Least General Generalizations in Description Logic: Verification and Existence

Jean Christoph Jung,[1] Carsten Lutz,[1] Frank Wolter[2]
[1]University of Bremen, Germany
[2]University of Liverpool, United Kingdom
{jeanjung, clu}@uni-bremen.de, wolter@liverpool.ac.uk

Abstract

We study two forms of least general generalizations in description logic, the least common subsumer (LCS) and most specific concept (MSC). While the LCS generalizes from examples that take the form of concepts, the MSC generalizes from individuals in data. Our focus is on the complexity of existence and verification, the latter meaning to decide whether a candidate concept is the LCS or MSC. We consider cases with and without a background TBox and a target signature. Our results range from coNP-complete for LCS and MSC verification in the description logic $\mathcal{EL}$ without TBoxes to undecidability of LCS and MSC verification and existence in $\mathcal{ELI}$ with TBoxes. To obtain results in the presence of a TBox, we establish a close link between the problems studied in this paper and concept learning from positive and negative examples. We also give a way to regain decidability in $\mathcal{ELI}$ with TBoxes and study single example MSC as a special case.

1 Introduction

Generalization is a fundamental method in relational learning and inductive logic programming (Plotkin 1970; Muggleton 1991). Given a finite number of positive examples, one seeks a description in a logical language that encompasses all examples and in this sense provides a generalization. To ensure that the description is as informative as possible, one aims at obtaining least general generalizations, that is, generalizations that cannot be made more specific without losing at least one example. Note that computing least general generalizations is a form of supervised learning in which only positive, but no negative examples are given.

In this paper, we study least general generalizations in the context of description logics (DLs), a widely known family of ontology languages that underpin the web ontology language OWL 2 (Baader et al. 2017). In DLs, *concepts* are the building blocks of an ontology and thus a prime target for being learned through generalization. There are in fact several applications in which this is useful, including ontology design by domain experts that are not sufficiently proficient in logical modeling (Baader and Küsters 1998; Baader, Küsters, and Molitor 1999; Baader, Sertkaya, and Turhan 2007; Donini et al. 2009), supporting the improvement and restructuring of an ontology (Cohen, Borgida, and Hirsh 1992; Küsters and Borgida 2001), and creative discovery of novel concepts through conceptual blending (Fauconnier and Turner 2008; Eppe et al. 2018). We focus on the two fundamental DLs $\mathcal{EL}$ and $\mathcal{ELI}$, fragments of first-order Horn logic that can express positive conjunctive existential properties, $\mathcal{ELI}$ extending $\mathcal{EL}$ with inverse roles. Both DLs are natural choices for generalization as their limited expressive power helps to avoid overfitting, that is, we cannot generalize by disjunctively combining descriptions of each single example, but are forced to find a true generalization. In fact, least general generalizations in $\mathcal{EL}$ have received significant attention (Baader, Küsters, and Molitor 1999; Baader 2003; Zarrieß and Turhan 2013) while, somewhat surprisingly, there appears to be no prior work on DLs with inverse roles.

There are two established notions of least general generalization in the DL context. When the examples are given in the form of concepts, the desired generalization is the *least common subsumer (LCS)*, the least general concept that subsumes all examples (Cohen, Borgida, and Hirsh 1992). A natural alternative is to give examples using relational data, which in DLs are represented as an ABox. Traditionally, one uses only a single example, which takes the form of an individual in the data, and then asks for the *most specific concept (MSC)*, that is, the least general concept that the individual is an instance of (Nebel 1990). However, there seems to be no good reason to restrict the MSC to a single example and thus we define it based on multiple examples. In this way, the LCS becomes a special form of MSC in which the data consists of a collection of trees. We remark that $\mathcal{EL}$ and $\mathcal{ELI}$ concepts can be viewed as natural tree query languages for graph databases and knowledge graphs and thus the MSC is useful for data exploration and comprehension, see e.g. (Colucci et al. 2016). It is also related to generating referring expressions (Borgida, Toman, and Weddell 2016).

For both the LCS and the MSC, we study the two decision problems *existence* and *verification*. In fact, both the LCS and the MSC need not exist because there can be an infinite sequence of less and less general generalizations. In verification, one is given a candidate concept and the question is whether the candidate is the LCS or MSC. Verification is relevant, for example, in approaches that try to find the LCS or MSC by refinement operators that move towards less general generalizations in a step-wise fashion (Badea and Nienhuys-Cheng 2000; Lehmann and Hitzler 2010; Lehmann and

Haase 2009) and check after each step whether the least general generalization has already been reached. We consider the case with and without a background TBox and with and without a target signature that the generalization should be formulated in. If the generalization does not exist, one can resort to approximations (Küsters and Molitor 2001; Baader, Sertkaya, and Turhan 2007).

We now summarize our main complexity and undecidability results. They are based on characterizations in terms of simulations between products of universal models, mildly varying characterizations given in (Zarrieß and Turhan 2013; Funk et al. 2019). We start with the case without TBoxes, for which we find LCS and MSC verification in $\mathcal{EL}$ to be CONP-complete. It is well-known that the LCS in $\mathcal{EL}$ always exists (Baader, Küsters, and Molitor 1999), and we complement this by proving that MSC existence in $\mathcal{EL}$ is PSPACE-complete. We then add inverse roles which introduce significant technical challenges. In particular, the structure of the relevant products from the mentioned characterizations is much more complex. As a consequence, the LCS in $\mathcal{ELI}$ is not guaranteed to exist. We prove that LCS and MSC existence and verification are PSPACE-hard and in EXPTIME. The lower bounds require a remarkably intricate construction and show as a by-product that the *product simulation problem* on trees (defined in the paper) is PSPACE-hard.

We then switch to the case with TBoxes, starting with observing a connection to concept learning (Badea and Nienhuys-Cheng 2000; Lehmann and Hitzler 2010; Lehmann and Haase 2009; Lisi 2012; Bühmann et al. 2018; Sarker and Hitzler 2019) and in particular to the concept separability problem (Funk et al. 2019) which asks whether there is a concept that separates given positive examples from given negative examples. It turns out that its complement reduces in polynomial time to MSC existence. Using results from (Funk et al. 2019), this can be used to show that MSC existence is undecidable in $\mathcal{ELI}$ and EXPTIME-complete in $\mathcal{EL}$. The same is true for verification as the two problems are mutually reducible in polynomial time when a TBox can be used. We consider it remarkable that inverse roles have such a dramatic computational effect. We also identify a way around undecidability, namely to consider for the generalization only *symmetry free* $\mathcal{ELI}$ concepts, that is, $\mathcal{ELI}$ concepts that do not admit a subconcept of the form $\exists r.(C \sqcap \exists r^-.D)$. In this case, the complexity drops to EXPTIME again. Up to this point, all mentioned complexity lower bounds and undecidability results hold without a signature restriction on the target concept while all upper bounds apply also with such a restriction. We finally consider the MSC of single examples and show that existence and verification are in PTIME in $\mathcal{EL}$ while they are complete for EXPTIME and 2EXPTIME in $\mathcal{ELI}$, depending on whether or not we assume the signature to be full. Thus once more, adding inverse roles has a drastic effect.

Note that in the literature, the LCS is sometimes restricted to only constantly many examples. In all of the above results, we do not assume a constant bound on the number of examples. We also make observations regarding that case, though. Without a TBox, the complexity typically drops to PTIME and the same is true for $\mathcal{EL}$ with TBoxes (Zarrieß and Turhan 2013). When both inverse roles and TBoxes are present, however, the complexity tends to not decrease. We remark that in the decidable cases, our constructions yield upper bounds on the role depth of the LCS and MSC, if they exists, which together with the characterizations can be used to actually construct them.

