **PR10** handle probabilistic concepts and therefore link the different worlds. For example, whenever we have $P_{>0}A$ in the subsumer set of B, then rule **PR6** will push A into the subsumer set of B for the world $v = P_{>0}A$, i.e., the world v is a witness of the subsumption. Similarly, whenever $P_{=1}A$ is in the subsumer set of B for some world v, then rules **PR7** and **PR10** will push $P_{=1}A$ into the subsumer sets of B for all other worlds w and rule **PR8** will finally put A into the subsumer sets of B for all worlds with non-zero probability (i.e. all worlds except world 0). Lastly, rule **PR11** also links the different worlds by distributing nominals in subsumer sets between all the worlds.

Consider as an example the subsumption $P_{=1}(C \sqcap D) \sqsubseteq P_{>0}C$. This subsumption follows from the empty TBox since $P_{=1}(C \sqcap D) \sqsubseteq P_{=1}C$ holds as does $P_{=1}C \sqsubseteq P_{>0}C$, i.e., each element that is an instance of C in all worlds with non-zero probability is also an instance of C in at least one world with non-zero probability, as there is always at least one such world. As far as the completion algorithm is concerned, any concept that is subsumed by $P_{=1}(C \sqcap D)$ will have C and D as subsumers in all worlds with non-zero probability, i.e., in all worlds except world 0 (by rules **PR7**, **PR10** and **PR8**). But then, this concept will also be subsumed by $P_{>0}C$ in *all* worlds by rule **PR9**, especially in world 0.

Compared to the completion rules for Prob-$\mathcal{EL}_c^{01}$ in [1], there are a few differences: Besides the notation in terms of completion sets S, the rules in this paper do not handle ABox assertions, as these can now simply be absorbed into the TBox. The rules **PR5** and **PR11** which handle nominals inside a world (similar to rule **OR7** for $\mathcal{ELOR}$) and distribute them between the worlds are of course new, as the original algorithm did not handle nominals. Finally, we have introduced the additional rule **PR7**, which is actually necessary to achieve completeness of the completion algorithm even for basic Prob-$\mathcal{EL}_c^{01}$ and was missing in [1]. To see this, consider the following TBox:

$$\mathcal{T}_{ex} = \{A \sqsubseteq P_{=1}B,\ B \sqsubseteq C,\ P_{=1}C \sqsubseteq D\}$$

Clearly, we have $A \sqsubseteq_{\mathcal{T}_{ex}} D$, however, without rule **PR7**, the completion algorithm is stuck with $P_{=1}B \in S_0^A(A,0)$ and will never derive $B \in S_0^A(A,1)$, $C \in S_0^A(A,1)$, $P_{=1}C \in S_0^A(A,0)$ and finally $D \in S_0^A(A,0)$. The completion algorithm for Prob-$\mathcal{ELO}_c^{01}$ is again sound and complete:

Proposition 12. *The completion algorithm is sound:*

$$C \in S_*^G(X,v) \quad \textit{implies} \quad G\colon P_*X \sqsubseteq_{\mathcal{T}} P_vC,$$
$$C \in S_*^G(X,r,v) \quad \textit{implies} \quad G\colon P_*X \sqsubseteq_{\mathcal{T}} P_v\exists r.C.$$

Procedure k-lcs$(C, D, \mathcal{T}, k)$
Input: C, D: Prob-$\mathcal{ELO}^{01}_c$-concepts; $\mathcal{T}$: Prob-$\mathcal{ELO}^{01}_c$-TBox; $k \in \mathbb{N}$
Output: k-lcs(C, D): Prob-$\mathcal{ELO}^{01}_c$-k-lcs of C, D w.r.t. $\mathcal{T}$ and k
$\mathcal{T}' :=$ normalize$(\mathcal{T} \cup \{A \equiv C, B \equiv D\})$
$\mathsf{S}_\mathcal{T} :=$ apply-completion-rules$(\mathcal{T}')$
$L :=$ k-lcs-r$(A, B, \mathsf{S}_\mathcal{T}, k, \mathsf{Sig}(\mathcal{T}))$
return L

Procedure k-lcs-r$(X, Y, \mathsf{S}, k, \mathsf{Sig}(\mathcal{T}))$
Input: X, Y: basic concepts; S: set of saturated completion sets;
$k \in \mathbb{N}$; signature of original TBox $\mathcal{T}$
Output: k-lcs(A, B): Prob-$\mathcal{ELO}^{01}_c$-k-lcs of X, Y w.r.t. $\mathcal{T}$ and k
$\mathsf{CN} := \bigsqcap_{E \in S_0^X(X,0) \cap S_0^Y(Y,0) \cap \mathsf{BC}_\mathcal{T}} E$
if $k = 0$ **then**
  return CN
else
  return $\mathsf{CN} \sqcap \bigsqcap_{r \in \mathsf{Sig}(\mathcal{T}) \cap N_R} \Big($
$$\bigsqcap_{(E,F) \in S_0^X(X,r,0) \times S_0^Y(Y,r,0)} \exists r.\text{k-lcs-r}(E, F, \mathsf{S}, k-1, \mathsf{Sig}(\mathcal{T})) \sqcap$$
$$\bigsqcap_{(E,F) \in S_0^X(X,r,1) \times S_0^Y(Y,r,1)} P_{=1}\big(\exists r.\text{k-lcs-r}(E, F, \mathsf{S}, k-1, \mathsf{Sig}(\mathcal{T}))\big) \sqcap$$
$$\bigsqcap_{(E,F) \in \mathsf{PR}^X(X,r) \times \mathsf{PR}^Y(Y,r)} P_{>0}\big(\exists r.\text{k-lcs-r}(E, F, \mathsf{S}, k-1, \mathsf{Sig}(\mathcal{T}))\big)\Big),$$
where $\mathsf{PR}^G(X, r) = \bigcup_{v \in V \setminus \{0\}} S_0^G(X, r, v)$

Fig. 6. Computation algorithm for role-depth bounded Prob-$\mathcal{ELO}^{01}_c$-lcs.

The completion algorithm is complete, i.e., for a normalized TBox $\mathcal{T}$, a concept name G, a basic concept B that occurs in $\mathcal{T}$, and a role name r we have:

$$G \sqsubseteq_\mathcal{T} B \quad \textit{implies} \quad B \in S_0^G(G, 0)$$
$$G \sqsubseteq_\mathcal{T} \exists r.B \quad \textit{implies} \quad \exists A \textit{ with } A \in S_0^G(G, r, 0) \textit{ and } B \in S_0^G(A, 0)$$

Note that, since $*$ only ranges over $\{0, \varepsilon\}$ for the completion sets S_*, we cannot directly query the completion sets for subsumers of concepts of the form $P_{=1}A$. However, in practice this does not matter: when one wants to know the subsumers of a concept $P_{=1}A$, one can simply introduce a new concept name $X \equiv P_{=1}A$ in the TBox and look up the subsumers of X. This is enough to compute the generalizations, since we introduce new concept names for all input concepts in the k-lcs algorithm anyway, and start from a nominal in the k-msc algorithm, but never directly from a probabilistic concept.

The completion algorithm for Prob-$\mathcal{ELO}^{01}_c$ still runs in polynomial time, since $|\mathsf{BC}_\mathcal{T}|$, $|\mathsf{Sig}(\mathcal{T}) \cap N_C|$, $|V|$, and the number of nominals are all linear in the size of $|\mathcal{T}| = n$. Hence we have $\mathcal{O}(n^3)$ completion sets of the form $S_*^G(X, v)$ and $\mathcal{O}(n^4)$ completion sets of the form $S_*^G(X, r, v)$, each of which contains at most $\mathcal{O}(n)$ many basic concepts. Therefore the number of rule applications that add concepts to the completion sets is bounded by $\mathcal{O}(n^5)$.

Like the previous completion algorithms, it has a 'pay as you go' behavior. For example, the number of worlds that are used in the completion is bounded by the number of occurrences of probabilistic constructors in the knowledge base. If no probabilistic concepts occur, then the algorithm will not introduce additional worlds (except from the three worlds $\{0, 1, \varepsilon\}$).

4.2. Computing the role-depth bounded Prob-$\mathcal{ELO}^{01}_c$-lcs

The computation of the role-depth bounded Prob-$\mathcal{ELO}^{01}_c$-lcs is similar to the classical case, where we intersect the direct subsumers stored in the completion sets and add the cross product of the direct existential restrictions of both concepts. However, in the presence of probabilistic concepts, we need to compute also probabilistic direct subsumers and existential restrictions. Therefore, this algorithm computes the intersections and the cross product three times: Once for the non-probabilistic concepts, once for those concepts with probability one and once for concepts with non-zero probability. The algorithm to compute the role-depth bounded lcs in Prob-$\mathcal{ELO}^{01}_c$ is displayed in Fig. 6.

In this algorithm the TBox is first extended, as before, with fresh concept names A and B, which are used in concept equivalences for the concept descriptions that are the input to the k-lcs. This extended TBox is then normalized and the completion sets are constructed and saturated by the completion rules in Fig. 5. Then, the recursive procedure is called for the concept names A and B and computes the k-lcs.

The recursive procedure k-lcs-r computes the role-depth bounded least common subsumer of two basic concepts X and Y by first taking all concepts that occur in the completions sets of both X and Y for world 0. Note that we do not have

Please cite this article in press as: A. Ecke et al., Completion-based generalization inferences for the Description Logic $\mathcal{ELOR}$ with subjective probabilities, International Journal of Approximate Reasoning (2014), http://dx.doi.org/10.1016/j.ijar.2014.03.001

Procedure k-msc$(a, \mathcal{K}, k)$
Input: a: individual from $\mathcal{K}$; $\mathcal{K}$ a Prob-$\mathcal{ELO}_c^{01}$-TBox; $k \in \mathbb{N}$
Output: role-depth bounded Prob-$\mathcal{EL}_c^{01}$-msc of a w.r.t. $\mathcal{K}$ and k.
$\mathcal{T}' :=$ normalize(absorb-ABox$(\mathcal{K})$)
$\mathsf{S}_{\mathcal{K}} :=$ apply-completion-rules$(\mathcal{T}')$
return traversal-concept$(\{a\}, \mathsf{S}_{\mathcal{K}}, k, \mathsf{Sig}(\mathcal{K}))$

Procedure traversal-concept(A, S, k)
Input: A: basic concept from $\mathcal{T}'$; S: set of completion sets; $k \in \mathbb{N}$
Output: role-depth bounded traversal concept.
$\mathsf{BC} := \bigsqcap_{E \in S_0^A(A,0) \cap \mathsf{BC}_{\mathcal{K}}} E$
if $k = 0$ **then**
  return BC
else
  return $\mathsf{BC} \sqcap \bigsqcap_{r \in \mathsf{Sig}(\mathcal{K}) \cap N_R} \Big($
  $\bigsqcap_{E \in S_0^A(A,r,0)} \exists r.\text{traversal-concept}(E, \mathsf{S}, k-1, \mathsf{Sig}(\mathcal{K})) \sqcap$
  $\bigsqcap_{E \in S_0^A(A,r,1)} P_{=1}\big(\exists r.\text{traversal-concept}(E, \mathsf{S}, k-1, \mathsf{Sig}(\mathcal{K}))\big) \sqcap$
  $\bigsqcap_{E \in \bigcup_{v \in V \setminus \{0\}} S_0^A(A,r,v)} P_{>0}\big(\exists r.\text{traversal-concept}(E, \mathsf{S}, k-1, \mathsf{Sig}(\mathcal{K}))\big)\Big)$

Fig. 7. Algorithm for the role-depth bounded Prob-$\mathcal{EL}_c^{01}$-msc w.r.t. a Prob-$\mathcal{ELO}_c^{01}$-KB.

to check the completion sets for all other worlds, since whenever $X \in S_0^A(A, v)$ with $v \neq 0$, then by completion rule **PR9** we have also that $P_{>0}X \in S_0^A(A, 0)$ and similarly, whenever $X \in S_0^A(A, 1)$ then we likewise have that $P_{=1}X \in S_0^A(A, 0)$ by completion rule **PR10**. If the role-depth bound k is 0, then the conjunction of these concepts is returned.

If k is greater than 0, then the algorithm additionally computes the cross product of all existential restrictions of X and Y for all role names, and recursively calls the k-lcs-r procedure for these existential restrictions. This time, the completion sets have to be traversed for all worlds: concepts in the completion set of world 0 yield existential restrictions $\exists r \ldots$, world 1 yields $P_{=1}\exists r \ldots$ and all worlds except 0 yield restrictions $P_{>0}\exists r \ldots$.

Similar to the non-probabilistic case, one has to make sure that the resulting k-lcs does not contain names introduced by normalization. This is achieved by taking only concept and role names from the original TBox during the construction of the k-lcs. All auxiliary names that were introduced during the normalization can simply be discarded.

Correctness of the role-depth bounded Prob-$\mathcal{ELO}_c^{01}$-lcs follows directly from soundness and completeness of the completion rules which are used to generate the underlying completion sets.

Theorem 13. *Let $\mathcal{T}$ be a Prob-$\mathcal{ELO}_c^{01}$-TBox, C and D be Prob-$\mathcal{ELO}_c^{01}$-concept names and k be a natural number. Then k-lcs$(C, D, \mathcal{T}, k)$ is the role-depth bounded Prob-$\mathcal{ELO}_c^{01}$-lcs of C and D w.r.t. $\mathcal{T}$ and the role-depth k.*

As in the classical case, the resulting k-lcs can have a size exponential in k, but it is still polynomial in the size of the input TBox $\mathcal{T}$ for a fixed k.

4.3. Computing role-depth bounded Prob-$\mathcal{EL}_c^{01}$-msc w.r.t. Prob-$\mathcal{ELO}_c^{01}$-KBs

As before, our role-depth bounded msc w.r.t. a Prob-$\mathcal{ELO}_c^{01}$-KB does not contain nominals (i.e., is a Prob-$\mathcal{EL}_c^{01}$-concept), since the msc becomes trivial otherwise. As a first step, the ABox is absorbed into the TBox, so that the completion algorithm as defined in Subsection 4.1 can be applied. To look up those concepts of which an individual a is an instance in the original knowledge base, we can then simply look for subsumers of the nominal $\{a\}$. This way, we can build the traversal concept for an individual by traversing the completion sets up to a certain role-depth, which gives exactly the role-depth bounded Prob-$\mathcal{EL}_c^{01}$-msc of an individual from the Prob-$\mathcal{ELO}_c^{01}$-KB.

Clearly, the msc needs to include probabilistic concepts. This is done in the same way as for the role-depth bounded lcs, by traversing the subsumer sets for existential restriction for all worlds V. The algorithm to compute the role-depth bounded Prob-$\mathcal{EL}_c^{01}$-msc w.r.t. a Prob-$\mathcal{ELO}_c^{01}$-KB is depicted in Fig. 7.

The algorithm first absorbs the ABox into the TBox, normalizes the resulting TBox and then constructs the completion sets. These sets are traversed in the procedure traversal-concept, starting from the nominal $\{a\}$ of the given individual a. The traversal first gathers all basic concepts that are direct subsumers of the current node. For these, the algorithm only needs to traverse the world 0, since whenever $X \in S_0^A(A, v)$ with $v \neq 0$, then by completion rule **PR9** we have $P_{>0}X \in S_0^A(A, 0)$ and similarly, whenever $X \in S_0^A(A, 1)$ then we also have $P_{=1}X \in S_0^A(A, 0)$ by completion rule **PR10**. For traversing the roles however, the algorithm has to look at the completion sets for all worlds and recursively traverse all concepts from these completion sets.

Correctness of the computation of the role-depth bounded Prob-$\mathcal{EL}_c^{01}$-msc w.r.t. Prob-$\mathcal{ELO}_c^{01}$-KBs follows directly from soundness and completeness of the completion rules which are used to generate the underlying completion sets. The size of the Prob-$\mathcal{EL}_c^{01}$-k-msc is again exponential in k, but polynomial in the size of $\mathcal{K}$.