A full version that contains all proof details is available at http://www.informatik.uni-bremen.de/tdki/research/.

2 Preliminaries

We introduce the basics of DLs as required for this paper, for full details see (Baader et al. 2017). Let $\mathsf{N_C}$ be a set of *concept names* and $\mathsf{N_R}$ a set of *role names*, both countably infinite. A *role* is either a role name or an *inverse role* r^-, r a role name. For uniformity, we identify $(r^-)^-$ with r. An $\mathcal{ELI}$ *concept* is formed according to the syntax rule

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

where A ranges over concept names and r over roles. An $\mathcal{EL}$ concept is an $\mathcal{ELI}$ concept that does not use inverse roles. The *depth* of a concept refers to the nesting depth of the operator $\exists r.C$.

For any DL $\mathcal{L}$, an $\mathcal{L}$ *TBox* is a finite set of *concept inclusions* (CIs) $C \sqsubseteq D$, where C and D are $\mathcal{L}$ concepts. Let $\mathsf{N_I}$ be a countably infinite set of *individual names*. An *ABox* $\mathcal{A}$ is a finite set of *concept assertions* $A(a)$ and *role assertions* $r(a, b)$ where $A \in \mathsf{N_C}$, $r \in \mathsf{N_R}$, and $a, b \in \mathsf{N_I}$. We often use $r(a, b)$ to denote $r^-(b, a)$ if r is an inverse role. We use $\mathsf{ind}(\mathcal{A})$ to denote the set of all individual names that occur in $\mathcal{A}$. An $\mathcal{L}$ *knowledge base* (KB) $(\mathcal{T}, \mathcal{A})$ consists of an $\mathcal{L}$ TBox $\mathcal{T}$ and an ABox $\mathcal{A}$.

The semantics of DLs is defined in terms of *interpretations* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set and $\cdot^\mathcal{I}$ maps each concept name $A \in \mathsf{N_C}$ to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$ and each role name $r \in \mathsf{N_R}$ to a binary relation $r^\mathcal{I}$ on $\Delta^\mathcal{I}$. We refer to (Baader et al. 2017) for details on how to extend $\cdot^\mathcal{I}$ to compound concepts. An interpretation $\mathcal{I}$ *satisfies* a CI $C \sqsubseteq D$ if $C^\mathcal{I} \subseteq D^\mathcal{I}$, a concept assertion $A(a)$ if $a \in A^\mathcal{I}$, and a role assertion $r(a, b)$ if $(a, b) \in r^\mathcal{I}$. $\mathcal{I}$ is a *model* of a TBox, an ABox, or a knowledge base if it satisfies all inclusions and assertions in it. The CI $C \sqsubseteq D$ is a *consequence of the TBox* $\mathcal{T}$, in symbols $\mathcal{T} \models C \sqsubseteq D$, if $C^\mathcal{I} \subseteq D^\mathcal{I}$ for all models $\mathcal{I}$ of $\mathcal{T}$. For a KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a concept C, and an individual $a \in \mathsf{ind}(\mathcal{A})$, we write $\mathcal{K} \models C(a)$ if $a \in C^\mathcal{I}$ for all models $\mathcal{I}$ of $\mathcal{K}$. For a DL $\mathcal{L}$, $\mathcal{L}$ *instance checking* is the problem to decide, given an $\mathcal{L}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, an $a \in \mathsf{ind}(\mathcal{A})$, and an $\mathcal{L}$ concept C, whether $\mathcal{K} \models C(a)$.

A *signature* Σ is a set of concept and role names. An $\mathcal{L}$ concept is an $\mathcal{L}(\Sigma)$ concept if it uses only concept and role names from Σ, and likewise for other syntactic objects such as TBoxes and ABoxes. The *signature* $\mathsf{sig}(O)$ of a syntactic object O is the set of concept and role names that occur in O. The Σ-*reduct* $\mathcal{I}_{|\Sigma}$ of an interpretation $\mathcal{I}$ is obtained from $\mathcal{I}$ by setting $A^\mathcal{I} = \emptyset$ and $r^\mathcal{I} = \emptyset$ for all concept names A and role names r not in Σ.

Each interpretation $\mathcal{I}$ gives rise to a directed graph $G_\mathcal{I} = (\Delta^\mathcal{I}, \{(d, e) \mid (d, e) \in r^\mathcal{I}\})$ and a corresponding undirected graph $G^u_\mathcal{I}$. We thus apply graph theoretic terminology directly to interpretations, speaking for example about their outdegree. An interpretation is *tree-shaped* (resp. *ditree-shaped*) if $G^u_{\mathcal{I}}$ (resp. $G_{\mathcal{I}}$) is a tree without multiedges, that is, $(d,e) \in r^{\mathcal{I}} \cap s^{\mathcal{I}}$ implies $r = s$ for all roles r, s. Each $\mathcal{ELI}$ (resp. $\mathcal{EL}$) concept C can be viewed as a tree-shaped (resp. ditree-shaped) interpretation and vice versa. All this also applies to ABoxes, which are only a different way to present finite interpretations. We use $\mathcal{A}_C$ to denote the $\mathcal{ELI}$ concept C viewed as a tree-shaped ABox and use ρ_C to denote the root of $\mathcal{A}_C$. For example, $C = A \sqcap \exists r.B \sqcap \exists r^-.\top$ gives $\mathcal{A}_C = \{A(\rho_C), r(\rho_C, b_1), B(b_1), r(b_2, \rho_C)\}$.

Lemma 1 *For all $\mathcal{ELI}$ TBoxes $\mathcal{T}$ and $\mathcal{ELI}$ concepts C, D, $\mathcal{T} \models C \sqsubseteq D$ iff $(\mathcal{T}, \mathcal{A}_C) \models D(\rho_C)$.*

We introduce simulations, universal models, and direct products. Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be interpretations. A relation $S \subseteq \Delta^{\mathcal{I}_1} \times \Delta^{\mathcal{I}_2}$ is an $\mathcal{EL}(\Sigma)$ *simulation* from $\mathcal{I}_1$ to $\mathcal{I}_2$ if for all $d, d' \in \Delta^{\mathcal{I}_1}$ and $e \in \Delta^{\mathcal{I}_2}$:

1. $d \in A^{\mathcal{I}_1}$ and $(d,e) \in S$ imply $e \in A^{\mathcal{I}_2}$, for all $A \in \Sigma$;
2. $(d,d') \in r^{\mathcal{I}_1}$ and $(d,e) \in S$ imply $(d',e') \in S$ and $(e,e') \in r^{\mathcal{I}_2}$ for some $e' \in \Delta^{\mathcal{I}_2}$, for all role names $r \in \Sigma$.

S is an $\mathcal{ELI}(\Sigma)$ *simulation* if Condition 2 also holds for inverse roles r^- with $r \in \Sigma$. Let $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$ and $(d,e) \in \Delta^{\mathcal{I}_1} \times \Delta^{\mathcal{I}_2}$. We write $(\mathcal{I}_1, d) \preceq_{\mathcal{L},\Sigma} (\mathcal{I}_2, e)$ if there exists an $\mathcal{L}(\Sigma)$ simulation from $\mathcal{I}_1$ to $\mathcal{I}_2$ that contains (d,e). We omit Σ if it is the *full signature* $\mathsf{N_C} \cup \mathsf{N_R}$, writing $\preceq_{\mathcal{L}}$ and speaking of $\mathcal{L}$ simulations. It can be checked in polynomial time whether $(\mathcal{I}_1, d) \preceq_{\mathcal{L},\Sigma} (\mathcal{I}_2, e)$. The following lemma shows that $\mathcal{L}(\Sigma)$ simulations characterize preservation of $\mathcal{L}(\Sigma)$ concepts.