Theorem 14. *Let $\mathcal{K}$ be a Prob-$\mathcal{ELO}_c^{01}$-KB, a be an individual and k be a natural number. Then* $\mathsf{k\text{-}msc}(a, \mathcal{K}, k)$ *is the role-depth bounded Prob-$\mathcal{EL}_c^{01}$-msc of a w.r.t. $\mathcal{K}$ and the role-depth k.*

5. Optimization and evaluation

As previously pointed out, the concept descriptions returned by the algorithms from the generalization algorithms described in the previous sections can grow exponentially large in the role-depth bound k in the worst case. On top of that, the returned concept descriptions are often highly redundant, which might be acceptable if used as an input for a similarity measure, but surely not if presented to a human reader. This high redundancy can be formalized using the following notion:

A concept description C is called *fully expanded* up to the role-depth k w.r.t. a TBox $\mathcal{T}$, if

1. for all $A \in \mathsf{BC}_\mathcal{T}$ with $C \sqsubseteq_\mathcal{T} A$ we have that A is a conjunct of C and
2. if $k > 0$ then for all concepts F with $C \sqsubseteq_\mathcal{T} \exists r.F$ we have an F' with $F' \sqsubseteq_\mathcal{T} F$ such that $\exists r.F'$ is a conjunct of C and F' is fully expanded up to role-depth $k-1$.

Both the k-lcs and k-msc procedures construct concept descriptions that are fully expanded. This can be illustrated by the introductory example, where we described a knowledge base similar to the following:

$$\begin{aligned}\mathcal{T} = \{ & \mathsf{Mother} \equiv \mathsf{Woman} \sqcap \exists \mathsf{has} - \mathsf{child}.\top, \\ & \mathsf{Obese} \sqsubseteq P_{=1} \exists \mathsf{hasCondition.HighPressure} \}, \\ \mathcal{A} = \{ & \mathsf{Woman(mary)}, \\ & \mathsf{has\text{-}child(mary, peter)}, \\ & \mathsf{Obese(mary)} \}.\end{aligned}$$

For this KB, the k-msc algorithm would compute the fully expanded concept $\mathsf{Obese} \sqcap P_{=1}\exists\mathsf{hasCondition.HighPressure} \sqcap \mathsf{Mother} \sqcap \mathsf{Woman} \sqcap \exists\mathsf{has\text{-}child}.\top$, which, even for this small KB, already contains quite some redundancies; in particular, the condition of high pressure is already implied by the concept Obese, while being a woman and having a child is implied by the concept Mother.

In this section, we introduce simplification and optimization procedures that are implemented in our system Gel [42], that allow us to output shorter and thus easier to understand—but still equivalent—concepts. In particular, these simplifications aid in speeding-up the generalization inferences. We then evaluate both our algorithms and the presented optimizations. Note that Gel only implements the generalization algorithms for $\mathcal{ELOR}$, since there is no implementation of the classification algorithm for probabilistic variants of $\mathcal{EL}$ available. Therefore, we introduce here some of the improvements for $\mathcal{ELOR}$ alone, rather than for full Prob-$\mathcal{ELOR}$. However, all of these improvements are easily applicable to the Prob-$\mathcal{ELO}_c^{01}$-variants of the generalization algorithms as well.

We now present two types of improvements for the algorithms: to obtain succinct rewritings of $\mathcal{ELOR}$-concept descriptions and to speed-up the k-lcs computation.

5.1. Simplifying $\mathcal{ELOR}$-concept descriptions

The fully expanded $\mathcal{ELOR}$-concept descriptions obtained from the k-lcs and k-msc algorithms need to be simplified, in order to make the resulting concept description readable to humans. The general idea for the simplification is to remove those subtrees from the syntax tree of the concept description which are subsumers of any of their sibling subtrees. For a conjunction of concept names, this results in removing all concept names except the minimal ones (w.r.t. $\sqsubseteq_\mathcal{T}$)—yielding the smallest equivalent $\mathcal{ELOR}$-concept.

The algorithm shown in Fig. 8 computes such simplifications for $\mathcal{ELOR}$-concept descriptions. For the correctness of the simplification procedure simplify, it is only necessary to ensure that the procedure subsumes-H is sound. However, for our purpose this procedure does not need to be complete, in the sense that the simplification can yield concepts that are equivalent to the input concept description but contain still some redundancies. The heuristic used in Gel is displayed in Fig. 9; it tries to find a syntactic argument for the subsumption by traversing both concepts structurally and using the computed subsumptions between concept names from the completion sets.

It is easy to see that the procedure subsumes-H is sound by an inspection of the different cases according to the structure of C and D. For instance, if $C = F_1 \sqcap F_2$ is a conjunction (line 9), the procedure only returns true if D is a subsumer of both

```
Procedure simplify(C, S_T, T)
Input: C: ELOR-concept; S_T: saturated completion sets; T: ELOR-TBox
Output: simplify(C): a succinct ELOR-concept equivalent to C w.r.t. T.
Let C = A_1 ⊓ ··· ⊓ A_n ⊓ ∃r_1.D_1 ⊓ ··· ⊓ ∃r_m.D_m with A_i ∈ BC_T (1 ⩽ i ⩽ n)
Conjuncts := {A_i | 1 ⩽ i ⩽ n} ∪ {∃r_j.D_j | 1 ⩽ j ⩽ m}
for all X ∈ Conjuncts do
   for all Y ∈ Conjuncts do
      if X ≠ Y ∧ subsumes-H(X, Y, S_T, T) then
         Conjuncts := Conjuncts \ {X}
         break
for all X ∈ Conjuncts do
   if X = ∃r_j.D_j then
      Conjuncts := (Conjuncts \ {∃r_j.D_j}) ∪ {∃r_j.simplify(D_j, S_T, T)}
return ⨅_{X∈Conjuncts} X
```

Fig. 8. Simplification algorithm for $\mathcal{ELOR}$-concept descriptions w.r.t. an $\mathcal{ELOR}$-TBox.

```
Procedure subsumes-H(C, D, S, T)
Input: C, D: ELOR-concepts; S: completion sets; T: ELOR-TBox
Output: Boolean value indicating whether D ⊑_T C
 1: if C ∈ BC_T then
 2:    if C = ⊤ then
 3:       return true
 4:    else if D ∈ BC_T then
 5:       return C ∈ S(D)
 6:    else if D = F_1 ⊓ F_2 then
 7:       return subsumes-H(C, F_1, S, T) ∨ subsumes-H(C, F_2, S, T)
 8:    return false
 9: else if C = F_1 ⊓ F_2 then
10:    return subsumes-H(F_1, D, S, T) ∧ subsumes-H(F_2, D, S, T)
11: else if C = ∃r.F then
12:    if D ∈ BC_T then
13:       return ∃E ∈ S(D, r) with F ∈ S(E)
14:    else if D = F_1 ⊓ F_1 then
15:       return subsumes-H(C, F_1, S, T) ∨ subsumes-H(C, F_2, S, T)
16:    else if D = ∃s.G then
17:       if s ⊑_T r and subsumes-H(F, G, S, T) then
18:          return true
19:       else if ∃t ∈ N_R such that s ∘ t ⊑_T r, G = ··· ⊓ ∃t.H ⊓ ... and
          subsumes-H(F, H, S, T) then
20:          return true
21:       return false
```

Fig. 9. Subsumption heuristic for $\mathcal{ELOR}$-concept descriptions based on completion sets.

conjuncts F_1 and F_2. If both $C = \exists r.F$ and $D = \exists s.G$ are existential restrictions (line 16), then the procedure returns true if $s \sqsubseteq_{\mathcal{T}} r$ and F is a subsumer of G, or if there is a role t with $s \circ t \sqsubseteq_{\mathcal{T}} r$ and G contains a top-level conjunct of the form $\exists t.H$ such that F is a subsumer of H. In both cases we clearly have $D \sqsubseteq_{\mathcal{T}} C$. Soundness for all other cases can be shown similarly.

Note that this simplification procedure can reduce the size of concepts massively. For example, for the TBox $\mathcal{T} = \{A \sqsubseteq \exists r.A \sqcap \exists s.A, B \sqsubseteq A\}$, the fully expanded k-lcs of A and B has size 2^k, while the simplified k-lcs is always of size 1: it is simply A. However, in case of the empty TBox, the fully expanded generalizations can usually not be simplified any further. Thus the effectiveness of simplification largely depends on the structure of the TBox.

5.2. Speeding-up the k-lcs algorithm

As explained before, the k-lcs always returns fully expanded concept descriptions. Even though most redundancy can be removed by the simplification procedure, this seems counter-intuitive: Why generate the fully expanded concept in the first place, if most of it gets removed afterwards anyway? Especially for large ontologies with a deep role hierarchy the fully expanded result may grow very large, which causes in turn long runtimes of the generalization algorithm. Therefore, the general idea for optimizations is to avoid generating this redundancy and apply some of the simplifications already during the construction of the result.

Optimization 1 *(Avoid unnecessary role-depth).* This simple optimization applies if one of the input concepts of the k-lcs-r procedure already subsumes the other one, in which case it is the lcs of both. Therefore in the procedure

Please cite this article in press as: A. Ecke et al., Completion-based generalization inferences for the Description Logic $\mathcal{ELOR}$ with subjective probabilities, International Journal of Approximate Reasoning (2014), http://dx.doi.org/10.1016/j.ijar.2014.03.001

k-lcs-r($A, B, \mathsf{S}_\mathcal{T}, k, \mathsf{Sig}(\mathcal{T})$), if $A \in S^B(B)$ we can simply return A and if $B \in S^A(A)$ we can return B. However, if in the first case A or in the second case B are normalization names, we still have to traverse the completion sets instead of simply returning $\top$. This subsumption check is a well-known optimization for the computation of the lcs for concept descriptions without reference to a TBox [43,44].

Optimization 2 *(Avoid unnecessary branching).* As motivation for this optimization consider the TBox

$$\mathcal{T}_n = \{A \sqsubseteq D \sqcap \exists r.C_i, B \sqsubseteq E \sqcap \exists r.C_i \mid 1 \leqslant i \leqslant n\} \cup \{C_1 \sqsubseteq C_i \mid 2 \leqslant i \leqslant n\}$$

In this case, the lcs of A and B is simply $\exists r.C_1$. However, the naive k-lcs algorithm would generate the complete product set

$$\{C_i \mid 1 \leqslant i \leqslant n\} \times \{C_j \mid 1 \leqslant j \leqslant n\}$$

and recursively call k-lcs-r for each pair, just to eliminate all $\exists r.C_i$ for $i > 1$ and all $\exists r.\top$ afterwards in the simplification step. Even for this simple example, the algorithm would require time quadratic in the size of the input TBox. Clearly, evaluating the C_is for $i > 1$ is not necessary, as they all subsume C_1. The same is true for role hierarchies, where for example

$$\mathcal{T}'_n = \{A \sqsubseteq D \sqcap \exists r.C_1, B \sqsubseteq E \sqcap \exists r.C_1\} \cup \big\{r \sqsubseteq r_i \mid i \in \{2 \ldots n\}\big\}$$

would lead to the same unnecessary blow-up of the concept description, and thus also needlessly increase the runtime of the k-lcs algorithm.

The idea to avoid this kind of branching, is at each step of the traversal to explicitly create the sets SA and SB of all role-successors $C \in S^X(X, r)$ and $D \in S^Y(Y, s)$ for the current basic concepts X and Y and to remove all role-successors which are subsumers of other role-successors in the same set (see Fig. 10). Recursive calls to k-lcs-r-o can then be made with one successor from each set SA and SB. However, we have to be careful with role-successors with different role-names. For example, for a role-successor $(r, C) \in$ SA and a role-successor $(s, D) \in$ SB, a recursive call has to be made for all minimal (w.r.t. $\sqsubseteq_\mathcal{T}$) role names t with $r \sqsubseteq_\mathcal{T} t$ and $s \sqsubseteq_\mathcal{T} t$ (see Fig. 10).

Optimization 3 *(Avoid computing unnecessary completion sets).* When traversing the completion graphs using the generalization algorithms from Section 3, one has to compute the completion sets $S^A(X)$ for all concept names A being traversed. However, it is often the case that most of these sets are very similar, or even the same. For example, whenever $A \rightsquigarrow_R B$ for concept names $A, B \in \mathsf{Sig}(\mathcal{T}) \cap N_C$, then the completion sets $S^A(X)$ always contain all (conditional) subsumptions that also $S^B(X)$ contains, i.e., $S^B(X) \subseteq S^A(X)$ and $S^B(X, r) \subseteq S^A(X, r)$ for all basic concepts X and role names r. The reason is that, whenever $A \rightsquigarrow_R B$, then non-emptiness of A also implies non-emptiness of B and thus, whenever a conditional subsumption $B : X \sqsubseteq_\mathcal{T} Y$ holds if B is non-empty, then it also holds if A is non-empty: $A : X \sqsubseteq_\mathcal{T} Y$.

For the k-lcs algorithm, this observation means that computing the completion sets $S^X(Z)$ and $S^Y(Z)$ for all traversed concepts X and Y is unnecessary, since all traversed concepts are always reachable from the initial concept names A and B. That is, if k-lcs-r is applied to the concept pair (C, D), then it holds that $A \rightsquigarrow_R C$ and $B \rightsquigarrow_R D$ and thus all subsumers of C and D can also be found in the completion sets for A and B. Therefore, it suffices to use the completion sets $S^A(X)$ and $S^B(X)$ for the initial concept names A and B only, thus also decreasing the complexity of the completion algorithm compared to full classification.

The case is similar for the k-msc algorithm. However, since this algorithm always starts with a nominal, and nominals are reachable from every concept by definition, it suffices to compute the completion sets $S^\top(X)$ and look up all subsumption relations in these completion sets.

Optimization 4 *(Avoid redundant probabilistic concepts).* Optimization 2 can be generalized to probabilistic existential restrictions. While restrictions of the form $\exists r.C$ and $P_{>0}\exists r.C$ or $P_{=1}\exists r.C$ are completely independent from each other, this is not true for $P_{>0}\exists r.C$ and $P_{=1}\exists r.C$. Indeed $P_{=1}\exists r.C \sqsubseteq_\mathcal{T} P_{>0}\exists r.C$ always holds and whenever the k-lcs-r function finds a subsumer $P_{=1}\exists r.C$ for a concept X, it also finds a subsumer $P_{>0}\exists r.C$ by the method in which the completion sets are traversed. This again leads to unwanted redundancies.

The idea to avoid this kind of blow-up in the k-lcs-r algorithm is to traverse probabilistic existential restrictions of the form $P_{>0}\exists r.$k-lcs-r$(E, F, \mathsf{S}, k, \mathsf{Sig}(\mathcal{T}))$ only if $E \in S_0^A(X, r, v)$ and $F \in S_0^B(Y, r, v')$ for $v, v' \neq 0$ and not both $E \in S_0^A(X, r, 1)$ and $F \in S_0^B(Y, r, 1)$. Similarly, we can avoid this redundancy in the k-msc algorithm by traversing $P_{>0}\exists r.$traversal-concept$(E, \mathsf{S}, k, \mathsf{Sig}(\mathcal{K}))$ only if $E \in S_0^\top(X, r, v)$ for $v \neq 0$ and $E \notin S_0^\top(X, r, 1)$.

Fig. 10 shows the optimized k-lcs algorithm for $\mathcal{ELOR}$ implementing optimizations 1, 2 and 3. We show that these optimizations are correct.