Lemma 2 *Let $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$, let $\mathcal{I}_1, \mathcal{I}_2$ be interpretations with finite outdegree, and let Σ be a signature. The following are equivalent:*

1. *$(\mathcal{I}_1, d) \preceq_{\mathcal{L},\Sigma} (\mathcal{I}_2, e)$;*
2. *for all $\mathcal{L}(\Sigma)$ concepts C: if $d \in C^{\mathcal{I}_1}$, then $e \in C^{\mathcal{I}_2}$.*

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KB and $\mathsf{sub}(\mathcal{T})$ be the set of all subconcepts of concepts that occur in $\mathcal{T}$. A *type for* $\mathcal{T}$ is a subset $t \subseteq \mathsf{sub}(\mathcal{T})$ such that $\mathcal{T} \models \bigsqcap t \sqsubseteq D$ implies $D \in t$ for all $D \in \mathsf{sub}(\mathcal{T})$. Denote by T the set of all types for $\mathcal{T}$. When $a \in \mathsf{ind}(\mathcal{A})$, $t, t' \in T$, and r is a role, we write

- $a \rightsquigarrow^{\mathcal{K}}_r t$ if $\mathcal{K} \models \exists r.\bigsqcap t(a)$ and t is maximal with this condition, and
- $t \rightsquigarrow^{\mathcal{T}}_r t'$ if $\mathcal{T} \models \bigsqcap t \sqsubseteq \exists r.\bigsqcap t'$ and t' is maximal with this condition.

A *path* p for $\mathcal{K}$ is a sequence $ar_0t_1 \cdots r_{n-1}t_n$ such that $a \in \mathsf{ind}(\mathcal{A})$, $r_0, \ldots, r_{n-1}$ are roles, $t_1, \ldots, t_n \in T$, $a \rightsquigarrow^{\mathcal{K}}_{r_0} t_1$, and $t_i \rightsquigarrow^{\mathcal{T}}_{r_i} t_{i+1}$ for all $i < n$. Let $\mathsf{tail}(p)$ denote the last element of the path p. Define the *universal model $\mathcal{U}_{\mathcal{K}}$ of $\mathcal{K}$* by taking as $\Delta^{\mathcal{U}_{\mathcal{K}}}$ the set of all paths for $\mathcal{K}$ and setting for all concept names A and role names r:

$$\begin{aligned}
A^{\mathcal{U}_{\mathcal{K}}} &= \{a \in \mathsf{ind}(\mathcal{A}) \mid \mathcal{T}, \mathcal{A} \models A(a)\} \cup \\
&\quad \{p \in \Delta^{\mathcal{U}_{\mathcal{K}}} \setminus \mathsf{ind}(\mathcal{A}) \mid A \in \mathsf{tail}(p)\} \\
r^{\mathcal{U}_{\mathcal{K}}} &= \{(a,b) \in \mathsf{ind}(\mathcal{A})^2 \mid r(a,b) \in \mathcal{A}\} \cup \\
&\quad \{(p, prt) \mid prt \in \Delta^{\mathcal{U}_{\mathcal{K}}}\} \cup \{(pr^-t, p) \mid pr^-t \in \Delta^{\mathcal{U}_{\mathcal{K}}}\}
\end{aligned}$$

The *universal model* $\mathcal{U}_{\mathcal{T},C}$ of an $\mathcal{ELI}$ TBox $\mathcal{T}$ and an $\mathcal{ELI}$ concept C is defined as $\mathcal{U}_{\mathcal{K}}$ where $\mathcal{K} = (\mathcal{T}, \mathcal{A}_C)$.

Lemma 3 *For all $\mathcal{ELI}$ KBs $\mathcal{K}$, $\mathcal{ELI}$ concepts C, and $a \in \mathsf{ind}(\mathcal{K})$, $\mathcal{K} \models C(a)$ iff $a \in C^{\mathcal{U}_{\mathcal{K}}}$.*

The *direct product* $\prod_{i=1}^n \mathcal{I}_i$ of interpretations $\mathcal{I}_1, \ldots, \mathcal{I}_n$ is defined by

$$\begin{aligned}
\Delta^{\prod_{i=1}^n \mathcal{I}_i} &= \Delta^{\mathcal{I}_1} \times \cdots \times \Delta^{\mathcal{I}_n} \\
A^{\prod_{i=1}^n \mathcal{I}_i} &= A^{\mathcal{I}_1} \times \cdots \times A^{\mathcal{I}_n} \\
r^{\prod_{i=1}^n \mathcal{I}_i} &= \{((d_1, \ldots, d_n), (e_1, \ldots, e_n)) \mid \forall i : (d_i, e_i) \in r^{\mathcal{I}_i}\}
\end{aligned}$$

If $(d_1, \ldots, d_n) \in \Delta^{\prod_{i=1}^n \mathcal{I}_i}$, then we write $\prod_{i=1}^n (\mathcal{I}_i, d_i)$ for the pair $(\prod_{i=1}^n \mathcal{I}_i, (d_1, \ldots, d_n))$.

Lemma 4 *For all $\mathcal{I}_1, \ldots, \mathcal{I}_n$, $(d_1, \ldots, d_n) \in \Delta^{\prod_{i=1}^n \mathcal{I}_i}$, and $\mathcal{ELI}$ concepts C, $(d_1, \ldots, d_n) \in C^{\prod_{i=1}^n \mathcal{I}_i}$ iff $d_i \in C^{\mathcal{I}_i}$ for $1 \leq i \leq n$.*

3 LCS and MSC: Basics

We introduce least common subsumers and most specific concepts, discuss their relationship, and give model-theoretic characterizations for verification and existence. The latter are mild extensions of characterizations established in (Zarrieß and Turhan 2013).

Definition 1 *Let $\mathcal{T}$ be a TBox, $C_1, \ldots, C_n$ concepts called* examples, *$\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$, and Σ a signature. An $\mathcal{L}(\Sigma)$ concept D is a* least common $\mathcal{L}(\Sigma)$ subsumer ($\mathcal{L}(\Sigma)$-LCS) of $C_1, \ldots, C_n$ w.r.t. $\mathcal{T}$ *if*

1. *$\mathcal{T} \models C_i \sqsubseteq D$ for all $i = 1, \ldots, n$;*
2. *if $\mathcal{T} \models C_i \sqsubseteq D'$ for all $i = 1, \ldots, n$, D' an $\mathcal{L}(\Sigma)$ concept, then $\mathcal{T} \models D \sqsubseteq D'$.*

If an $\mathcal{L}(\Sigma)$-LCS w.r.t. a TBox $\mathcal{T}$ exists, then it is unique up to equivalence w.r.t. $\mathcal{T}$. We thus speak about *the* $\mathcal{L}(\Sigma)$-LCS. We omit Σ if it contains $\mathsf{sig}(\mathcal{T} \cup \{C_1, \ldots, C_n\})$, speaking of the *$\mathcal{L}$-LCS* w.r.t. $\mathcal{T}$. Clearly, no $\mathcal{L}$-LCS can contain symbols that are not in the TBox or the examples. Thus, all signatures between the finite $\mathsf{sig}(\mathcal{T} \cup \{C_1, \ldots, C_n\})$ and the full signature behave in the same way. We also omit $\mathcal{T}$ if it is empty, speaking of the *$\mathcal{L}(\Sigma)$-LCS*.

Example 1 (1) Let $C_1 = \exists \mathsf{attend}.\mathsf{MLConf}$ and $C_2 = \exists \mathsf{attend}.\mathsf{KRConf}$. Then $\exists \mathsf{attend}.\top$. is the $\mathcal{EL}$ (and $\mathcal{ELI}$) LCS of C_1, C_2. Let $\mathcal{T} = \{\mathsf{MLConf} \sqsubseteq \mathsf{AIConf}, \mathsf{KRConf} \sqsubseteq \mathsf{AIConf}\}$. Then $\exists \mathsf{attend}.\mathsf{AIConf}$ is the $\mathcal{EL}$ (and $\mathcal{ELI}$) LCS of C_1, C_2 w.r.t. $\mathcal{T}$.

(2) The $\mathcal{L}$-LCS, $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$, of a single $\mathcal{L}$ concept C w.r.t. an $\mathcal{L}$ TBox $\mathcal{T}$ is just C. For $\Sigma \subsetneq \mathsf{sig}(C)$, however, the $\mathcal{L}(\Sigma)$-LCS of C w.r.t. $\mathcal{T}$ does not always exist. Take, for example, $\mathcal{T} = \{A \sqsubseteq \exists r.A\}$ and $\Sigma = \{r\}$. Then neither the $\mathcal{ELI}(\Sigma)$-LCS nor the $\mathcal{EL}(\Sigma)$-LCS of A w.r.t. $\mathcal{T}$ exists as $\mathcal{T} \models A \sqsubseteq \exists r^n.\top$ for all $n \geq 0$, but there is no $\mathcal{ELI}(\Sigma)$ concept C with $\mathcal{T} \models A \sqsubseteq C$ and $\mathcal{T} \models C \sqsubseteq \exists r^n.\top$ for all n.

Definition 2 *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a KB, $a_1, \ldots, a_n \in \mathsf{ind}(\mathcal{A})$ individuals called* examples, *$\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$, and Σ a signature. An $\mathcal{L}(\Sigma)$ concept C is a* most specific $\mathcal{L}(\Sigma)$ concept ($\mathcal{L}(\Sigma)$ MSC) of $a_1, \ldots, a_n$ w.r.t. $\mathcal{K}$ *if*

1. *$\mathcal{K} \models C(a_i)$ for all $i = 1, \ldots, n$;*
2. *if $\mathcal{K} \models D(a_i)$ for all $i = 1, \ldots, n$, D an $\mathcal{L}(\Sigma)$ concept, then $\mathcal{T} \models C \sqsubseteq D$.*

Like the LCS, the MSC is unique up to equivalence w.r.t. $\mathcal{T}$ (if it exists) and thus we speak of *the* MSC. We drop Σ if $\Sigma \supseteq \mathsf{sig}(\mathcal{K})$. As for the LCS, a symbol that does not occur in the KB cannot occur in the MSC.

Example 2 *(1) In contrast to the $\mathcal{EL}$-LCS, the $\mathcal{EL}$-MSC of a single example does not always exist, even when the TBox is empty, due to cycles in the ABox. For example, for $\mathcal{A} = \{A(a), r(a,a)\}$ the $\mathcal{EL}$-MSC of a w.r.t. $\mathcal{K} = (\emptyset, \mathcal{A})$ does not exist (use that $\mathcal{K} \models \exists r^n.\top(a)$ for all $n \geq 0$). In contrast, the $\mathcal{EL}$-MSC of a w.r.t. $\mathcal{K}' = (\{A \sqsubseteq \exists r.A\}, \mathcal{A})$ is A.*

(2) A common proposal to generalize from individuals is to compute the MSC of each individual separately and then generalize by applying the LCS, provided that all MSCs exist (Baader, Küsters, and Molitor 1999). It pays off, however, to directly apply the MSC to multiple individuals. Let, for example, $\mathcal{K} = (\emptyset, \mathcal{A})$, $\mathcal{A} = \{A(a), r(a,a), A(b), s(b,b)\}$. Then the $\mathcal{EL}$-MSC of a alone w.r.t. $\mathcal{K}$ does not exist, and likewise for b. In constrast, the $\mathcal{EL}$-MSC of a, b w.r.t. $\mathcal{K}$ is A.

The next theorem, which is an immediate consequence of Lemma 1, shows that the LCS is a special form of MSC.

Theorem 1 *Let $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$, $\mathcal{T}$ be an $\mathcal{L}$ TBox, $C_1, \dots, C_n$ $\mathcal{L}$ concepts, and Σ a signature. Then an $\mathcal{L}(\Sigma)$ concept D is the $\mathcal{L}(\Sigma)$-LCS of $C_1, \dots, C_n$ w.r.t. $\mathcal{T}$ iff D is the $\mathcal{L}(\Sigma)$-MSC of $\rho_{C_1}, \dots, \rho_{C_n}$ w.r.t. the KB $(\mathcal{T}, \mathcal{A})$, $\mathcal{A} = \mathcal{A}_{C_1} \cup \dots \cup \mathcal{A}_{C_n}$.*

LCS and MSC give rise to the four decision problems studied in this paper. Let $\mathcal{L}$ be a description logic. *$\mathcal{L}$-LCS existence w.r.t. TBoxes* means to decide, given $\mathcal{L}$ concepts $C_1, \dots, C_n$, an $\mathcal{L}$ TBox $\mathcal{T}$, and a finite signature Σ, whether the $\mathcal{L}(\Sigma)$-LCS of $C_1, \dots, C_n$ w.r.t. $\mathcal{T}$ exists. By the remark made after Definition 1, it is without loss of generality to consider only finite signatures. In particular, we can use $\mathsf{sig}(\mathcal{T} \cup \{C_1, \dots, C_n\})$ instead of the full signature. *$\mathcal{L}$-MSC existence w.r.t. TBoxes* is defined accordingly, the input consisting of a KB $(\mathcal{T}, \mathcal{A})$ with $\mathcal{T}$ an $\mathcal{L}$ TBox, $a_1, \dots, a_n \in \mathsf{ind}(\mathcal{A})$, and a finite signature Σ. In *$\mathcal{L}$-LCS* (resp. *$\mathcal{L}$-MSC*) *verification w.r.t. TBoxes*, we are given as an additional input a candidate $\mathcal{L}(\Sigma)$ concept C and the question is whether C is the $\mathcal{L}(\Sigma)$-LCS of $C_1, \dots, C_n$ w.r.t. $\mathcal{T}$ (resp. the $\mathcal{L}(\Sigma)$-MSC of $a_1, \dots, a_n$ w.r.t. $\mathcal{K}$).

Theorem 1 provides a reduction from $\mathcal{L}$-LCS existence w.r.t. TBoxes to $\mathcal{L}$-MSC existence w.r.t. TBoxes, and likewise for verification. In this reduction, neither the TBox nor the signature nor the number of examples change. We now present a converse reduction which, however, requires to modify the TBox.