Lemma 15. *The results of the k-lcs-r procedure for $\mathcal{ELOR}$-TBoxes given in Fig. 3 and the k-lcs-r-o procedure in Fig. 10 are equivalent.*

Proof. Let $\mathcal{T}$ be a TBox, $\mathcal{T}'$ be its normalized version, $\mathsf{S}_\mathcal{T}$ be the completion sets, $A, B \in \mathsf{Sig}(\mathcal{T}) \cap N_C$, $X, Y \in \mathsf{BC}_{\mathcal{T}'}$ such that $A \rightsquigarrow_R X$, $B \rightsquigarrow_R Y$ and $k \in \mathbb{N}$. Let $L =$ k-lcs-r$(X, Y, \mathsf{S}_\mathcal{T}, k, \mathsf{Sig}(\mathcal{T}))$ and $L_o =$ k-lcs-r-o$(X, Y, \mathsf{S}_\mathcal{T}, k, A, B, \mathsf{Sig}(\mathcal{T}))$. We have to

Procedure k-lcs-o($C, D, \mathcal{T}, k$)
Input: C, D: $\mathcal{ELOR}$-concept descriptions; $\mathcal{T}$: $\mathcal{ELOR}$-TBox; $k \in \mathbb{N}$
Output: role-depth bounded $\mathcal{ELOR}$-lcs of C, D w.r.t. $\mathcal{T}$ and k

$\mathcal{T}' :=$ normalize($\mathcal{T} \cup \{A \equiv C, B \equiv D\}$)
$\mathsf{S}_\mathcal{T} :=$ apply-completion-rules($\mathcal{T}'$)
$L :=$ k-lcs-r-o($A, B, \mathsf{S}_\mathcal{T}, k, A, B, \mathsf{Sig}(\mathcal{T}) \cup \{A, B\}$)
if $L = A$ **then**
  return C
else if $L = B$ **then**
  return D
else
  return L

Procedure k-lcs-r-o($X, Y, \mathsf{S}_\mathcal{T}, k, A, B, \mathsf{Sig}(\mathcal{T})$)
Input: $A, B \in N_C$; X, Y: basic concepts with $A \rightsquigarrow_R X$, $B \rightsquigarrow_R Y$; $k \in \mathbb{N}$
$\mathsf{S}_\mathcal{T}$: set of saturated completion sets; signature of original TBox $\mathcal{T}$
Output: role-depth bounded $\mathcal{ELOR}$-lcs of A, B w.r.t. $\mathcal{T}$ and k

if $X \in S^B(Y)$ and $X \in \mathsf{BC}_\mathcal{T}$ **then**
  return X
else if $Y \in S^A(X)$ and $Y \in \mathsf{BC}_\mathcal{T}$ **then**
  return Y
common-names $:= S^A(X) \cap S^B(Y) \cap \mathsf{BC}_\mathcal{T}$
if $k = 0$ **then**
  return $\bigsqcap_{P \in \text{common-names}} P$
else
  SA := remove-redundant($\{(r, C) \mid C \in S^A(X, r)\}$)
  SB := remove-redundant($\{(s, D) \mid D \in S^B(Y, s)\}$)
  return $\bigsqcap_{P \in \text{common-names}} P \sqcap \bigsqcap_{\substack{(r,C)\in \mathrm{SA},\ (s,D)\in \mathrm{SB},\\ t \in \mathsf{Sig}(\mathcal{T}) \cap N_R \text{ minimal with } r \sqsubseteq_\mathcal{T} t \wedge s \sqsubseteq_\mathcal{T} t}} \exists t.\text{k-lcs-r-o}(C, D, \mathsf{S}_\mathcal{T}, k-1, A, B, \mathsf{Sig}(\mathcal{T}))$

Procedure remove-redundant($S, \mathsf{S}_\mathcal{T}$)
Input: S: set of role-successors; $\mathsf{S}_\mathcal{T}$: set of saturated completion sets
Output: simplified set

for all $(r, C) \in S$ **do**
  for all $(s, D) \in S$ **do**
    if $(r, C) \neq (s, D)$ and $r \sqsubseteq_\mathcal{T} s$ and $D \in S(C)$ **then**
      $S := S \setminus \{(s, D)\}$
return S

Fig. 10. Computation of role-depth bounded $\mathcal{ELOR}$-lcs.

show that $L \equiv_\mathcal{T} L_o$. Note that if Optimization 1 is applicable, then $X \in S(Y)$ or $Y \in S(X)$ and hence X (resp. Y) is the lcs and thus equivalent to L. Otherwise, we show that $L \equiv_\mathcal{T} L_o$ by induction on the role-depth bound k.

For $k = 0$ both procedures return the same concept description, i.e., $L = L_o$.

For $k > 0$, we first show that for each $r \in \mathsf{Sig}(\mathcal{T}) \cap N_R$ and all $C \in S^X(X, r)$, $D \in S^Y(Y, r)$, we have $L_o \sqsubseteq_\mathcal{T} \exists r.\text{k-lcs-r}(C, D, \mathsf{S}_\mathcal{T}, k-1, \mathsf{Sig}(\mathcal{T}))$. For every $C \in S^X(X, r)$ there exists $(s_1, C') \in \mathrm{SA}$ with $s_1 \sqsubseteq_\mathcal{T} r$ and $C \in S(C')$. Similarly, there exists $(s_2, D') \in \mathrm{SB}$ with $s_2 \sqsubseteq_\mathcal{T} r$ and $D \in S(D')$. Therefore, we know that $\exists t.\text{k-lcs-r-o}(C', D', \mathsf{S}_\mathcal{T}, k-1, A, B, \mathsf{Sig}(\mathcal{T}))$ is a conjunct of L_o for all minimal $t \in N_{R,\mathcal{T}}$ with $s_1 \sqsubseteq_\mathcal{T} t$ and $s_2 \sqsubseteq_\mathcal{T} t$. Since $s_1 \sqsubseteq_\mathcal{T} r$ and $s_2 \sqsubseteq_\mathcal{T} r$, there is at least one minimal $t_0 \in \mathsf{Sig}(\mathcal{T}) \cap N_R$ with $t_0 \sqsubseteq r$ for which $\exists t_0.\text{k-lcs-r-o}(C', D', \mathsf{S}_\mathcal{T}, k-1, A, B, \mathsf{Sig}(\mathcal{T}))$ is conjunct of L_o.

The induction hypothesis yields

$$\text{k-lcs-r-o}\big(C', D', \mathsf{S}_\mathcal{T}, k-1, A, B, \mathsf{Sig}(\mathcal{T})\big) \equiv_\mathcal{T} \text{k-lcs-r}\big(C', D', \mathsf{S}_\mathcal{T}, k-1, \mathsf{Sig}(\mathcal{T})\big)$$

and we know by definition of the k-lcs that

$$\text{k-lcs-r}\big(C', D', \mathsf{S}_\mathcal{T}, k-1, \mathsf{Sig}(\mathcal{T})\big) \sqsubseteq_\mathcal{T} \text{k-lcs-r}\big(C, D, \mathsf{S}_\mathcal{T}, k-1, \mathsf{Sig}(\mathcal{T})\big)$$

for $C' \sqsubseteq_\mathcal{T} C$ and $D' \sqsubseteq_\mathcal{T} D$. Thus,

$$\exists t_0.\text{k-lcs-r-o}\big(C', D', \mathsf{S}_\mathcal{T}, k-1, A, B, \mathsf{Sig}(\mathcal{T})\big) \sqsubseteq_\mathcal{T} \exists r.\text{k-lcs-r}\big(C, D, \mathsf{S}_\mathcal{T}, k-1, \mathsf{Sig}(\mathcal{T})\big),$$

i.e. $L_o \sqsubseteq_\mathcal{T} \exists r.E$ for all conjuncts $\exists r.E$ in L. Since it also holds that $L_o \sqsubseteq_\mathcal{T} C$ for $C \in$ common-names, we have $L_o \sqsubseteq_\mathcal{T} L$. Obviously, k-lcs-r computes all of the recursive concept descriptions (and possibly more) that k-lcs-r-o computes and hence $L \sqsubseteq_\mathcal{T} L_o$. This finally yields $L \equiv_\mathcal{T} L_o$. □

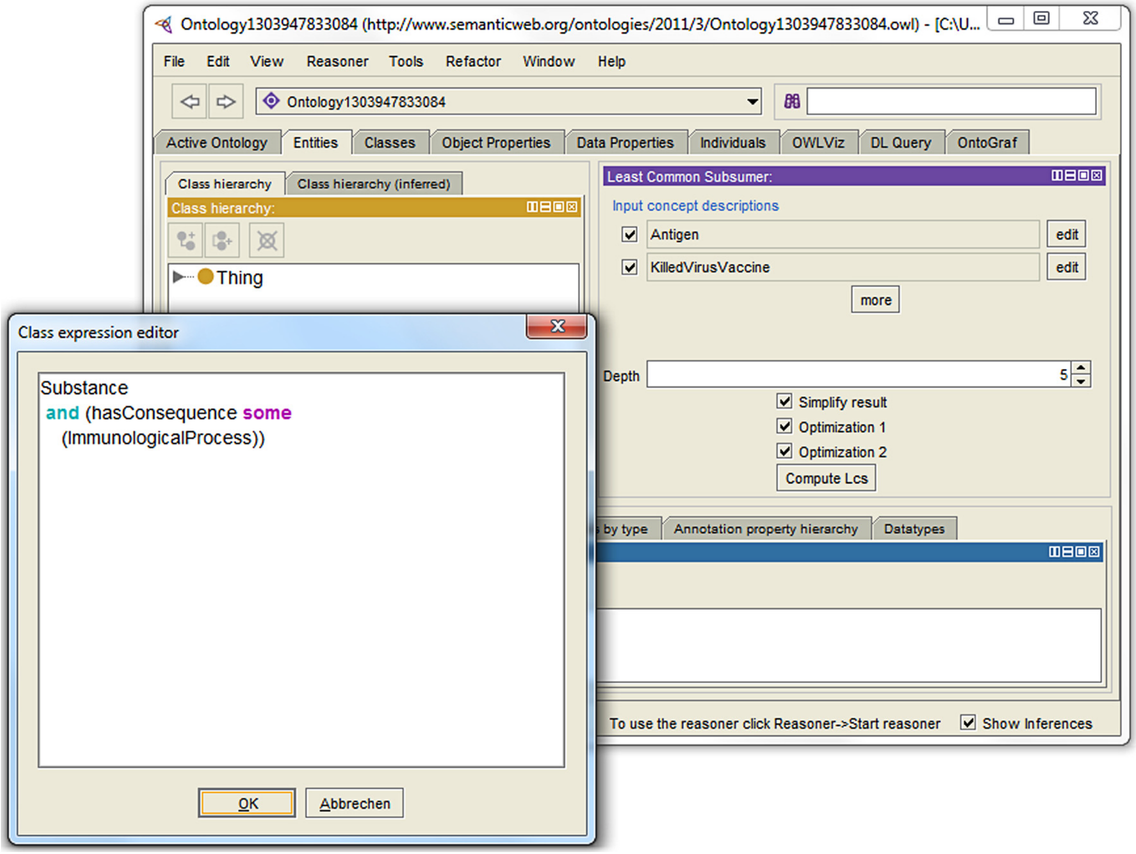

Fig. 11. Screen-shot of the PROTÉGÉ plug-in of GEL.

Note that, while we presented the first three optimizations for the k-lcs algorithm in terms of $\mathcal{ELOR}$, they can easily be lifted to Prob-$\mathcal{ELO}^{01}_{c}$. Additionally, Optimization 3 is easy to include in the k-msc algorithm, by simply replacing all completion sets $S^A(X)$ in k-msc-r by $S^\top(X)$. As described before, all four optimizations perform some of the simplification steps during the construction of the lcs. However, they are not exhaustive; thus the resulting generalizations still contain some redundancies that can be removed by subsequently applying the simplification procedure.

5.3. Evaluation

The completion algorithm for classifying crisp $\mathcal{EL}$-TBoxes was first implemented in the CEL reasoner [45]. We used its successor system JCEL [37] as a starting point for our implementation of the role-depth bounded lcs and msc. Our system GEL, first introduced in [37], allows for the computation of both k-lcs and k-msc in the classical description logic $\mathcal{ELOR}$. Note that, while GEL handles nominals adequately in most cases, its implementation is based on the completion algorithm published in [4] which is sound but not complete for TBoxes containing nominals, as shown in Kazakov et al. [32]. However, if applied to ontologies from applications, the algorithm employed in JCEL usually behaves well, yielding the same result as the complete classification algorithm. In particular, this is true for our test ontologies. We did not implement the generalization algorithms for the probabilistic variants, since currently no completion-based reasoner for Prob-$\mathcal{EL}$ or its extensions is available and neither are test knowledge bases. GEL is implemented in Java and provides a simple GUI for the ontology editor PROTÉGÉ, shown in the screen-shot in Fig. 11.

We tested our system GEL extensively with constructed and real-world knowledge bases. The real-world ontologies used in our tests were the Gene Ontology [46] and NOT-GALEN, a version of the GALEN ontology [47] pruned to the expressivity of the DL $\mathcal{ELR}$. Since most random tuples of concepts in these ontologies have no commonalities, their k-lcs would trivially be $\top$. To exclude such uninteresting cases, we selected approximately 50 tuples of named concepts by hand, selecting concepts classified as sibling concepts that had similar existential restrictions. We computed the k-lcs for these input concepts with various role-depth bounds—both with and without the first two optimizations. The third optimization was always enabled, but note that none of the real-world ontologies we found for testing contained enough nominals.

In the worst case the role-depth bounded lcs can have a size that is exponential in the role-depth bound k. However, it largely depends on the ontology whether such worst-case behavior occurs. For the Gene Ontology, the role-depth bounded

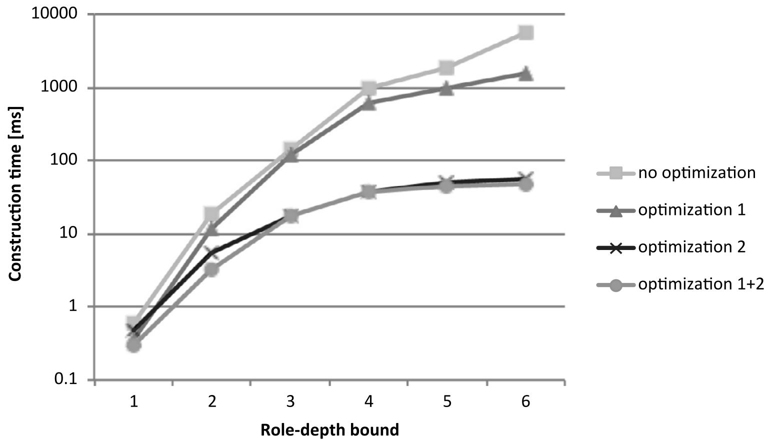

Fig. 12. Average k-lcs-r-construction time for concept descriptions from NOT-GALEN.

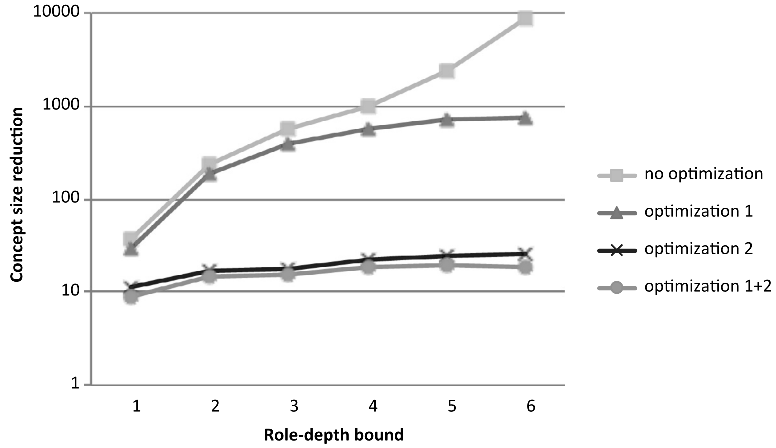

Fig. 13. Average factor of the reduction of concept sizes by the simplification procedure for the k-lcs computed w.r.t. NOT-GALEN.

lcs was always constructed and simplified almost instantly. The runtime was totally dominated by the classification time used by JCEL.

For NOT-GALEN on the other hand, a few input concept pairs resulted in long runtimes, most of this time was spent in the k-lcs-r-procedure and not in the classification step. Fig. 12 shows the average k-lcs-r-construction runtime of GEL on various input pairs for different values of k. Please note the use of a logarithmic scale. The figure also shows the effect of the optimizations on the construction time. Optimization 1, which returns one of the input concepts if it subsumes the other, is able to cut-off the k-lcs-r-o recursion before the maximum role-depth is reached. This improves the runtime by a factor of about 2 compared to the basic k-lcs-r procedure with no optimizations. Optimization 2, which removes redundant successor nodes before product construction, reduces the branching factor of the recursion. In our tests, it yielded even better runtime improvements than Optimization 1—on average by factor 30. Each optimization yields a larger speed-up for increasing role-depth bounds. Combining the optimizations yielded the best runtime for most cases, which indicates that the optimizations are independent.