Theorem 2 *Let $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$. Then $\mathcal{L}$-MSC verification (resp. existence) w.r.t. TBoxes can be reduced in polynomial time to $\mathcal{L}$-LCS verification (resp. existence). This also holds in the full signature case if there are at least two examples.*

Proof. Let $\mathcal{T}$ be an $\mathcal{L}$ TBox, $\mathcal{A}$ an ABox, $a_1, \dots, a_n \in \mathsf{ind}(\mathcal{A})$. We may assume w.l.o.g. that $\mathcal{A}$ is the disjoint union of ABoxes $\mathcal{A}_1, \dots, \mathcal{A}_n$ such that $a_i \in \mathsf{ind}(\mathcal{A}_i)$ for $i = 1, \dots, n$. Let X_a be a fresh concept name for every $a \in \mathsf{ind}(\mathcal{A})$ and let $\mathcal{T}'$ be the extension of $\mathcal{T}$ with

$$X_a \sqsubseteq A \qquad \text{for all } A(a) \in \mathcal{A},$$
$$X_a \sqsubseteq \exists r.X_{a'} \qquad \text{for all } r(a,a') \in \mathcal{A}.$$

(If $\mathcal{L} = \mathcal{ELI}$, then also add $X_a \sqsubseteq \exists r^-.X_{a'}$ if $r(a',a) \in \mathcal{A}$.) Then for every signature Σ that does not contain $\{X_{a_1}, \dots, X_{a_n}\}$ and every $\mathcal{L}(\Sigma)$ concept D, D is the $\mathcal{L}(\Sigma)$-MSC of $a_1, \dots, a_n$ w.r.t. $(\mathcal{T}, \mathcal{A})$ iff D is the $\mathcal{L}(\Sigma)$-LCS of $X_{a_1}, \dots, X_{a_n}$ w.r.t. $\mathcal{T}'$.

In the case of the full signature, we have to consider the $\mathcal{L}(\Sigma \cup \{X_{a_1}, \dots, X_{a_n}\})$-LCS in place of the $\mathcal{L}(\Sigma)$-LCS. The assumption that there are at least two examples ensures that the concept names X_a cannot occur in the LCS. ❑

We next provide model-theoretic characterizations for MSC verification and existence based on products and simulations. Corresponding characterizations for LCS verification and existence can be obtained in a straightforward way via Theorem 1, see the appendix. Note that Point 1 below can also be viewed as a simulation condition.

Theorem 3 (MSC Verification) *Let $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$, $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{L}$ KB, $a_1, \dots, a_n \in \mathsf{ind}(\mathcal{A})$, and Σ a signature. An $\mathcal{L}(\Sigma)$ concept C is the $\mathcal{L}(\Sigma)$-MSC of $a_1, \dots, a_n$ w.r.t. $\mathcal{K}$ iff the following conditions hold:*

1. *$(a_1, \dots, a_n) \in C^{\Pi_{i=1}^n \mathcal{U}_\mathcal{K}}$;*
2. *$\Pi_{i=1}^n (\mathcal{U}_\mathcal{K}, a_i) \preceq_{\mathcal{L},\Sigma} \mathcal{U}_{\mathcal{T},C}, \rho_C$.*

Proof. By Lemmas 3 and 4, Condition 1 is equivalent to Condition 1 of the definition of MSCs. By Lemmas 2, 3, and 4, Condition 2 is equivalent to Condition 2 of the definition of MSCs. ❑

For an interpretation $\mathcal{I}$ and a $d_0 \in \Delta^\mathcal{I}$, a *$d_0$-path of length k in $\mathcal{I}$* is a sequence $d_0 r_0 \cdots r_{k-1} d_k$ with $(d_i, d_{i+1}) \in r_i^\mathcal{I}$ for all $i < k$, each r_i a (potentially inverse) role. Denote by $\mathsf{tail}(p)$ the last element of p. The *$\mathcal{ELI}, k$-unfolding of $\mathcal{I}$ at d_0*, denoted $(\mathcal{I}, d_0)^{\downarrow \mathcal{ELI}, k}$, is the interpretation defined by taking $\Delta^{(\mathcal{I}, d_0)^{\downarrow \mathcal{ELI}, k}}$ to be the set of all d_0-paths of length at most k and setting

$$\begin{aligned} A^{(\mathcal{I}, d_0)^{\downarrow \mathcal{ELI}, k}} &= \{p \mid \mathsf{tail}(p) \in A^\mathcal{I}\} \\ r^{(\mathcal{I}, d_0)^{\downarrow \mathcal{ELI}, k}} &= \{(p, prt) \mid prt \in \Delta^{(\mathcal{I}, a)^{\downarrow \mathcal{ELI}, k}}\} \cup \\ &\qquad \{(pr^- t, p) \mid prt \in \Delta^{(\mathcal{I}, a)^{\downarrow \mathcal{ELI}, k}}\}. \end{aligned}$$

The *$\mathcal{EL}, k$-unfolding of $\mathcal{I}$ at d_0*, denoted $(\mathcal{I}, d_0)^{\downarrow \mathcal{EL}, k}$, is defined accordingly, but only admitting role names in paths. For $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$ and an $\mathcal{L}$ KB $\mathcal{K}$, we use $(\Pi_{i=1}^n (\mathcal{U}_\mathcal{K}, d_i))^{\downarrow \mathcal{L}, k}_{|\Sigma}$ to denote the $\mathcal{L}, k$-unfolding of the Σ-reduct of $\Pi_{i=1}^n (\mathcal{U}_\mathcal{K}, d_i)$ at $(d_1, \dots, d_n)$. It can be verified that this interpretation is tree-shaped for $\mathcal{L} = \mathcal{ELI}$ and ditree-shaped for $\mathcal{L} = \mathcal{EL}$ and can thus be viewed as an $\mathcal{L}$ concept C_k.

Theorem 4 (MSC Existence) *Let $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$, $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{L}$ KB, $a_1, \dots, a_n \in \mathsf{ind}(\mathcal{A})$, and Σ a signature. The following are equivalent, for $C_k = (\Pi_{i=1}^n (\mathcal{U}_\mathcal{K}, a_i))^{\downarrow \mathcal{L}, k}_{|\Sigma}$:*

1. *the $\mathcal{L}(\Sigma)$-MSC of $a_1, \dots, a_n$ w.r.t. $\mathcal{K}$ exists;*
2. *C_k is the $\mathcal{L}(\Sigma)$-MSC of $a_1, \dots, a_n$ w.r.t. $\mathcal{K}$, for a $k \geq 0$;*
3. *$\Pi_{i=1}^n (\mathcal{U}_\mathcal{K}, a_i) \preceq_{\mathcal{L},\Sigma} (\mathcal{U}_{\mathcal{T},C_k}, \rho_{C_k})$ for some $k \geq 0$.*

Proof. "2 $\Rightarrow$ 1" is trivial. "3 $\Rightarrow$ 2" is an immediate consequence of Theorem 3. For "1 $\Rightarrow$ 3", let the $\mathcal{L}(\Sigma)$-MSC D be of depth k. It then follows from Theorem 3 that

$(a_1, \dots, a_n) \in D^{\prod_{i=1}^n \mathcal{U}_\mathcal{K}}$ which implies $\rho_{C_k} \in D^{\mathcal{U}_{\mathcal{T},C_k}}$. Now Point 3 follows from the definition of the MSC and Lemmas 2, 3, and 4. ❏

Note that Theorems 3 and 4 link MSC-verification and existence, as well as LCS-verification and existence (via Theorem 1) to product simulation problems. For $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$, the *$\mathcal{L}$-product simulation problem* is to decide given $(\mathcal{I}_1, d_1), \dots, (\mathcal{I}_n, d_n), (\mathcal{J}, e)$, whether $\prod_{i=1}^n (\mathcal{I}_i, d_i) \preceq_\mathcal{L} (\mathcal{J}, e)$. These are fundamental problems that have received attention in several areas such as verification and database theory (Harel, Kupferman, and Vardi 2002; Barceló and Romero 2017; ten Cate and Dalmau 2015).