In practice, the runtime with only Optimization 1 (or no optimization at all) was sometimes too long to be useful. For some input concepts, like *PepticUlcer* and *AreaOfAtrophicGastritis*, it did not return any result within an hour for a role-depth bound as low as 4. However, with both optimizations enabled, the result for a role-depth bound of 4 was computed in 1.9 s (with a classification time of 330 ms). The reason that Optimization 2 is so effective on NOT-GALEN is that this ontology contains a deep role hierarchy. Without the optimization for each role also all subsuming roles would generate a recursive k-lcs-r call, which yields large branching factors.

The runtime for the simplification method which rewrites the resulting concepts into shorter ones is proportional in the size of the concept description before simplification. However, as illustrated by Fig. 13, simplification reduces the concept sizes dramatically—even with both optimizations enabled the size of the result is still reduced by a factor of 16 on average. Of course, since both optimizations apply simplification steps during the construction of the k-lcs, the size-reduction for results computed without optimizations is even larger.

All in all, the results from Gel with the first two optimizations and the simplification method active were computed generally quite fast and were small enough to be inspected manually, thus we believe it is practical enough for many applications.

We were not able to test the performance of the generalization algorithms for Prob-$\mathcal{ELO}_c^{01}$, but we assume that it scales with the amount of probabilistic constructors used in the knowledge base: If the Prob-$\mathcal{ELO}_c^{01}$-KB contains no probabilities at all, the generalization algorithms will not introduce any existential restrictions of the form $P_{>0}\exists r.C$ or $P_{=1}\exists r.C$ and thus behave exactly the same as the algorithms for $\mathcal{ELOR}$. We conjecture that when applying Optimization 4, which reduces the number of redundant probabilistic concepts, the overall performance of the k-lcs or k-msc construction in Prob-$\mathcal{ELO}_c^{01}$ is only slightly worse compared to the respective algorithms for $\mathcal{ELOR}$.

6. Conclusions

In this paper we have studied extensions of the light-weight description logic $\mathcal{EL}$ that are capable of handling uncertainty. The full-blown logic Prob-$\mathcal{ELOR}$ extends the classical DL $\mathcal{ELOR}$ with probabilistic concept constructors and probabilistic assertions. The former can be used to describe the class of individuals that belong to a concept within some probabilistic bounds, while the latter provides a range of probabilities for an individual to satisfy a given concept. Formally, Prob-$\mathcal{ELOR}$ augments classical concepts with subjective (or Type 2) probabilities, which are interpreted through multiple-world semantics.

One of the characterizing features of $\mathcal{EL}$ and $\mathcal{ELOR}$ is that they allow for polynomial time reasoning. This feature, which is necessary for the feasibility of reasoning with huge knowledge bases, such as Snomed CT, is unfortunately lost in Prob-$\mathcal{EL}$ and Prob-$\mathcal{ELOR}$. Thus, we restrict our attention to the sublogic Prob-$\mathcal{ELOR}_c^{01}$, in which probabilities can only be used in expressions of the form “the probability is 1” or “the probability is greater than 0.” Despite its seemingly low expressivity, this logic extends the language currently used for the representation of large biomedical knowledge bases, and is capable of expressing relevant probabilistic concepts.

The contribution of the paper is manifold. First, we have shown that standard reasoning remains tractable in the logic Prob-$\mathcal{ELOR}_c^{01}$. We provide a completion algorithm that generalizes the previously known algorithm for Prob-$\mathcal{EL}_c^{01}$, with correct rules for handling nominals. As a side benefit, we obtain a completion algorithm for classical $\mathcal{ELOR}$ that exhibits a pay-as-you-go behavior, and cover, with a few exceptions like datatypes, the whole OWL 2 EL profile.

Second, we describe how the completion sets obtained from the completion algorithm can be combined to compute (approximations of) the most specific concept and the least common subsumer. The most specific concept of a given individual is the smallest (w.r.t. subsumption) concept description that contains a given individual. Since Prob-$\mathcal{ELOR}_c^{01}$ allows for nominals, which are concepts that only contain a named individual, this task is trivial in that logic. Thus, we restricted our attention to the use of the target language Prob-$\mathcal{EL}_c^{01}$: finding the most specific Prob-$\mathcal{EL}_c^{01}$-concept of which a given individual must be an instance of. The least common subsumer, on the other hand, is the most specific concept that generalizes two given concepts. Both, the msc and the lcs do not exist in general, even for the classical logic $\mathcal{EL}$. However, we approximate them up to a role-depth that can be specified by the user. Our approximation has the characteristic that, the larger the allowed role-depth is, the better the approximation we obtain. Moreover, if the msc (respectively, the lcs) exists, then it will be found after some role-depth, thus in those cases our generalization algorithms yield the exact solution if k is guessed correctly. Therefore, our algorithm can be used as an ‘any-time’ approximation method that converges to the optimal solution.

Third, we present an empirical evaluation of our prototypical system Gel for computing the generalization inferences role-depth bounded msc and lcs. Despite our implementation being still in a very early stage, the experimental results on well-known bio-medical knowledge bases show promising performance of our algorithms. The extension of Gel to handle also probabilistic concepts presents some challenges, high among them is the reduction of the space used. As described in this paper, the completion algorithm for Prob-$\mathcal{ELOR}_c^{01}$ needs to store a set of completion sets for *each* probabilistic concept appearing in the knowledge base. While this does not affect the theoretical complexity of the algorithm (remains polynomial), it does make its application infeasible for knowledge bases having hundreds of thousands of concepts, even if only a handful of them are probabilistic. A direction of future research is to optimize the algorithm in a way that only a limited amount of the completion sets is generated, taking advantage of its pay-as-you-go behavior. The generalization algorithms itself on the other hand are conjectured to perform well even for DLs with subjective probability.

There exist several other non-standard inferences that have been studied for classical description logics and would be of interest in the context of subjective probabilities. One of them is the discovery of the precise axioms from a knowledge base that are responsible for a consequence to follow. This task is usually known as *axiom pinpointing* in the literature [48–50]. The generalization inferences msc and lcs are usually employed for the bottom-up creation of a knowledge base. Axiom pinpointing is useful then for debugging possible errors introduced during this modeling phase. Once the axioms responsible for this error have been identified, it is necessary to correct them in an adequate manner. The use of probabilities introduces a new challenge as seemingly innocuous axioms may interact to produce unexpected (and possibly unwanted) consequences. A further study of this problem will be a matter of future work.

Please cite this article in press as: A. Ecke et al., Completion-based generalization inferences for the Description Logic $\mathcal{ELOR}$ with subjective probabilities, International Journal of Approximate Reasoning (2014), http://dx.doi.org/10.1016/j.ijar.2014.03.001

Appendix A. Omitted proofs for Section 3

A.1. Proof for Proposition 6

To proof the correctness of the completion algorithm for $\mathcal{ELOR}$ as given in Proposition 6, we split it in two parts: soundness and completeness.

Lemma 16 *(Soundness of the $\mathcal{ELOR}$ completion algorithm). Let $\mathcal{T}$ be an $\mathcal{ELOR}$-TBox in normal form, $A, B \in \mathsf{BC}_\mathcal{T}$, $r \in \mathsf{Sig}(\mathcal{T}) \cap N_R$. Then, the following properties hold:*

$$B \in S^G(A) \implies G : A \sqsubseteq_\mathcal{T} B \tag{A.1}$$

$$B \in S^G(A, r) \implies G : A \sqsubseteq_\mathcal{T} \exists r.B \tag{A.2}$$

Proof. We proof soundness by induction on the number of rule applications. More precisely, we show that the properties (A.1) and (A.2) hold for the initial subsumer sets and are preserved by any rule application.

- Initially, $S^G(A) = \{A, \top\}$ and $S^G(A, r) = \emptyset$. Since $G : A \sqsubseteq_\mathcal{T} A$ and $G : A \sqsubseteq_\mathcal{T} \top$ always holds, (A.1) and (A.2) are satisfied.
- Assume that rule **OR1** has been applied to $A_1 \in S^G(A)$ and $A_1 \sqsubseteq B \in \mathcal{T}$. By induction, $A_1 \in S^G(A)$ implies $G : A \sqsubseteq_\mathcal{T} A_1$, which then yields $G : A \sqsubseteq_\mathcal{T} B$. Thus, after adding B to $S^G(A)$, (A.1) and (A.2) are still satisfied.
- Suppose that rule **OR2** has been applied to $A_1, A_2 \in S^G(A)$ and that $A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{T}$. By induction, $A_1 \in S^G(A)$ implies $G : A \sqsubseteq_\mathcal{T} A_1$ and similarly, $A_2 \in S^G(A)$ implies $G : A \sqsubseteq_\mathcal{T} A_2$. This yields $G : A \sqsubseteq_\mathcal{T} B$. Thus, after adding B to $S^G(A)$, (A.1) and (A.2) are still satisfied.
- Assume that rule **OR3** has been applied to $A_1 \in S^G(A)$ and $A_1 \sqsubseteq \exists r.B \in \mathcal{T}$. By induction, $A_1 \in S^G(A)$ implies $G : A \sqsubseteq_\mathcal{T} A_1$, which then yields $G : A \sqsubseteq_\mathcal{T} \exists r.B$. Thus, after adding B to $S^G(A, r)$, (A.1) and (A.2) are still satisfied.
- Assume that rule **OR4** has been applied to $B \in S^G(A, r)$, $B_1 \in S^G(B)$, and $\exists r.B_1 \sqsubseteq C \in \mathcal{T}$. By induction, $B \in S^G(A, r)$ implies $G : A \sqsubseteq_\mathcal{T} \exists r.B$ and similarly, $B_1 \in S^G(B)$ implies $G : B \sqsubseteq_\mathcal{T} B_1$. Hence $G : A \sqsubseteq_\mathcal{T} \exists r.B_1 \sqsubseteq_\mathcal{T} C$. Thus, after adding C to $S^G(A)$, (A.1) and (A.2) are still satisfied.
- Assume that rule **OR5** has been applied to $B \in S^G(A, r)$ and $r \sqsubseteq s \in \mathcal{T}$. By induction, $B \in S^G(A, r)$ implies $G : A \sqsubseteq_\mathcal{T} \exists r.B$, which then yields $G : A \sqsubseteq_\mathcal{T} \exists s.B$. Thus, after adding B to $S^G(A, s)$, (A.1) and (A.2) are still satisfied.
- Assume that rule **OR6** has been applied to $B \in S^G(A, r_1)$, $C \in S^G(B, r_2)$ and $r_1 \circ r_2 \sqsubseteq s \in \mathcal{T}$. By induction, $B \in S^G(A, r_1)$ implies $G : A \sqsubseteq_\mathcal{T} \exists r_1.B$ and similarly, $C \in S^G(B, r_2)$ implies $G : B \sqsubseteq_\mathcal{T} \exists r_2.C$, which then yield $G : A \sqsubseteq_\mathcal{T} \exists r_1.\exists r_2.C \sqsubseteq_\mathcal{T} \exists s.C$. Thus, after adding C to $S^G(A, s)$, (A.1) and (A.2) are still satisfied.
- Assume that rule **OR7** has been applied to $\{a\} \in S^G(A_1) \cap S^G(A_2)$ and $G \rightsquigarrow_R A_2$. By induction, $\{a\} \in S^G(A_i)$ implies $G : A_1 \sqsubseteq_\mathcal{T} \{a\}$ for $i = 1, 2$. If A_2 has a nonempty interpretation, then $A_2 : A_2 \equiv_\mathcal{T} \{a\}$ and this A_2 is reachable from G, also $G : A_2 \equiv_\mathcal{T} \{a\}$. This yields $G : A_1 \equiv_\mathcal{T} A_2$. Thus, after adding A_1 to $S^G(A_1)$, (A.1) and (A.2) are still satisfied. □

Lemma 17 *(Completeness of the $\mathcal{ELOR}$ completion algorithm). Let $\mathcal{T}$ be an $\mathcal{ELOR}$-TBox in normal form, $A, B \in \mathsf{BC}_\mathcal{T}$, $r \in \mathsf{Sig}(\mathcal{T}) \cap N_R$, and $G = A$ or $G \rightsquigarrow_R A$ if $A \in N_C$ and $G \in \mathsf{BC}_\mathcal{T}$ otherwise. Then, the following properties hold:*

- *If $A \sqsubseteq_\mathcal{T} B$, then $B \in S^G(A)$, and*
- *if $A \sqsubseteq_\mathcal{T} \exists r.B$, there exists $E \in \mathsf{BC}_\mathcal{T}$ s.t. $E \in S^G(A, r)$ and $B \in S^G(E)$.*

Proof. To show completeness, we assume that $B \notin S^G(A)$ (that there is no $E \in \mathsf{BC}_\mathcal{T}$ s.t. $E \in S^G(A, r)$ and $B \in S^G(E)$) and then construct a model $\mathcal{I}_G$ of $\mathcal{T}$ that shows $A \not\sqsubseteq_\mathcal{T} B$ ($A \not\sqsubseteq_\mathcal{T} \exists r.B$, respectively).

To construct the interpretation $\mathcal{I}_G$, we need to map each individual name to a single element of the domain. However, since do not make the unique name assumption, different individuals may be interpreted as the same element of the domain. We thus need to consider equivalence classes of (equivalent) individuals: $[a] = \{b \in \mathsf{Sig}(\mathcal{T}) \cap N_I \mid \{a\} \in S^G(\{b\})\}$. By rule **OR7** and the fact that nominals are always reachable from any G, these equivalence classes are well-defined: If $\{a\} \in S^G(\{b\})$, then $S^G(\{a\}) = S^G(\{b\})$. The domain of $\mathcal{I}_G$ will then contain all nominals modulo this equivalence and all concepts that are not subsumed by a nominal and can be reached from G or a nominal using the relation $\rightsquigarrow_R$. Thus, we can define $\mathcal{I}_G$ as follows:

$$\begin{aligned}
\Delta^{\mathcal{I}_G} &= \big\{[a] \,\big|\, a \in \mathsf{Sig}(\mathcal{T}) \cap N_I\big\} \cup \big\{A \in \mathsf{Sig}(\mathcal{T}) \cap N_C \mid G \rightsquigarrow_R A, \{a\} \notin S^G(A)\big\} \\
a^{\mathcal{I}_G} &= [a], \quad \text{for all } a \in N_I[\mathcal{T}] \\
A^{\mathcal{I}_G} &= \big\{x \,\big|\, A \in S^G(x)\big\} \\
r^{\mathcal{I}_G} &= \big\{\big(x, [a]\big) \in \Delta^{\mathcal{I}_G} \times \Delta^{\mathcal{I}_G} \,\big|\, \exists A\colon\ A \in S^G(x, r) \wedge \{a\} \in S^G(A)\big\} \cup \\
&\quad\ \big\{(x, A) \in \Delta^{\mathcal{I}_G} \times \Delta^{\mathcal{I}_G} \,\big|\, A \in S^G(x, r)\big\}
\end{aligned}$$

where x is either a nominal or a concept name from the domain $\Delta^{\mathcal{I}_G}$.

This interpretation $\mathcal{I}_G$ is indeed a model of $\mathcal{T}$, i.e., it satisfies all axioms in $\mathcal{T}$:

- Let $A \sqsubseteq B \in \mathcal{T}$ and $x \in A^{\mathcal{I}_G}$. By definition of $\mathcal{I}_G$, this implies $A \in S^G(x)$ and by rule **OR1** also $B \in S^G(x)$. But then we have $x \in B^{\mathcal{I}_G}$.
- Let $A_1 \sqcap A_2 \sqsubseteq B \in \mathcal{T}$ and $x \in (A_1 \sqcap A_2)^{\mathcal{I}_G}$, i.e., $x \in A_1^{\mathcal{I}_G}$ and $x \in A_2^{\mathcal{I}_G}$. By definition of $\mathcal{I}_G$, this implies $A_1, A_2 \in S^G(x)$ and by rule **OR1** also $B \in S^G(x)$. But then we have $x \in B^{\mathcal{I}_G}$.
- Let $A \sqsubseteq \exists r.B \in \mathcal{T}$ and $x \in A^{\mathcal{I}_G}$. By definition of $\mathcal{I}_G$, this implies $A \in S^G(x)$ and by rule **OR3** also $B \in S^G(x, r)$. Since $B \in S^G(B)$, we then have $B \in B^{\mathcal{I}_G}$ and $(x, B) \in r^{\mathcal{I}_G}$ and thus $x \in (\exists r.B)^{\mathcal{I}_G}$.
- Let $\exists r.A \sqsubseteq B \in \mathcal{T}$ and $x \in (\exists r.A)^{\mathcal{I}_G}$, i.e., there exists $x_1 \in \Delta^{\mathcal{I}_G}$ such that $(x, x_1) \in r^{\mathcal{I}_G}$ and $x_1 \in A^{\mathcal{I}_G}$, thus $A \in S^G(x_1)$ by definition of $\mathcal{I}_G$. There are two cases: If $x_1 \in \mathsf{Sig}(\mathcal{T}) \cap N_C$ with $G \rightsquigarrow_R x_1$, $\{a\} \notin S^G(x_1)$, this implies that $x_1 \in S^G(x, r)$, and thus by rule **OR4** we have $B \in S^G(x)$ and finally $x \in B^{\mathcal{I}_G}$. If $x_1 = [a]$ for an individual name $a \in \mathsf{Sig}(\mathcal{T}) \cap N_I$, then we have that there is $y \in \Delta^{\mathcal{I}_G}$ with $y \in S^G(x, r)$ and $\{a\} \in S^G(y)$ by the definition of $\mathcal{I}_G$. Then the completion algorithm will deduce $A \in S^G(y)$ the same way as it did for $A \in S^G(x_1)$, and thus **OR4** yields $B \in S^G(x)$ and hence $x \in B^{\mathcal{I}_G}$.
- Let $r \sqsubseteq s \in \mathcal{T}$ and $(x_1, x_2) \in r^{\mathcal{I}_G}$. There are two cases: If $x_1 \in \mathsf{Sig}(\mathcal{T}) \cap N_C$ with $G \rightsquigarrow_R x_1$, $\{a\} \notin S^G(x_1)$, then from the definition of $\mathcal{I}_G$ it follows that $x_2 \in S^G(x_1, r)$ and by rule **OR5** we have $x_2 \in S^G(x_1, s)$. But then $(x_1, x_2) \in s^{\mathcal{I}_G}$. If $x_2 = [a]$ for an individual name $a \in \mathsf{Sig}(\mathcal{T}) \cap N_I$, then we have that there is $y \in \Delta^{\mathcal{I}_G}$ with $y \in S^G(x_1, r)$ and $\{a\} \in S^G(y)$ by the definition of $\mathcal{I}_G$. Then **OR5** yields $y \in S^G(x_1, s)$ by together with $\{a\} \in S^G(y)$ we have $(x_1, x_2) \in s^{\mathcal{I}_G}$.
- Let $r_1 \circ r_2 \sqsubseteq s \in \mathcal{T}$ and $(x_1, x_3) \in (r_1 \circ r_2)^{\mathcal{I}_G}$, i.e., there exists $x_2 \in \Delta^{\mathcal{I}_G}$ with $(x_1, x_2) \in r_1^{\mathcal{I}_G}$ and $(x_2, x_3) \in r_2^{\mathcal{I}_G}$. Again, there are two cases:
  If $x_2 \in \mathsf{Sig}(\mathcal{T}) \cap N_C$ with $G \rightsquigarrow_R x_2$, $\{a\} \notin S^G(x_2)$, then the definition of $\mathcal{I}_G$ implies that $x_2 \in S^G(x_1, r_1)$. Then we know that, similar to the above case, there is a y with $y \in S^G(x_2, r_2)$ with $x_3 \in S^G(y)$ if $x_3 = [b]$ for an individual b or $y = x_3$ otherwise.
  If $x_2 = [a]$ for an individual name $a \in \mathsf{Sig}(\mathcal{T}) \cap N_I$, then there exists a $y_2 \in \Delta^{\mathcal{I}_G}$ with $y_1 \in S^G(x_1, r_1)$ and $\{a\} \in S^G(y_1)$. Again, similar to above, there is a y with $y \in S^G(x_2, r_2)$ with $x_3 \in S^G(y)$ if $x_3 = [b]$ for an individual b or $y = x_3$ otherwise. The completion algorithm will deduce $y \in S^G(y_1, r_2)$ the same way as it did for $y \in S^G(x_2, r_2)$.
  In both cases, rule **OR6** will yield $y \in S^G(x_1, s)$, and by definition of $\mathcal{I}_G$ we finally have $(x_1, x_3) \in s^{\mathcal{I}_G}$.

Since we assume that $B \notin S^G(A)$ (that there is no $E \in \mathsf{BC}_\mathcal{T}$ such that $E \in S^G(A, r)$ and $B \in S^G(E)$), it follows from the definition of $\mathcal{I}_G$ that $A \notin B^{\mathcal{I}_G}$ ($A \notin (\exists r.B)^{\mathcal{I}_G}$). Since we always have $A \in A^{\mathcal{I}_G}$, $\mathcal{I}_G$ shows that $A \not\sqsubseteq_\mathcal{T} B$ (respectively, $A \not\sqsubseteq_\mathcal{T} \exists r.B$), and thus completeness of the completion algorithm. □

Proposition 6 then follows directly from Lemmas 16 and 17.

A.2. Proof for Proposition 7

Again, we split the proof for the correctness of the k-lcs algorithm in two parts: first we show that k-lcs computes indeed a common subsumer of the input concepts, then we show that this common subsumer is the least one w.r.t. subsumption.

Lemma 18. *Let $\mathcal{T}$ be an $\mathcal{ELOR}$-TBox, $\mathcal{T}'$ be the TBox obtained from $\mathcal{T}$ by applying the normalization rules,* S *be the set of completion sets obtained from $\mathcal{T}'$, A, B be concept names, X, Y be basic concepts with $A \rightsquigarrow_R X$, $B \rightsquigarrow_R Y$, k be a natural number and* $L = \text{k-lcs-r}(X, Y, \mathsf{S}, k, A, B)$. *Then $X \sqsubseteq_{\mathcal{T}'} L$ and $Y \sqsubseteq_{\mathcal{T}'} L$.*

Proof. This lemma can be shown by induction on k for the recursive procedure k-lcs-r. For the case $k = 0$, the result

$$L = \bigsqcap_{E \in S^A(X) \cap S^B(Y) \cap \mathsf{BC}_\mathcal{T}} E$$

of k-lcs-r is a conjunction of basic concepts, but no existential restrictions. By soundness of the completion rules, we know that $E \in S^A(X) \cap S^B(Y)$ implies $X \sqsubseteq_{\mathcal{T}'} E$ and $Y \sqsubseteq_{\mathcal{T}'} E$. Since L contains exactly those conjuncts, we also have $X \sqsubseteq_{\mathcal{T}'} L$ and $Y \sqsubseteq_{\mathcal{T}'} L$.

For $k > 0$, L is a conjunction of concept names and existential restrictions $\exists r.E$. For the concept names, the same argument used for the case where $k = 0$ applies. For existential restrictions of the form $\exists r.\text{k-lcs-r}(E, F, \mathsf{S}, k - 1, A, B)$ with $(E, F) \in S^A(X, r) \times S^B(Y, r)$, $E \in S^A(X, r)$ implies $X \sqsubseteq_{\mathcal{T}'} \exists r.E$ by soundness of the completion algorithm, and similar $Y \sqsubseteq_{\mathcal{T}'} \exists r.F$. Then the induction hypothesis yields that for $L' = \text{k-lcs-r}(E, F, \mathsf{S}, k - 1, A, B)$ we have $E \sqsubseteq_{\mathcal{T}'} L'$ and $F \sqsubseteq_{\mathcal{T}'} L'$ and thus also $X \sqsubseteq_{\mathcal{T}'} \exists r.L'$ and $Y \sqsubseteq_{\mathcal{T}'} \exists r.L'$. All together, this means $X \sqsubseteq_{\mathcal{T}'} L$ and $Y \sqsubseteq_{\mathcal{T}'} L$ is analog. □

Lemma 19. *Let $\mathcal{T}$ be an $\mathcal{ELOR}$-TBox, $\mathcal{T}'$ be the TBox obtained from $\mathcal{T}$ by applying the normalization rules,* S *be the set of completion sets obtained from $\mathcal{T}'$, A, B be concept names, X, Y be basic concepts with $A \rightsquigarrow_R X$, $B \rightsquigarrow_R Y$, k be a natural number and* $L = \text{k-lcs-r}(X, Y, \mathsf{S}, k, A, B)$. *Then for each $\mathcal{ELOR}$-concept F with* $\mathsf{Sig}(\mathrm{F}) \subseteq \mathsf{Sig}(\mathcal{T})$ *and $rd(F) \leqslant k$, $X \sqsubseteq_{\mathcal{T}'} F$ and $Y \sqsubseteq_{\mathcal{T}'} F$ imply $L \sqsubseteq_{\mathcal{T}'} F$.*

Proof. By induction on the role-depth $rd(F)$. Let $rd(F) = 0$, i.e. $F = \bigsqcap_{i \in I} E_i$ contains no existential restrictions but only basic concepts E_i. Since $X \sqsubseteq_{\mathcal{T}'} F$ and $Y \sqsubseteq_{\mathcal{T}'} F$, we also have $X \sqsubseteq_{\mathcal{T}'} E_i$ and $Y \sqsubseteq_{\mathcal{T}'} E_i$ for all conjuncts E_i of F. Then, completeness of the completion algorithm yields that $E_i \in S^X(X)$ and since $A \rightsquigarrow_R X$ also $E \in S^A(X)$. Similarly, we have $E_i \in S^B(Y)$ for all conjuncts E_i of F and thus

$$L = \bigsqcap_{E \in S^A(X) \cap S^B(Y) \cap \mathsf{BC}_{\mathcal{T}}} E \sqsubseteq_{\mathcal{T}'} F.$$

If $rd(F) > 0$, F may contain two kinds of conjuncts: basic concepts and existential restrictions. The basic concepts in F must appear in L as well by an argument analog to the case $rd(F) = 0$. Let $\exists r.F'$ be a top-level conjunct of F. Since $X \sqsubseteq_{\mathcal{T}'} F$ and $Y \sqsubseteq_{\mathcal{T}'} F$, completeness yields that there exists an $E \in S^A(X, r)$ such that $F' \in S^A(E)$ (i.e. $E \sqsubseteq_{\mathcal{T}'} F'$), and an $E' \in S^B(Y, r)$ such that $F' \in S^B(E')$ (i.e. $E' \sqsubseteq_{\mathcal{T}'} F'$). By induction hypothesis, it follows that $\mathsf{k\text{-}lcs\text{-}r}(E, E', \mathsf{S}, k-1, A, B) \sqsubseteq_{\mathcal{T}'} F'$, and thus it also holds that $L \sqsubseteq_{\mathcal{T}'} \exists r.\mathsf{k\text{-}lcs\text{-}r}(E, E', \mathsf{S}, k-1, A, B) \sqsubseteq_{\mathcal{T}'} \exists r.F'$. All together, we get $L \sqsubseteq_{\mathcal{T}'} F$. □

Proposition 7 follows directly from Lemmas 18 and 19.

A.3. Proof for Proposition 8

As for the k-lcs, we split the proof for the correctness of the k-msc algorithm in two parts: first we show that k-msc computes indeed a concept that has the given individual as instance, then we show that this concept is the least one w.r.t. subsumption.

Lemma 20. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ELOR}$-KB, $\mathcal{T}'$ be the TBox obtained from $\mathcal{T}$ by absorbing $\mathcal{A}$ and applying the normalization rules,* S *be the set of completion sets obtained from $\mathcal{T}'$, X be a basic concept with $\top \rightsquigarrow_R X$, k be a natural number and $L = \mathsf{traversal\text{-}concept}(X, \mathsf{S}, k)$. Then $X \sqsubseteq_{\mathcal{T}'} L$.*

Proof. By induction on k. For the case $k = 0$, the result $L = \bigsqcap_{i \in I} E_i$ of traversal-concept is a conjunction of concepts names $E_i \in S^\top(X) \cap \mathsf{BC}_{\mathcal{K}}$, but no existential restrictions. By soundness of the completion rules, we know that $E_i \in S^\top(X)$ implies $X \sqsubseteq_{\mathcal{T}'} E$. Since L contains exactly those conjuncts, we also have $X \sqsubseteq_{\mathcal{T}'} L$.

For the case $k > 0$, L is a conjunction of concept names and existential restrictions $\exists r.E$. For the concept names, the same argument as for the case $k = 0$ applies. For existential restrictions of the form $\exists r.\mathsf{traversal\text{-}concept}(E, \mathsf{S}, k-1)$ with $E \in S^\top(X, r)$, soundness yields $X \sqsubseteq_{\mathcal{T}'} \exists r.E$. Then the induction hypothesis yields that for $L' = \mathsf{traversal\text{-}concept}(E, \mathsf{S}, k-1)$ we have $E \sqsubseteq_{\mathcal{T}'} L'$ and thus also $X \sqsubseteq_{\mathcal{T}'} \exists r.L'$. All together, this means $X \sqsubseteq_{\mathcal{T}'} L$. □

Lemma 21. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ELOR}$-KB, $\mathcal{T}'$ be the TBox obtained from $\mathcal{T}$ by absorbing $\mathcal{A}$ and applying the normalization rules,* S *be the set of completion sets obtained from $\mathcal{T}'$, X be a basic concept with $\top \rightsquigarrow_R X$, k be a natural number and $L = \mathsf{traversal\text{-}concept}(X, \mathsf{S}, k)$. Then, for every $\mathcal{ELR}$-concept F with $\mathsf{Sig}(F) \subseteq \mathsf{Sig}(\mathcal{K})$ and $rd(F) \leqslant k$, $X \sqsubseteq_{\mathcal{T}'} F$ implies $L \sqsubseteq_{\mathcal{T}'} F$.*

Proof. By induction on the role-depth $rd(F)$. Let $rd(F) = 0$, i.e. $F = \bigsqcap_{i \in I} E_i$ contains no existential restrictions. Since $X \sqsubseteq_{\mathcal{T}'} F$, we also have $X \sqsubseteq_{\mathcal{T}'} E_i$ for all conjuncts E_i of F. Then, completeness of the completion algorithm yields that $E_i \in S^X(X)$ and since $\top \rightsquigarrow_R X$ also $E_i \in S^\top(X)$. Therefore

$$L = \bigsqcap_{E \in S^\top(X) \cap (\mathsf{Sig}(\mathcal{K}) \cap N_C \cup \{\top\})} E \sqsubseteq_{\mathcal{T}'} F.$$

If $rd(F) > 0$, F may contain two kinds of conjuncts: concepts names and existential restrictions. The concepts names in F must appear in L as well by the same argument as in case $rd(F) = 0$. Let $\exists r.F'$ be a top-level conjunct of F. Since $X \sqsubseteq_{\mathcal{T}'} F \sqsubseteq_{\mathcal{T}'} \exists r.F'$, completeness yields that there exists an $E \in S^\top(X, r)$ such that $F' \in S^\top(E)$, i.e. $E \sqsubseteq_{\mathcal{T}'} F'$. Since $rd(F') < rd(F)$, the induction hypothesis yields $\mathsf{traversal\text{-}concept}(E, \mathsf{S}, k-1) \sqsubseteq_{\mathcal{T}'} F'$, and thus also $L \sqsubseteq_{\mathcal{T}'} \exists r.\mathsf{traversal\text{-}concept}(E, \mathsf{S}, k-1) \sqsubseteq_{\mathcal{T}'} \exists r.F'$. Everything together we get $L \sqsubseteq_{\mathcal{T}'} F$. □

Remark. In the previous lemmas, if X is a nominal $\{a\}$, then $\{a\} \sqsubseteq_{\mathcal{T}'} L$ is equivalent to $\mathcal{T}' \models L(a)$.

Proposition 8 follows directly from Lemmas 20 and 21.