4 Without TBoxes

We start with studying least general generalizations in the case without TBoxes, beginning with verification in $\mathcal{EL}$.

Theorem 5 *In $\mathcal{EL}$, LCS and MSC verification w.r.t. the empty TBox are* CONP*-complete. The lower bounds apply even when the signature is full.*

Proof. (sketch) The upper bound uses Theorem 3, the fact that instance checking in $\mathcal{EL}$ is in PTIME, and the observation that the $\mathcal{EL}$-product simulation problem is in CONP if the interpretation $\mathcal{J}$ is tree-shaped (here, it is even ditree-shaped). In fact, if $(\mathcal{I}, d) \not\preceq_{\mathcal{EL},\Sigma} (\mathcal{J}, e)$ with $\mathcal{J}$ tree-shaped, then there is a subinterpretation $\mathcal{I}_0$ of $\mathcal{I}$ of polynomial size such that $(\mathcal{I}_0, d) \not\preceq_{\mathcal{EL},\Sigma} (\mathcal{J}, e)$. The lower bound is proved by reducing the satisfiability problem for propositional logic to the complement of $\mathcal{EL}$-LCS verification. It also establishes CONP-hardness of the $\mathcal{EL}$-product simulation problem in the case that $\mathcal{J}$ is tree-shaped. ❏

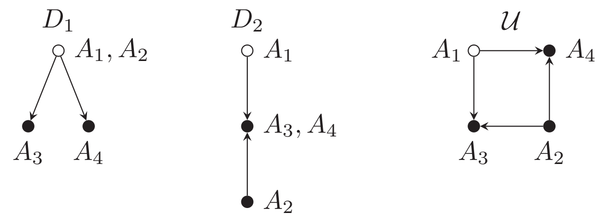
Regarding existence, a first well-known observation is that the $\mathcal{EL}$-LCS always exists, even if the signature is not full. This follows from Theorem 4 and the fact that if $\mathcal{K} = (\emptyset, \mathcal{A}_{C_1} \cup \dots \cup \mathcal{A}_{C_n})$ then the (reachable part of the) Σ-reduct of $\prod_{i=1}^n (\mathcal{U}_\mathcal{K}, \rho_{C_i})$ is ditree-shaped and coincides with $\Pi_{i=1}^n (\mathcal{U}_\mathcal{K}, \rho_{C_i})^{\downarrow \mathcal{EL},k}_{|\Sigma}$, k the maximum depth of $C_1, \dots, C_n$. In contrast, the $\mathcal{EL}$-MSC does not always exist even with the empty TBox, see Example 2.

Theorem 6 *In $\mathcal{EL}$, MSC existence w.r.t. the empty TBox is* PSPACE*-complete. The lower bound applies even when the signature is full.*

Proof. (sketch) Using Theorem 4, one can show that the $\mathcal{EL}(\Sigma)$-MSC of $a_1, \dots, a_n$ w.r.t. a KB $\mathcal{K} = (\emptyset, \mathcal{A})$ exists if and only if there is no infinite Σ-path in $\mathcal{A}^n = \prod_{i=1}^n \mathcal{A}$ that starts at $(a_1, \dots, a_n)$—we view ABoxes as finite interpretations here. We can thus decide existence of the $\mathcal{EL}(\Sigma)$-MSC in polynomial space in the standard way: guess an element a of $\mathcal{A}^n$ and, proceeding step by step, a path through $\mathcal{A}^n$ that starts at $(a_1, \dots, a_n)$ and follows only role names from Σ. Reject if the element a is seen twice. The lower bound is established by reducing the word problem of deterministic polynomially space-bounded Turing machines. ❏

We next turn to $\mathcal{ELI}$. In contrast to $\mathcal{EL}$, here the LCS does not always exist even when the TBox is empty.

Example 3 Consider the following $\mathcal{ELI}$ concepts D_1, D_2 over concept names $A_1, \dots, A_4$ and a single role r:



The interpretation $\mathcal{U}$ is the part of $\mathcal{A}_{D_1} \times \mathcal{A}_{D_2}$ that is reachable from its root $\circ$. One can show that the infinite path in $\mathcal{U}$ labeled with $(A_1, r, A_3, r^-, A_2, r, A_4, r^-)^\omega$ is not $\mathcal{ELI}$ simulated by $(\mathcal{U}^{\downarrow \mathcal{ELI},k}, \circ)$, for any $k \geq 0$. Thus, the $\mathcal{ELI}$-LCS of D_1, D_2 does not exist by Theorem 4.

The next theorem summarizes our results regarding $\mathcal{ELI}$.

Theorem 7 *In $\mathcal{ELI}$, LCS and MSC existence and verification w.r.t. the empty TBox are* PSPACE*-hard and in* EXPTIME*. The lower bounds apply when the signature is full.*

Proof. (sketch) The main ingredient to the PSPACE lower bounds is a rather intricate proof that the $\mathcal{ELI}$-product simulation problem is PSPACE-hard already when restricted to tree-shaped interpretations. In fact, this is the case even when interpretations on the left-hand sides are trees of depth two and the interpretation on the right-hand side is fixed (and of depth eleven). It is interesting to contrast this with the fact that the $\mathcal{EL}$-product simulation problem is CONP-complete on tree-shaped interpretations, see the proof of Theorem 5. To obtain a PSPACE lower bound for LCS verification and existence, we then use reductions from $\mathcal{ELI}$-product simulation on tree shaped interpretations.

The upper bound for MSC verification (and thus also for LCS verification) is obtained by recalling that $\mathcal{ELI}$ instance checking is EXPTIME-complete and adapting the EXPTIME upper bound from (Zarrieß and Turhan 2013) for the $\mathcal{EL}$-product simulation problem to $\mathcal{ELI}$.

The EXPTIME upper bound for MSC existence (and thus also for LCS existence) can be proved similarly to the upper bound in Theorem 6. The main difference is that we now work with $\mathcal{ELI}$ simulations rather than $\mathcal{EL}$ simulations and thus need to be more careful about the paths we consider. In fact, we use paths $d_0, r_0, d_1, r_1, d_2, \dots$ through $\mathcal{A}^n = \prod_{i=1}^n \mathcal{A}$ that start at $d_0 = (a_1, \dots, a_n)$, follow only Σ-roles, and satisfy the following for all $i \geq 0$: 1. if $r_i = r_{i+1}^-$, then $(\mathcal{A}^n, d_{i+2}) \not\preceq_{\mathcal{ELI},\Sigma} (\mathcal{A}^n, d_i)$; 2. there is no $e \neq d_{i+1}$ such that $r_i(d_i, e) \in \mathcal{A}^n$, $(\mathcal{A}^n, d_{i+1}) \preceq_{\mathcal{ELI},\Sigma} (\mathcal{A}^n, e)$, and $(\mathcal{A}^n, e) \not\preceq_{\mathcal{ELI},\Sigma} (\mathcal{A}^n, d_{i+1})$. ❏

All problems studied in this section are solvable in PTIME if the number of examples is bounded by a constant. This follows from an analysis of the presented upper bound proofs and has in some cases also been established before (Baader, Küsters, and Molitor 1999; Zarrieß and Turhan 2013).