Appendix B. Omitted proofs for Section 4

B.1. Missing rules for Prob-$\mathcal{ELOR}_c^{01}$

The following rules can to be added to the set of completion rules described for Prob-$\mathcal{ELO}_c^{01}$ in Fig. 5 on page 14 to handle the additional constructors of Prob-$\mathcal{ELOR}_c^{01}$:

PR12 If $B \in S_*^G(X, r, v)$, $r \sqsubseteq s \in \mathcal{T}$ and $B \notin S_*^G(X, s, v)$, then $S_*^G(X, s, v) := S_*^G(X, s, v) \cup \{B\}$

PR13 If $B \in S_*^G(X, r_1, v)$, $C \in S_{\gamma(v)}^G(B, r_2, \gamma(v))$, $r_1 \circ r_2 \sqsubseteq s \in \mathcal{T}$ and $C \notin S_*^G(X, s, v)$, then $S_*^G(X, s, v) := S_*^G(X, s, v) \cup \{C\}$

These rules are simple adaptations of the rules **OR5** and **OR6** from the classification algorithm for crisp $\mathcal{ELOR}$ to the probabilistic variant.

B.2. Proof for Proposition 12

Lemma 22. *The completion algorithm is sound, i.e.*

$$C \in S_*^G(X, v) \quad \textit{implies} \quad G : P_* X \sqsubseteq_{\mathcal{T}} P_v C, \tag{B.1}$$

$$C \in S_*^G(X, r, v) \quad \textit{implies} \quad G : P_* X \sqsubseteq_{\mathcal{T}} P_v \exists r.C. \tag{B.2}$$

Proof. We show this by induction on the number of rule applications. It is easy to see that the initial subsumer sets satisfy (B.1) and (B.2). Also, after each rule application (B.1) and (B.2) will still be satisfied. This proof is similar to the soundness proof in [1]: even though the completion rules there look very different from ours, there is a direct correspondence. Thus, we will only show (B.1) for those completion rules, that are not in the original completion algorithm in [1]. For property (B.2), note that none of these new rules changes the subsumer sets $S_*^G(X, r, v)$.

PR7 If $P_{=1}A \in S_*^G(X, 0)$, then by induction hypothesis $G : P_* X \sqsubseteq_{\mathcal{T}} P_{=1}A$. Then the implication $A \in S_*^G(X, 1) \Longrightarrow G : P_* X \sqsubseteq_{\mathcal{T}} P_{=1}A$ is obviously correct, and we can add A to $S_*^G(X, 1)$.

PR11 If $\{a\} \in S_*^G(X, v)$, then by induction hypothesis $G : P_* X \sqsubseteq_{\mathcal{T}} P_v\{a\}$, i.e. for all models $\mathcal{I}$ of $\mathcal{T}$ and all worlds $w \in W$ we have: if G is not empty, then $(P_* X)^{\mathcal{I},w} \subseteq (P_v\{a\})^{\mathcal{I},w}$. Together with

$$\begin{aligned}\left(P_{>0}\{a\}\right)^{\mathcal{I},w} &= \left\{d \mid \exists v \in W\colon\ \mu(v) > 0 \wedge d \in \{a\}^{\mathcal{I},v}\right\}\\ &= \left\{d \mid \exists v \in W\colon\ \mu(v) > 0 \wedge d = a^{\mathcal{I}}\right\} = \left\{a^{\mathcal{I}}\right\} = \{a\}^{\mathcal{I},w}\end{aligned}$$

and similarly $(P_{=1}\{a\})^{\mathcal{I},w} = \{a\}^{\mathcal{I},w}$, this yields: if G is not empty, then $(P_* X)^{\mathcal{I},w} \subseteq \{a\}^{\mathcal{I},w} = \{a^{\mathcal{I}}\} = \{a\}^{\mathcal{I},v'}$ for all $v' \in W$, and hence it holds that $G : P_* X \sqsubseteq_{\mathcal{T}} P_{v'}\{a\}$. This means that the addition of $\{a\}$ to $S_*^G(X, v')$ still satisfies (B.1).

PR5 If $\{a\} \in S_{*_1}^G(X, *_1) \cap S_{*_2}^G(D, *_2)$, then by induction hypothesis we have that $G : P_{*_1} X \sqsubseteq_{\mathcal{T}} \{a\}$ and $G : P_{*_2} D \sqsubseteq_{\mathcal{T}} \{a\}$. Additionally, by definition of $\rightsquigarrow_R$, $G \overset{*_2}{\rightsquigarrow}_R D$ implies that if G is not empty, then $P_{*_2} D$ must be not empty as well. Thus we have $G : P_{*_2} D \equiv_{\mathcal{T}} \{a\}$. This implies that $G : P_{*_1} X \sqsubseteq P_{*_2} D$ and hence the addition of $P_{*_2} D$ to $S_{*_1}^G(X, *_1)$ still satisfies (B.1). □

Lemma 23. *The completion algorithm is complete, i.e. for a normalized TBox $\mathcal{T}$, a concept name G, a basic concept B that occurs in $\mathcal{T}$, and a role name r we have*

$$G \sqsubseteq_{\mathcal{T}} B \quad \textit{implies} \quad B \in S_0^G(G, 0)$$

$$G \sqsubseteq_{\mathcal{T}} \exists r.B \quad \textit{implies} \quad \exists A \textit{ with } A \in S_0^G(G, r, 0) \textit{ and } B \in S_0^G(A, 0)$$

Proof. We assume that $B \notin S_0^G(G, 0)$ (resp. there is no A with $A \in S_0^G(G, r, 0)$ and $B \in S_0^G(A, 0)$) and construct a model $\mathcal{I}_G$ of $\mathcal{T}$ which shows that $G \not\sqsubseteq_{\mathcal{T}} B$ (resp. $G \not\sqsubseteq_{\mathcal{T}} \exists r.B$). To construct this model, we need equivalence classes of nominals: $[a] = \{b \in N_I[\mathcal{T}] \mid \{a\} \in S_0^G(\{b\}, 0)\}$. By rules **PR5** and **PR11** and the fact that nominals are always reachable from any G, these equivalence classes are well-defined: If $\{a\} \in S_0^G(\{b\}, 0)$, then $S_0^G(\{a\}, v) = S_0^G(\{b\}, w)$ for all $v, w \in V$. The domain of the interpretation will contain all nominals (modulo equivalence) and for each world $w \in V$ all concepts that are not subsumed by a nominal and can be reached from G or a nominal using the relation $\rightsquigarrow_R$:

Let $\mathcal{I}_G = (\Delta^{\mathcal{I}_G}, W, (\mathcal{I}_{G,w})_{w \in W}, \mu)$ be the following interpretation:

$$\begin{aligned}\Delta^{\mathcal{I}_G} &:= \left\{[a] \mid a \in N_I[\mathcal{T}]\right\} \cup\\ &\quad \left\{(A, v) \in N_C[\mathcal{T}] \times V \mid G \overset{\gamma(v)}{\rightsquigarrow}_R A,\ \{a\} \notin S_{\gamma(v)}^G\big(A, \gamma(v)\big)\right\}\\ W &:= V\\ \mu(0) &:= 0\\ \mu(w) &:= \frac{1}{|W \setminus \{0\}|} \quad \text{for all } w \in W \setminus \{0\}\\ a^{\mathcal{I}_G} &= [a] \quad \text{for all } a \in N_I[\mathcal{T}]\end{aligned}$$

To interpret concept and role names, we need a bijection $\pi_v(w): W \to W$ for each $v \in W \setminus \{0\}$ with $\pi_v(v) = \varepsilon$ and $\pi_v(0) = 0$. Moreover, π_0 is the identity mapping on W.

Then

$$\begin{aligned} A^{\mathcal{I}_G,w} &= \big\{[a] \mid A \in S_0^G(\{a\}, w)\big\} \cup \\ &\quad \big\{(B, v) \in \Delta^{\mathcal{I}_G} \mid A \in S_{\gamma(v)}^G\big(B, \pi_v(w)\big)\big\} \\ r^{\mathcal{I}_G,w} &= \big\{\big([a],[b]\big) \in \Delta^{\mathcal{I}_G} \times \Delta^{\mathcal{I}_G} \mid \exists A\colon\ A \in S_0^G(\{a\}, r, w),\ \{b\} \in S_{\gamma(w)}^G\big(A, \gamma(w)\big)\big\} \cup \\ &\quad \big\{\big([a],(A,w)\big) \in \Delta^{\mathcal{I}_G} \times \Delta^{\mathcal{I}_G} \mid A \in S_0^G\big(\{a\}, r, w\big)\big\} \cup \\ &\quad \big\{\big((B,v),[b]\big) \in \Delta^{\mathcal{I}_G} \times \Delta^{\mathcal{I}_G} \mid \exists A\colon\ A \in S_{\gamma(v)}^G\big(B, r, \pi_v(w)\big),\ \{b\} \in S_{\gamma(w)}^G\big(A, \gamma(w)\big)\big\} \cup \\ &\quad \big\{\big((B,v),(A,w)\big) \in \Delta^{\mathcal{I}_G} \times \Delta^{\mathcal{I}_G} \mid A \in S_{\gamma(v)}^G\big(B, r, \pi_v(w)\big)\big\} \end{aligned}$$

Before proving that $\mathcal{I}_G$ is indeed a model of $\mathcal{T}$, we generalize the definition of $A^{\mathcal{I}_G,w}$ to probabilistic concepts:

$$X \in S_{\gamma(v)}^G\big(B, \pi_v(w)\big) \text{ iff } (B, v) \in X^{\mathcal{I}_G,w} \quad \text{for } X \in \mathsf{BC}_{\mathcal{T}},\ (B, v) \in \Delta^{\mathcal{I}_G} \tag{B.3}$$

$$X \in S_0^G\big(\{a\}, 0\big) \text{ iff } [a] \in X^{\mathcal{I}_G,w} \quad \text{for } X \in \mathsf{BC}_{\mathcal{T}},\ [a] \in \Delta^{\mathcal{I}_G} \tag{B.4}$$

To show this, we make a case distinction on the forms of X. First notice that in (B.3) X cannot be a nominal, since otherwise B would be subsumed by one and hence not be in the domain $\Delta^{\mathcal{I}_G}$ as we assumed.

- If $X = \top$, then (B.3) and (B.4) are true by definition of $\top^{\mathcal{I}_G,w} = \Delta^{\mathcal{I}_G}$ and the fact that $\top$ is in each subsumer set.
- If $X = A \in N_C$, then (B.3) and (B.4) are true by definition of $A^{\mathcal{I}_G,w}$.
- If $X = P_{>0}A$. For the "⇒" direction, let $P_{>0}A \in S_{\gamma(v)}^G(B, \pi_v(w))$. By rule **PR6**, we have $A \in S_{\gamma(v)}^G(B, P_{>0}A)$ and by definition of $\mathcal{I}_G$ $(B, v) \in A^{\mathcal{I}_G,u}$ with $\pi_v(u) = P_{>0}A$. By definition of π_v and $\mathcal{I}_G$, $\mu(u) > 0$ and thus $(B, v) \in (P_{>0}A)^{\mathcal{I}_G,w}$. For the "⇐" direction, let $(B, v) \in (P_{>0}A)^{\mathcal{I}_G,w}$, i.e. there is $u \in W \setminus \{0\}$ with $(B, v) \in A^{\mathcal{I}_G,u}$. The definition of $\mathcal{I}_G$ yields $A \in S_{\gamma(v)}^G(B, \pi_v(u))$ with $\pi_v(u) \neq 0$ by definition of π_v. Then by rule **PR9** $P_{>0}A \in S_{\gamma(v)}^G(B, \pi_v(w))$.
- If $X = P_{=1}A$. For the "⇒" direction, let $P_{=1}A \in S_{\gamma(v)}^G(B, \pi_v(w))$. By rules **PR7** and **PR10** we have $P_{=1}A \in S_{\gamma(v)}^G(B, u)$ for all $u \in W$ and by rule **PR8** $A \in S_{\gamma(v)}^G(B, u)$ for all $u \in W \setminus \{0\}$. Since π_v is a bijection on W with $\pi_v(0) = 0$, this also means $A \in S_{\gamma(v)}^G(B, \pi_v(u'))$ for all $u' \in W \setminus \{0\}$ and hence by definition of $\mathcal{I}_G$, $(B, v) \in A^{\mathcal{I}_G,u'}$ for all $u' \in W \setminus \{0\}$. Finally, the definition of μ then yields $(B, v) \in (P_{=1}A)^{\mathcal{I}_G,w}$. For the "⇐" direction, let $(B, v) \in (P_{=1}A)^{\mathcal{I}_G,w}$, i.e. for all $u \in W \setminus \{0\}$ we have $(B, v) \in A^{\mathcal{I}_G,u}$, especially for u' with $\pi_v(u') = 1$. Then, the definition of $\mathcal{I}_G$ yields $A \in S_{\gamma(v)}^G(B, \pi_v(u') = 1)$ and by rule **PR10** $P_{=1}A \in S_{\gamma(v)}^G(B, w')$ for all $w' \in W$, especially $P_{=1}A \in S_{\gamma(v)}^G(B, \pi_v(w))$.

This interpretation $\mathcal{I}_G$ is indeed a model of $\mathcal{T}$, which we will show using a case distinction on the types of GCIs in $\mathcal{T}$.

- $C \sqsubseteq D \in \mathcal{T}$. Let $(B, v) \in C^{\mathcal{I}_G,w}$, then (B.3) yields $C \in S_{\gamma(v)}^G(B, \pi_v(w))$ and by rule **PR1** also $D \in S_{\gamma(v)}^G(B, \pi_v(w))$. (B.3) yields $(B, v) \in D^{\mathcal{I}_G,w}$.
  Let $[a] \in C^{\mathcal{I}_G,w}$, then $C \in S_0^G(\{a\}, 0)$ by (B.4) and by rule **PR1** also $D \in S_0^G(\{a\}, 0)$. (B.4) then yields $[a] \in D^{\mathcal{I}_G,w}$.
- $C_1 \sqcap C_2 \sqsubseteq D \in \mathcal{T}$. Let $(B, v) \in (C_1 \sqcap C_2)^{\mathcal{I}_G,w}$, i.e. by the semantics of conjunction $(B, v) \in C_1^{\mathcal{I}_G,w}$ and $(B, v) \in C_2^{\mathcal{I}_G,w}$. Then (B.3) yields that $C_1, C_2 \in S_{\gamma(v)}^G(B, \pi_v(w))$, and by rule **PR2** $D \in S_{\gamma(v)}^G(B, \pi_v(w))$. (B.3) then yields $(B, v) \in D^{\mathcal{I}_G,w}$.
  Let $[a] \in (C_1 \sqcap C_2)^{\mathcal{I}_G,w}$, i.e. $[a] \in C_1^{\mathcal{I}_G,w}$ and $[a] \in C_2^{\mathcal{I}_G,w}$. Then we have $C_1, C_2 \in S_0^G(\{a\}, 0)$ by (B.4) and by rule **PR2** also $D \in S_0^G(\{a\}, 0)$. (B.4) then yields $[a] \in D^{\mathcal{I}_G,w}$.
- $C \sqsubseteq \exists r.A$. Let $(B, v) \in C^{\mathcal{I}_G,w}$, then (B.3) yields $C \in S_{\gamma(v)}^G(B, \pi_v(w))$ and by rule **PR3** $A \in S_{\gamma(v)}^G(B, r, \pi_v(w))$. Then, there are two cases: If $(A, w) \in \Delta^{\mathcal{I}_G}$, i.e. there is no nominal $\{b\} \in S_{\gamma(w)}^G(A, \gamma(w))$, then the definition of $r^{\mathcal{I}_G,w}$ yields $((B, v), (A, w)) \in r^{\mathcal{I}_G,w}$. By the initialization of the completion sets we also have $A \in S_{\gamma(w)}^G(A, \pi_w(w))$ as $\gamma(w) = \pi_w(w)$ by definition, and thus $(A, w) \in A^{\mathcal{I}_G,w}$. Since $((B, v), (A, w)) \in r^{\mathcal{I}_G,w}$, this yields $(B, v) \in (\exists r.A)^{\mathcal{I}_G,w}$.
  If $(A, w) \notin \Delta^{\mathcal{I}_G}$, then there is a nominal $\{b\} \in S_{\gamma(w)}^G(A, \gamma(w))$ and the definition of $r^{\mathcal{I}_G,w}$ yields $((B, v), [b]) \in r^{\mathcal{I}_G,w}$. On the other hand, rule **PR11** with $\{b\} \in S_{\gamma(w)}^G(A, \gamma(w))$ yields also $\{b\} \in S_{\gamma(w)}^G(A, 0)$ and then rule **PR5** with $\{b\} \in S_0^G(\{b\}, 0) \cap S_{\gamma(w)}^G(A, 0)$ and $G \overset{\gamma(w)}{\rightsquigarrow}_R A$ also implies $P_{\gamma(w)}A \in S_0^G(\{b\}, 0)$ and thus $A \in S_0^G(\{b\}, w)$, i.e. $[b] \in A^{\mathcal{I}_G,w}$.
  Similarly, let $[a] \in C^{\mathcal{I}_G,w}$, then (B.4) yields $C \in S_0^G(\{a\}, 0)$. By rule **PR3** $A \in S_0^G(\{a\}, r, 0)$. Again, we have the two cases as before, which can be shown analogously.
- $\exists r.A \sqsubseteq D$. Let $(B, v) \in (\exists r.A)^{\mathcal{I}_G,w}$, i.e. there is an $\alpha \in \Delta^{\mathcal{I}_G,w}$ with $((B, v), \alpha) \in r^{\mathcal{I}_G,w}$ and $\alpha \in A^{\mathcal{I}_G,w}$. By definition of $\mathcal{I}_G$, there are two cases: If $\alpha = (C, w) \in \Delta^{\mathcal{I}_G}$, then the definitions of $A^{\mathcal{I}_G,w}$ and $r^{\mathcal{I}_G,w}$ yield $A \in S_{\gamma(w)}^G(C, \pi_w(w))$ and $C \in$

$S^G_{\gamma(v)}(B, r, \pi_v(w))$. Since $\pi_w(w) = \gamma(w)$, and $\gamma(\pi_v(w)) = \gamma(w)$ for all $v \in V$, by rule **PR4** we get $D \in S^G_{\gamma(v)}(B, \pi_v(w))$ and thus by (B.3) $(B, v) \in D^{\mathcal{I}_G, w}$.

If $\alpha = [b] \in \Delta^{\mathcal{I}_G}$, then by the definitions of $A^{\mathcal{I}_G, w}$ and $r^{\mathcal{I}_G, w}$ we have $A \in S^G_0(\{b\}, w)$ and there exists C such that $C \in S^G_{\gamma(v)}(B, r, \pi_v(w))$ and $\{b\} \in S^G_{\gamma(w)}(C, \gamma(w))$. Because of $\{b\} \in S^G_{\gamma(w)}(C, \gamma(w))$, we have $S^G_{\gamma(w)}(C, \gamma(w)) \supseteq S^G_0(\{b\}, w)$ (which can be shown by induction on the number of rule applications to the latter), and hence $A \in S^G_{\gamma(w)}(C, \gamma(w))$. Together with $C \in S^G_{\gamma(v)}(B, r, \pi_v(w))$ and the facts $\pi_w(w) = \gamma(w)$, and $\gamma(\pi_v(w)) = \gamma(w)$ for all $v \in V$, rule **PR4** finally yields $D \in S^G_{\gamma(v)}(B, \pi_v(w))$ and thus by (B.3) $(B, v) \in D^{\mathcal{I}_G, w}$.

Similarly, let $[a] \in (\exists r.A)^{\mathcal{I}_G, w}$, i.e. there is an $\alpha \in \Delta^{\mathcal{I}_G, w}$ such that $([a], \alpha) \in r^{\mathcal{I}_G, w}$ and $\alpha \in A^{\mathcal{I}_G, w}$. Again, we have the two cases as before, which can be shown analogously.

Finally, by the assumption $B \notin S^G_0(G, 0)$ and the definition of $\mathcal{I}_G$ we have $(G, 0) \notin B^{\mathcal{I}_G, 0}$, whereas $G \in S^G_0(G, 0)$ yields $(G, 0) \in G^{\mathcal{I}_G, 0}$. Since $\mathcal{I}_G$ is a model of $\mathcal{T}$, this proves $G \not\sqsubseteq_{\mathcal{T}} B$.

The second case is similar. Assume that there is no A with $A \in S^G_0(G, r, 0)$ and $B \in S^G_0(A, 0)$, then by definition of the interpretation $\mathcal{I}_G$, there is no element $\alpha \in \Delta^{\mathcal{I}_G}$ with $((G, 0), \alpha) \in r^{\mathcal{I}_G, 0}$ and $\alpha \in B^{\mathcal{I}_G, 0}$. Since $\mathcal{I}_G$ is a model of $\mathcal{T}$, this shows that $G \not\sqsubseteq_{\mathcal{T}} \exists r.B$. □

B.3. Proof for Theorem 13

Again, we divide the proof for the correctness of the k-lcs algorithm for Prob-$\mathcal{ELO}^{01}_c$ in two parts: first we show that k-lcs computes indeed a common subsumer of the input concepts, then we show that this common subsumer is the least one w.r.t. subsumption.

Lemma 24. *Let $\mathcal{T}$ be a Prob-$\mathcal{ELO}^{01}_c$-TBox, $\mathcal{T}'$ be the TBox obtained from $\mathcal{T}$ by applying the normalization rules,* S *be the set of completion sets obtained from $\mathcal{T}'$, A, B be concept names, X, Y be basic concepts with $A \rightsquigarrow_R X$, $B \rightsquigarrow_R Y$, k be a natural number and $L = \text{k-lcs-r}(X, Y, \mathsf{S}, k, A, B)$. Then $X \sqsubseteq_{\mathcal{T}'} L$ and $Y \sqsubseteq_{\mathcal{T}'} L$.*

Proof. Similar to the crisp case, this lemma can be shown by induction on k for the recursive procedure k-lcs-r. For the case $k = 0$, the result

$$L = \bigsqcap_{E \in S^A_0(X,0) \cap S^B_0(Y,0) \cap \mathsf{BC}_{\mathcal{T}}} E$$

of k-lcs-r is a conjunction of (possibly probabilistic) concept names, but no existential restrictions. By soundness of the completion rules, we know that $E \in S^A_0(X, 0) \cap S^B_0(Y, 0)$ implies $X \sqsubseteq_{\mathcal{T}'} E$ and $Y \sqsubseteq_{\mathcal{T}'} E$. Since L contains exactly those conjuncts, we also have $X \sqsubseteq_{\mathcal{T}'} L$ and $Y \sqsubseteq_{\mathcal{T}'} L$.

For the case $k > 0$, L is a conjunction of concept names (possibly probabilistic) and existential restrictions $\exists r.E$, $P_{=1}\exists r.E$, and $P_{>0}\exists r.E$. For concept names, the same argument as for the case $k = 0$ applies. For existential restrictions $\exists r.\text{k-lcs-r}(E, F, \mathsf{S}, k-1, A, B)$ with $(E, F) \in S^A_0(X, r, 0) \times S^B_0(Y, r, 0)$, $E \in S^A_0(X, r, 0)$ implies $X \sqsubseteq_{\mathcal{T}'} \exists r.E$ by soundness of the completion algorithm, and similar $Y \sqsubseteq_{\mathcal{T}'} \exists r.F$. Then the induction hypothesis yields that for $L' = \text{k-lcs-r}(E, F, \mathsf{S}, k-1, A, B)$ we have $E \sqsubseteq_{\mathcal{T}'} L'$ and $F \sqsubseteq_{\mathcal{T}'} L'$ and thus also $X \sqsubseteq_{\mathcal{T}'} \exists r.L'$ and $Y \sqsubseteq_{\mathcal{T}'} \exists r.L'$.

Similarly, by soundness $E \in S^A_0(X, r, 1)$ implies $X \sqsubseteq_{\mathcal{T}'} P_{=1}\exists r.E$ and also $E \in \mathsf{PR}^A(X, r)$ implies $X \sqsubseteq_{\mathcal{T}'} P_{>0}\exists r.E$, respectively. By induction hypothesis $E \sqsubseteq_{\mathcal{T}'} \text{k-lcs-r}(E, F, \mathsf{S}, k-1, A, B)$, thus $X \sqsubseteq_{\mathcal{T}'} P_{=1}\exists r.\text{k-lcs-r}(E, F, \mathsf{S}, k-1, A, B)$ and $X \sqsubseteq_{\mathcal{T}'} P_{>0}\exists r.\text{k-lcs-r}(E, F, \mathsf{S}, k-1, A, B)$, respectively. All together, this means $X \sqsubseteq_{\mathcal{T}'} L$. The case for $Y \sqsubseteq_{\mathcal{T}'} L$ is analogous. □

Lemma 25. *Let $\mathcal{T}$ be a Prob-$\mathcal{ELO}^{01}_c$-TBox, $\mathcal{T}'$ be the TBox obtained from $\mathcal{T}$ by applying the normalization rules,* S *be the set of completion sets obtained from $\mathcal{T}'$, A, B be concept names, X, Y be basic concepts with $A \rightsquigarrow_R X$, $B \rightsquigarrow_R Y$, k be a natural number and $L = \text{k-lcs-r}(X, Y, \mathsf{S}, k, A, B)$. Then for each Prob-$\mathcal{ELO}^{01}_c$-concept F with $\mathsf{Sig}(F) \subseteq \mathsf{Sig}(\mathcal{T})$ and $rd(F) \leqslant k$, $X \sqsubseteq_{\mathcal{T}'} F$ and $Y \sqsubseteq_{\mathcal{T}'} F$ imply $L \sqsubseteq_{\mathcal{T}'} F$.*

Proof. By induction on the role-depth $rd(F)$. Let $rd(F) = 0$, i.e. $F = \bigsqcap E$ contains no existential restrictions. Since $X \sqsubseteq_{\mathcal{T}'} F$ and $Y \sqsubseteq_{\mathcal{T}'} F$, we also have $X \sqsubseteq_{\mathcal{T}'} E$ and $Y \sqsubseteq_{\mathcal{T}'} E$ for all conjuncts E of F. Then, completeness of the completion algorithm yields that $E \in S^X_0(X, 0)$ and since $A \rightsquigarrow_R X$ also $E \in S^A_0(X, 0)$. Similarly, we have $E \in S^B_0(Y, 0)$ for all conjuncts E of F and thus

$$L = \bigsqcap_{E \in S^A_0(X,0) \cap S^B_0(Y,0) \cap \mathsf{BC}_{\mathcal{T}}} E \sqsubseteq_{\mathcal{T}'} F.$$

If $rd(F) > 0$, F may contain two kinds of conjuncts: basic concepts and (possibly probabilistic) existential restrictions. The basic concepts in F must appear in L as well by an argument analog to the case $rd(F) = 0$. Let $\exists r.F'$ be a top-level conjunct of F. Since $X \sqsubseteq_{\mathcal{T}'} F$ and $Y \sqsubseteq_{\mathcal{T}'} F$, completeness yields that there exists an $E \in S_0^A(X, r, 0)$ such that $F' \in S_0^A(E, 0)$ (i.e. $E \sqsubseteq_{\mathcal{T}'} F'$), and an $E' \in S_0^B(Y, r, 0)$ such that $F' \in S_0^B(E', 0)$ (i.e. $E' \sqsubseteq_{\mathcal{T}'} F'$). By induction hypothesis, it follows that $\mathsf{k\text{-}lcs\text{-}r}(E, E', \mathsf{S}, k-1, A, B) \sqsubseteq_{\mathcal{T}'} F'$, and thus also $L \sqsubseteq_{\mathcal{T}'} \exists r.\mathsf{k\text{-}lcs\text{-}r}(E, E', \mathsf{S}, k-1, A, B) \sqsubseteq_{\mathcal{T}'} \exists r.F'$. The other two cases of probabilistic existential conjuncts $P_{=1}\exists r.F'$ and $P_{>0}\exists r.F'$ of F are similar, so together we get $L \sqsubseteq_{\mathcal{T}'} F$. □

Together, Lemmata 24 and 25 fulfill all requirements of the definition of role-depth bounded least common subsumer. Thus, Theorem 13 is a direct consequence of both lemmas and the fact that the k-lcs procedure introduces new concept names A and B for the concepts C and D and then calls the procedure k-lcs-r for these new concept names A and B, using the completion sets of the extended and normalized TBox.

B.4. Proof for Theorem 14

As for the k-lcs, we split the proof for the correctness of the k-msc algorithm for Prob-$\mathcal{ELO}_c^{01}$ in two parts: first we show that k-msc computes indeed a concept that has the given individual as instance, then we show that this concept is the least one w.r.t. subsumption.

Lemma 26. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a Prob-$\mathcal{ELO}_c^{01}$-KB, $\mathcal{T}'$ be the TBox obtained from $\mathcal{T}$ by absorbing $\mathcal{A}$ and applying the normalization rules,* S *be the set of completion sets obtained from $\mathcal{T}'$, X be a basic concept with $\top \rightsquigarrow_R X$, k be a natural number and $L = \mathsf{traversal\text{-}concept}(X, \mathsf{S}, k)$. Then $X \sqsubseteq_{\mathcal{T}'} L$.*

Proof. By induction on k. For $k = 0$, the result $L = \bigsqcap E$ of traversal-concept is a conjunction of (possibly probabilistic) concept names $E \in S_0^\top(X, 0) \cap \mathsf{BC}_\mathcal{K}$, but no existential restrictions. By soundness of the completion rules, we know that $E \in S_0^\top(X, 0)$ implies $X \sqsubseteq_{\mathcal{T}'} E$. Since L contains exactly those conjuncts, we also have $X \sqsubseteq_{\mathcal{T}'} L$.

For the case $k > 0$, L is a conjunction of concept names (possibly probabilistic) and existential restrictions $\exists r.E$, $P_{=1}\exists r.E$, and $P_{>0}\exists r.E$. For the concept names, the same argument as for the case $k = 0$ applies. For existential restrictions of the form $\exists r.\mathsf{traversal\text{-}concept}(E, \mathsf{S}, k-1)$ with $E \in S_0^\top(X, r, 0)$, soundness yields $X \sqsubseteq_{\mathcal{T}'} \exists r.E$. Then the induction hypothesis yields that for $L' = \mathsf{traversal\text{-}concept}(E, \mathsf{S}, k-1)$ we have $E \sqsubseteq_{\mathcal{T}'} L'$ and thus also $X \sqsubseteq_{\mathcal{T}'} \exists r.L'$.