5 With TBoxes

We now add TBoxes to the picture. It turns out that, in this case, we can transfer results from the concept separability problem, which has been considered in concept learning from positive and negative examples (Funk et al. 2019).

Definition 3 *Let $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$. An $\mathcal{L}$* learning instance *is a triple $(\mathcal{K}, P, N)$ with $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ an $\mathcal{L}$ KB and $P, N \subseteq$*

$\mathsf{ind}(\mathcal{A})$ *sets of* positive *and* negative examples. *Let* Σ *be a signature. An* $\mathcal{L}(\Sigma)$ solution *to* $(\mathcal{K}, P, N)$ *is an* $\mathcal{L}(\Sigma)$ *concept* C *such that* $\mathcal{K} \models C(a)$ *for all* $a \in P$ *and* $\mathcal{K} \not\models C(a)$ *for all* $a \in N$.

This definition gives rise to the decision problem of $\mathcal{L}$ *concept separability*: given an $\mathcal{L}$ learning instance $(\mathcal{K}, P, N)$ and a signature Σ, decide whether it admits an $\mathcal{L}(\Sigma)$ solution. As the conjunction of $\mathcal{L}(\Sigma)$ solutions to $(\mathcal{K}, P, \{b\})$, $b \in N$, is an $\mathcal{L}(\Sigma)$ solution to $(\mathcal{K}, P, N)$, it suffices to consider instances with N singleton. Note that in (Funk et al. 2019) only the full signature case is considered.

One can easily derive from (Funk et al. 2019) that $(\mathcal{K}, P, \{b\})$ has an $\mathcal{L}(\Sigma)$ solution iff $\prod_{a\in P}(\mathcal{U}_\mathcal{K}, a) \not\preceq_{\mathcal{L},\Sigma} (\mathcal{U}_\mathcal{K}, b)$. By encoding b as a concept D as in the proof of Theorem 2, we can thus view $\mathcal{L}(\Sigma)$ concept separability as the problem to decide for an $\mathcal{L}$ KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, examples $a_1, \dots, a_n \in \mathsf{ind}(\mathcal{A})$, and an $\mathcal{L}$ concept D whether $\prod_{i=1}^n(\mathcal{U}_\mathcal{K}, a_i) \not\preceq_{\mathcal{L},\Sigma} (\mathcal{U}_{\mathcal{T},D}, \rho_D)$, which is exactly the negation of Condition 2 of the characterization of MSC verification in Theorem 3. This provides the basis for the following.

Theorem 8 *For* $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$*, the complement of* $\mathcal{L}$ *concept separability can be reduced in polynomial time to* $\mathcal{L}$*-MSC verification and existence. This also holds for the full signature.*

Proof. (sketch) We consider $\mathcal{EL}$ and the full signature case. Given $\mathcal{K}$, $a_1, \dots, a_n$, and D, we extend $\mathcal{K}$ by adding assertions $v(\rho_i, a_i), v(\rho_i, b_i), D(b_i)$, where ρ_i and b_i are fresh individuals, v a fresh role name, and $D(b_i)$ stands for $\mathcal{A}_D$ rooted at b_i. Then $\prod_{i=1}^n(\mathcal{U}_\mathcal{K}, a_i) \not\preceq_{\mathcal{EL}} (\mathcal{U}_{\mathcal{T},D}, \rho_D)$ iff $\exists v.D$ is the $\mathcal{EL}$-MSC of $\rho_1, \dots, \rho_n$ w.r.t. the extended KB (under mild assumptions). For the reduction to MSC existence, we additionally generate infinite r-chains starting at a_i and b_i using CIs $X \sqsubseteq \exists r.X$ and adding $X(a_i)$ and $X(b_i)$ to the ABox, where the concept names X are distinct for distinct a_i but coincide for all b_i. If we assume w.l.o.g. that $n \geq 2$, then $\prod_{i=1}^n(\mathcal{U}_\mathcal{K}, a_i) \not\preceq_{\mathcal{EL}} (\mathcal{U}_{\mathcal{T},D}, \rho_D)$ iff the $\mathcal{EL}$-MSC of $\rho_1, \dots, \rho_n$ w.r.t. the extended KB exists. ❏

It is shown in (Funk et al. 2019) that $\mathcal{ELI}$ concept separability is undecidable already in the full signature case and even with only two positive examples. We thus obtain the following from Theorems 8 and 2 and the fact that the number of examples remains unchanged under the reductions.

Theorem 9 *In* $\mathcal{ELI}$*, MSC and LCS verification and existence are undecidable. This is already the case when the signature is full and there are at most two examples.*

It is also shown in (Funk et al. 2019) that $\mathcal{EL}$ concept separability is ExpTime-hard. In this case, the number of positive examples is not bounded by a constant.

Theorem 10 *In* $\mathcal{EL}$*, MSC and LCS verification and existence are* ExpTime*-complete. The lower bounds already apply when the signature is full.*

Proof. (sketch) The lower bounds come from Theorems 8 and 2. ExpTime upper bounds for LCS existence and verification with the full signature are in (Zarrieß and Turhan 2013), the former explicitly and the latter implicitly. They extend to other signatures in a straightforward way. To lift these bounds to the MSC, we use Theorem 2. ❏

When the number of examples is bounded, then all problems in Theorem 10 can be solved in PTime (which was known for LCS existence (Zarrieß and Turhan 2013)).

We close this section with observing that $\mathcal{L}$-MSC verification can be reduced to the complement of concept separability, and thus, by Theorem 8, to $\mathcal{L}$-MSC existence.

Theorem 11 *For* $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELI}\}$*,* $\mathcal{L}$*-MSC verification can be reduced in polynomial time to the complement of* $\mathcal{L}$ *concept separability. This also holds for the full signature.*

Proof. (sketch) Recall that Condition 2 of Theorem 3 is the complement of concept separability. By Lemmas 3 and 2, Condition 1 is equivalent to requiring $\mathcal{U}_{\mathcal{T},C}, \rho_C \preceq_\mathcal{L} \mathcal{U}_\mathcal{K}, a_i$, for all i. These simulation checks can be incorporated into Condition 2 by extending the ABox. ❏

6 Symmetry Free $\mathcal{ELI}$

An inspection of the proof of the undecidability results in Theorem 9 reveals that it crucially depends on the MSC and LCS to contain subconcepts of the form $\exists r.(C \sqcap \exists r^-.D)$. Indeed, concept separability is decidable when the TBox is formulated in $\mathcal{ELI}$ while separating concepts are restricted to $\mathcal{EL}$ (Funk et al. 2019). We consider a more general case by restricting the MSC and LCS to *symmetry free* $\mathcal{ELI}$ concepts ($\mathcal{ELI}^{\mathsf{sf}}$ *concepts* for short), that is, $\mathcal{ELI}$ concepts that do not contain such subconcepts. With $\mathcal{ELI}^{\mathsf{sf}}$*-LCS and MSC verification and existence w.r.t.* $\mathcal{ELI}$ *TBoxes*, we mean that the TBox is formulated in unrestricted $\mathcal{ELI}$ while least general generalizations are formulated in $\mathcal{ELI}^{\mathsf{sf}}$. For the LCS, also the examples are formulated in unrestricted $\mathcal{ELI}$.