Similarly, by soundness we get that $E \in S_0^\top(X, r, 1)$ implies $X \sqsubseteq_{\mathcal{T}'} P_{=1}\exists r.E$ and $E' \in S_0^\top(X, r, v)$ for $v \in V \setminus \{0\}$ implies $X \sqsubseteq_{\mathcal{T}'} P_{>0}\exists r.E'$. By induction hypothesis we have that $E \sqsubseteq_{\mathcal{T}'} \mathsf{traversal\text{-}concept}(E, \mathsf{S}, k-1)$ and thus it follows that $X \sqsubseteq_{\mathcal{T}'} P_{=1}\exists r.\mathsf{traversal\text{-}concept}(E, \mathsf{S}, k-1)$ and analogously we get that $X \sqsubseteq_{\mathcal{T}'} P_{>0}\exists r.\mathsf{traversal\text{-}concept}(E', \mathsf{S}, k-1)$. Together, this means $X \sqsubseteq_{\mathcal{T}'} L$. □

Lemma 27. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a Prob-$\mathcal{ELO}_c^{01}$-KB, $\mathcal{T}'$ be the TBox obtained from $\mathcal{T}$ by absorbing $\mathcal{A}$ and applying the normalization rules,* S *be the set of completion sets obtained from $\mathcal{T}'$, X be a basic concept with $\top \rightsquigarrow_R X$, k be a natural number and $L = \mathsf{traversal\text{-}concept}(X, \mathsf{S}, k)$. Then for each Prob-$\mathcal{ELO}_c^{01}$-concept F with $\mathsf{Sig}(F) \subseteq \mathsf{Sig}(\mathcal{K})$ and $rd(F) \leqslant k$, $X \sqsubseteq_{\mathcal{T}'} F$ implies $L \sqsubseteq_{\mathcal{T}'} F$.*

Proof. By induction on the role-depth $rd(F)$. Let $rd(F) = 0$, i.e. $F = \bigsqcap E_i$ contains no existential restrictions. Since $X \sqsubseteq_{\mathcal{T}'} F$, we also have $X \sqsubseteq_{\mathcal{T}'} E_i$ for all conjuncts E_i of F. Then, completeness of the completion algorithm yields that $E_i \in S_0^X(X, 0)$ and since $\top \rightsquigarrow_R X$ also $E_i \in S_0^\top(X, 0)$. Therefore

$$L = \bigsqcap_{E \in S_0^\top(X,0) \cap \mathsf{BC}_\mathcal{T}} E \sqsubseteq_{\mathcal{T}'} F.$$

If $rd(F) > 0$, F may contain two kinds of conjuncts: basic concepts and (possibly probabilistic) existential restrictions. The basic concepts in F must appear in L as well by the same argument as in case $rd(F) = 0$. Let $\exists r.F'$ be a top-level conjunct of F. Since $X \sqsubseteq_{\mathcal{T}'} F \sqsubseteq_{\mathcal{T}'} \exists r.F'$, completeness yields that there exists an $E \in S_0^\top(X, r, 0)$ such that $F' \in S_0^\top(E, 0)$, i.e. $E \sqsubseteq_{\mathcal{T}'} F'$. Since $rdF' < rdF$, the induction hypothesis yields $\mathsf{traversal\text{-}concept}(E, \mathsf{S}, k-1) \sqsubseteq_{\mathcal{T}'} F'$, and thus also $L \sqsubseteq_{\mathcal{T}'} \exists r.\mathsf{traversal\text{-}concept}(E, \mathsf{S}, k-1) \sqsubseteq_{\mathcal{T}'} \exists r.F'$. The other two cases of probabilistic existential conjuncts $P_{=1}\exists r.F'$ and $P_{>0}\exists r.F'$ of F are similar, so together we get $L \sqsubseteq_{\mathcal{T}'} F$. □

Since $\mathcal{K} \models C(a)$ and $\mathcal{T}' \models \{a\} \sqsubseteq C$ are equivalent if $\mathcal{T}'$ arises from $\mathcal{K}$ by absorbing the ABox (see Lemma 10), the previous two lemmata imply the correctness of Theorem 14, as k-msc algorithm absorbs the ABox into the TBox $\mathcal{T}'$ and then computes the traversal-concept of the nominal $\{a\}$. Then Lemma 26 implies $\{a\} \sqsubseteq_{\mathcal{T}'} C$, i.e. $\mathcal{K} \models C(a)$, and Lemma 27 implies that for any concept F with $rd(F) \leqslant k$, $\mathcal{K} \models F(a)$ implies $\mathcal{K} \models C(a)$.

References

[1] C. Lutz, L. Schröder, Probabilistic description logics for subjective uncertainty, in: F. Lin, U. Sattler, M. Truszczynski (Eds.), Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-10), AAAI Press, 2010.

[2] R. Peñaloza, A.-Y. Turhan, A practical approach for computing generalization inferences in $\mathcal{EL}$, in: M. Grobelnik, E. Simperl (Eds.), Proc. of the 8th European Semantic Web Conf. (ESWC'11), in: Lecture Notes in Computer Science, Springer, 2011.

Please cite this article in press as: A. Ecke et al., Completion-based generalization inferences for the Description Logic $\mathcal{ELOR}$ with subjective probabilities, International Journal of Approximate Reasoning (2014), http://dx.doi.org/10.1016/j.ijar.2014.03.001

[3] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider (Eds.), The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press, 2003.
[4] F. Baader, S. Brandt, C. Lutz, Pushing the $\mathcal{EL}$ envelope, in: Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI-05), Morgan Kaufmann Publishers, Edinburgh, UK, 2005.
[5] R. Cote, D. Rothwell, J. Palotay, R. Beckett, L. Brochu, The systematized nomenclature of human and veterinary medicine, Technical Report SNOMED International, College of American Pathologists, Northfield, IL, 1993.
[6] K. Spackman, Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT, J. Am. Med. Inform. Assoc. (2000), Fall Symposium Special Issue.
[7] T.G.O. Consortium, Gene ontology: tool for the unification of biology, Nat. Genet. 25 (2000) 25–29.
[8] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, C. Lutz, OWL 2 web ontology language profiles, W3C Recommendation, http://www.w3.org/TR/2009/REC-owl2-profiles-20091027/, 2009.
[9] A. Ecke, R. Peñaloza, A.-Y. Turhan, Computing role-depth bounded generalizations in the description logic $\mathcal{ELOR}$, in: Proceedings of the 36th German Conference on Artificial Intelligence (KI 2013), in: Lecture Notes in Artificial Intelligence, vol. 8077, Springer-Verlag, Koblenz, Germany, 2013.
[10] T. Lukasiewicz, Expressive probabilistic description logics, Artif. Intell. 172 (2008) 852–883.
[11] P. Klinov, B. Parsia, U. Sattler, On correspondences between probabilistic first-order and description logics, in: B. Cuenca Grau, I. Horrocks, B. Motik, U. Sattler (Eds.), International Workshop on Description Logic (DL-2009), in: CEUR Workshop Proceedings, vol. 477, CEUR-WS.org, 2009.
[12] T. Lukasiewicz, U. Straccia, Managing uncertainty and vagueness in description logics for the semantic web, J. Web Semant. 6 (2008) 291–308.
[13] J.Y. Halpern, An analysis of first-order logics of probability, Artif. Intell. 46 (1990) 311–350.
[14] W.W. Cohen, A. Borgida, H. Hirsh, Computing least common subsumers in description logics, in: W. Swartout (Ed.), Proc. of the 10th Nat. Conf. on Artificial Intelligence (AAAI-92), AAAI Press/The MIT Press, San Jose, CA, 1992, pp. 754–760.
[15] F. Baader, R. Küsters, R. Molitor, Computing least common subsumers in description logics with existential restrictions, in: T. Dean (Ed.), Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99), Morgan Kaufmann, Los Altos, Stockholm, Sweden, 1999, pp. 96–101.
[16] S. Brandt, A.-Y. Turhan, Using non-standard inferences in description logics — what does it buy me?, in: G. Görz, V. Haarslev, C. Lutz, R. Möller (Eds.), Proc. of the 2001 Applications of Description Logic Workshop (ADL 2001), in: CEUR Workshop Proceedings, vol. 44, RWTH Aachen, Vienna, Austria, 2001. See http://CEUR-WS.org/Vol-44/.
[17] A. Borgida, T. Walsh, H. Hirsh, Towards measuring similarity in description logics, in: Proc. of the 2005 Description Logic Workshop (DL 2005), in: CEUR Workshop, vol. 147, 2005.
[18] C. d'Amato, N. Fanizzi, F. Esposito, A semantic similarity measure for expressive description logics, in: Proc. of Convegno Italiano di Logica Computazionale, CILC05, 2005.
[19] R. Möller, V. Haarslev, B. Neumann, Semantics-based information retrieval, in: Proceedings IT&KNOWS-98: International Conference on Information Technology and Knowledge Systems, Vienna, Budapest, 1998, p. 49.
[20] M.-S. Hacid, A. Léger, C. Rey, F. Toumani, Dynamic discovery of e-services in a knowledge representation and reasoning context, in: Proceedings of the 18th French conference on advanced databases (BDA), Evry, France, 2002, pp. 21–25.
[21] A. Ecke, R. Peñaloza, A.-Y. Turhan, Towards instance query answering for concepts relaxed by similarity measures, in: Workshop on Weighted Logics for AI (in conjunction with IJCAI'13), Beijing, China, 2013.
[22] F. Baader, Terminological cycles in a description logic with existential restrictions, in: G. Gottlob, T. Walsh (Eds.), Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI-03), Morgan Kaufmann, 2003, pp. 319–324.
[23] R. Küsters, R. Molitor, Approximating most specific concepts in description logics with existential restrictions, in: F. Baader, G. Brewka, T. Eiter (Eds.), Proc. of the 24th German Annual Conf. in Artificial Intelligence (KI'01), in: Lecture Notes in Artificial Intelligence, vol. 2174, Springer, 2001, pp. 33–47.
[24] C. Lutz, R. Piro, F. Wolter, $\mathcal{EL}$-concepts go second-order: Greatest fixpoints and simulation quantifiers, in: H. Coelho, R. Studer, M. Wooldridge (Eds.), Proc. of the 19th European Conf. on Artificial Intelligence (ECAI-10), in: Frontiers in Artificial Intelligence and Applications, vol. 215, IOS Press, 2010, pp. 41–46.
[25] S. Brandt, Standard and non-standard reasoning in description logics, Ph.D. thesis, Institute for theoretical computer science, TU Dresden, 2006.
[26] R. Küsters, R. Molitor, Approximating most specific concepts in description logics with existential restrictions, AI Commun. 15 (2002) 47–59.
[27] A. Ecke, A.-Y. Turhan, Role-depth bounded least common subsumers for $\mathcal{EL}^+$ and $\mathcal{ELI}$, in: Y. Kazakov, D. Lembo, F. Wolter (Eds.), Proc. of the 2012 Description Logic Workshop (DL 2012), in: CEUR Workshop Proceedings, vol. 846, 2012.
[28] R. Peñaloza, A.-Y. Turhan, Instance-based non-standard inferences in $\mathcal{EL}$ with subjective probabilities, in: F. Bobillo, P.C.G. Costa, C. d'Amato, N. Fanizzi, K.B. Laskey, K.J. Laskey, T. Lukasiewicz, M. Nickles, M. Pool (Eds.), Uncertainty Reasoning for the Semantic Web II, in: Lecture Notes in Computer Science, vol. 7123, Springer, 2013, pp. 80–98.
[29] F. Baader, C. Lutz, B. Suntisrivaraporn, CEL—a polynomial-time reasoner for life science ontologies, in: U. Furbach, N. Shankar (Eds.), Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR'06), in: Lecture Notes in Artificial Intelligence, vol. 4130, Springer-Verlag, 2006, pp. 287–291.
[30] Y. Kazakov, M. Krötzsch, F. Simancik, Concurrent classification of el ontologies, in: L. Aroyo, C. Welty, H. Alani, J. Taylor, A. Bernstein, L. Kagal, N.F. Noy, E. Blomqvist (Eds.), Proceedings of the 10th International Semantic Web Conference (ISWC 2011), Part I, in: Lecture Notes in Computer Science, vol. 7031, Springer, 2011, pp. 305–320.
[31] J. Mendez, jcel: A modular rule-based reasoner, in: Proceedings of the 1st International Workshop on OWL Reasoner Evaluation (ORE 2012), in: CEUR Workshop Proceedings, vol. 858, CEUR-WS.org, 2012.
[32] Y. Kazakov, M. Krötzsch, F. Simančík, Practical reasoning with nominals in the $\mathcal{EL}$ family of description logics, in: G. Brewka, T. Eiter, S.A. McIlraith (Eds.), Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-12), AAAI Press, 2012, pp. 264–274.
[33] Y. Kazakov, Consequence-driven reasoning for horn SHIQ ontologies, in: Proceedings of the 21st International Conference on Artificial Intelligence (IJCAI 2009), 2009, pp. 2040–2045.
[34] B. Zarrieß, A.-Y. Turhan, Most specific generalizations w.r.t. general $\mathcal{EL}$-TBoxes, in: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI'13), AAAI Press, Beijing, China, 2013.
[35] A.-Y. Turhan, B. Zarrieß, Computing the lcs w.r.t. general $\mathcal{EL}^+$-TBoxes, in: Proceedings of the 26th International Workshop on Description Logics (DL-2013), CEUR Workshop Proceedings, CEUR-WS.org, Ulm, Germany, 2013.
[36] O. Curé, Y. Prié, P.-A. Champin, A knowledge-based approach to augment applications with interaction traces, in: A. ten Teije, J. Völker, S. Handschuh, H. Stuckenschmidt, M. d'Aquin, A. Nikolov, N. Aussenac-Gilles, N. Hernandez (Eds.), Proc. of the 18th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2012), in: Lecture Notes in Computer Science, vol. 7603, Springer, 2012, pp. 317–326.
[37] J. Mendez, A. Ecke, A.-Y. Turhan, Implementing completion-based inferences for the $\mathcal{EL}$-family, in: R. Rosati, S. Rudolph, M. Zakharyaschev (Eds.), Proc. of the 2011 Description Logic Workshop (DL 2011), in: CEUR Workshop Proceedings, vol. 745, 2011.
[38] F. Baader, S. Brandt, C. Lutz, Pushing the $\mathcal{EL}$ envelope further, in: K. Clark, P.F. Patel-Schneider (Eds.), Proc. of the OWLED Workshop, 2008.
[39] S. Brandt, R. Küsters, A.-Y. Turhan, Approximation and difference in description logics, in: D. Fensel, D. McGuinness, M.-A. Williams (Eds.), Proc. of the 8th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-02), Morgan Kaufmann Publishers, San Francisco, CA, 2002.
[40] A.-Y. Turhan, On the computation of common subsumers in description logics, Ph.D. thesis, TU Dresden, Institute for Theoretical Computer Science, 2007.

[41] V. Gutiérrez-Basulto, J.C. Jung, C. Lutz, L. Schröder, A closer look at the probabilistic description logic prob-$\mathcal{EL}$, in: Proceedings of Twenty-Fifth Conference on Artificial Intelligence (AAAI-11), 2011.

[42] A. Ecke, A.-Y. Turhan, Optimizations for the role-depth bounded least common subsumer in $\mathcal{EL}+$, in: P. Klinov, M. Horridge (Eds.), Proceedings of OWL: Experiences and Directions Workshop, in: CEUR Workshop Proceedings, vol. 849, 2012.

[43] A.-Y. Turhan, R. Molitor, Using lazy unfolding for the computation of least common subsumers, in: C. Goble, R. Möller, P. Patel-Schneider (Eds.), Proc. of the 2001 Description Logic Workshop (DL 2001), in: CEUR Workshop Proceedings, vol. 49, 2001. See http://CEUR-WS.org/Vol-49/.

[44] F. Baader, A.-Y. Turhan, On the problem of computing small representations of least common subsumers, in: M. Jarke, J. Köhler, G. Lakemeyer (Eds.), Proc. of the 25th German Annual Conf. in Artificial Intelligence (KI'02), in: Lecture Notes in Artificial Intelligence, vol. 2479, Springer, 2002.

[45] F. Baader, C. Lutz, B. Suntisrivaraporn, CEL—a polynomial-time reasoner for life science ontologies, in: U. Furbach, N. Shankar (Eds.), Proc. of the 3rd Int. Joint Conf. in Automated Reasoning (IJCAR-06), in: Lecture Notes in Artificial Intelligence, vol. 4130, Springer-Verlag, 2006, pp. 287–291. CEL download page: http://lat.inf.tu-dresden.de/systems/cel/.

[46] M. Ashburner, Gene ontology: Tool for the unification of biology, Nat. Genet. 25 (2000) 25–29.

[47] A. Rector, I. Horrocks, Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions, in: Proceedings of the Workshop on Ontological Engineering, AAAI Spring Symposium (AAAI'97), AAAI Press, Stanford, CA, 1997.

[48] F. Baader, R. Peñaloza, B. Suntisrivaraporn, Pinpointing in the description logic $\mathcal{EL}$, in: Proceedings of the 30th German Conference on Artificial Intelligence (KI2007), in: Lecture Notes in Artificial Intelligence, vol. 4667, Springer-Verlag, Osnabrück, Germany, 2007, pp. 52–67.

[49] F. Baader, R. Peñaloza, Automata-based axiom pinpointing, in: Selected Papers from IJCAR 2008, J. Autom. Reason. 45 (Special Issue) (2010) 91–129.

[50] F. Baader, R. Peñaloza, Axiom pinpointing in general tableaux, in: Tableaux and Analytic Proof Methods, J. Log. Comput. 20 (Special Issue) (2010) 5–34.

Please cite this article in press as: A. Ecke et al., Completion-based generalization inferences for the Description Logic $\mathcal{ELOR}$ with subjective probabilities, International Journal of Approximate Reasoning (2014), http://dx.doi.org/10.1016/j.ijar.2014.03.001