We start with providing a characterization of $\mathcal{ELI}^{\mathsf{sf}}(\Sigma)$-MSC existence. To achieve this, we modify the notion of $\mathcal{ELI}$, k-unfolding of an interpretation $\mathcal{I}$ at a $d_0 \in \Delta^\mathcal{I}$ given in Section 3 by restricting the domain of the resulting interpretation to *symmetry free* d_0*-paths* of length k, that is, to d_0-paths $d_0 r_0 \cdots r_{m-1} d_m$, $m \leq k$, that satisfy $r_i \neq r_{i+1}^-$ for all $i < m$. We speak of the $\mathcal{ELI}^{\mathsf{sf}}$, k*-unfolding of* $\mathcal{I}$ *at* d_0, denoted $(\mathcal{I}, d_0)^{\downarrow \mathcal{ELI}^{\mathsf{sf}},k}$. We further use $(\mathcal{I}, d_0)^{\downarrow \mathcal{ELI}^{\mathsf{sf}}}$ to denote the unbounded $\mathcal{ELI}^{\mathsf{sf}}$-unfolding of $\mathcal{I}$ at d_0, that is, the union of all $(\mathcal{I}, d_0)^{\downarrow \mathcal{ELI}^{\mathsf{sf}},k}$, $k \geq 0$. Now let Σ be a signature. For an $\mathcal{ELI}$ KB $\mathcal{K}$, we use $(\Pi_{i=1}^n(\mathcal{U}_\mathcal{K}, d_i))_{|\Sigma}^{\downarrow \mathcal{ELI}^{\mathsf{sf}},k}$ to denote the $\mathcal{ELI}^{\mathsf{sf}}$, k-unfolding of the Σ-reduct of $\Pi_{i=1}^n(\mathcal{U}_\mathcal{K}, d_i)$ at $(d_1, \dots, d_n)$. As this interpretation is tree-shaped, it can be viewed as an $\mathcal{ELI}$ concept which is even an $\mathcal{ELI}^{\mathsf{sf}}$ concept.

Theorem 12 **($\mathcal{ELI}^{\mathsf{sf}}$-MSC Existence w.r.t. $\mathcal{ELI}$ TBoxes)** *Let* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *be an* $\mathcal{ELI}$ *KB,* $a_1, \dots, a_n \in \mathsf{ind}(\mathcal{A})$*, and* Σ *a signature. The following are equivalent, for* $C_k = (\Pi_{i=1}^n(\mathcal{U}_\mathcal{K}, a_i))_{|\Sigma}^{\downarrow \mathcal{ELI}^{\mathsf{sf}},k}$:

1. *the* $\mathcal{ELI}^{\mathsf{sf}}(\Sigma)$*-MSC of* $a_1, \dots, a_n$ *w.r.t.* $\mathcal{K}$ *exists;*
2. C_k *is the* $\mathcal{ELI}^{\mathsf{sf}}(\Sigma)$*-MSC of* $a_1, \dots, a_n$ *w.r.t.* $\mathcal{K}$*, for a* $k \geq 0$*;*
3. $\Pi_{i=1}^n(\mathcal{U}_\mathcal{K}, a_i)^{\downarrow \mathcal{ELI}^{\mathsf{sf}}} \preceq_{\mathcal{ELI},\Sigma} (\mathcal{U}_{\mathcal{T},C_k}, \rho_{C_k})$ *for a* $k \geq 0$.

Since Theorem 1 extends to the case considered in this section, Theorem 12 also yields a characterization for $\mathcal{ELI}^{\mathsf{sf}}$

LCS existence w.r.t. $\mathcal{ELI}$ TBoxes. Theorems 8 and 11 can also be adapted using a version of concept separability where the separating concepts are formulated in $\mathcal{ELI}^{\mathsf{sf}}$. Thus verification reduces to existence in polynomial time and we refrain from giving an explicit characterization.

Theorem 12 provides the basis for proving that symmetry freeness regains decidability.

Theorem 13 *$\mathcal{ELI}^{\mathsf{sf}}$-MSC and LCS existence and verification with respect to $\mathcal{ELI}$ TBoxes are* ExpTime*-complete. The lower bounds hold in the full signature case and with only one example.*

The lower bounds are easy to prove by reduction from the subsumption of concept names w.r.t. $\mathcal{ELI}$ TBoxes (Baader, Brandt, and Lutz 2008). For the upper bounds, we use an approach based on automata on infinite trees. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{ELI}$ KB, $a_1, \ldots, a_n \in \mathsf{ind}(\mathcal{A})$, and Σ a signature. Theorem 12 suggests to test emptiness of two tree automata $\mathfrak{A}$ and $\mathfrak{B}$ where $\mathfrak{A}$ accepts precisely the tree-shaped interpretations that admit an $\mathcal{ELI}(\Sigma)$ simulation from $\mathcal{U} := (\Pi_{i=1}^n(\mathcal{U}_\mathcal{K}, a_i))^{\downarrow\mathcal{ELI}^{\mathsf{sf}}}$ and $\mathfrak{B}$ accepts precisely the tree-shaped interpretations $\mathcal{U}_{\mathcal{T},C_k}, \rho_{C_k}$, $k \geq 0$. In particular, the automaton $\mathfrak{A}$ visits all elements of $\mathcal{U}$ using its states, assigning to each of them a simulating element in the input interpretation. Elements in $\mathcal{U}$ are represented by their type t and the role that led to it—note that these uniquely determine the successors, and that this is not the case without symmetry freeness. We thus have (at least) exponentially many states. To obtain an ExpTime upper bound, we therefore use non-deterministic tree automata (NTA) rather than alternating ones. To avoid having a state for every set of types, we must further make sure that every element in $\mathcal{U}$ is simulated by a different element in the input tree. To have enough room when moving down in the input tree, we slightly refine our characterization.

A simulation S from $\mathcal{I}_1$ to $\mathcal{I}_2$ is *injective* if for all $e \in \Delta^{\mathcal{I}_2}$, there is at most one $d \in \Delta^{\mathcal{I}_1}$ with $(d, e) \in S$. We write $(\mathcal{I}_1, d_1) \preceq^{\mathsf{in}}_{\mathcal{ELI},\Sigma} (\mathcal{I}_2, d_2)$ if there is an injective $\mathcal{ELI}(\Sigma)$-simulation from $\mathcal{I}_1$ to $\mathcal{I}_2$ that contains (d_1, d_2). Let $\mathcal{I}^{\times\ell}$ denote the interpretation that is obtained from a tree-shaped interpretation $\mathcal{I}$ by duplicating every successor in the tree so that it occurs ℓ times.

Lemma 5 *Let N be the outdegree of $\Pi_{i=1}^n\mathcal{U}_\mathcal{K}$. Then the $\mathcal{ELI}^{\mathsf{sf}}(\Sigma)$-MSC of $a_1, \ldots, a_n$ w.r.t. $\mathcal{K}$ exists iff, for some subconcept D of $(\Pi_{i=1}^n(\mathcal{U}_\mathcal{K}, a_i))^{\downarrow\mathcal{ELI}^{\mathsf{sf}}}_\Sigma$, we have:*

$$(\Pi_{i=1}^n(\mathcal{U}_\mathcal{K}, a_i))^{\downarrow\mathcal{ELI}^{\mathsf{sf}}} \preceq^{in}_{\mathcal{ELI},\Sigma} (\mathcal{U}^{\times N}_{\mathcal{T},D}, \rho_D).$$

Now, $\mathfrak{A}$ accepts the tree-shaped interpretations that admit *injective* $\mathcal{ELI}(\Sigma)$ simulations from $(\Pi_{i=1}^n(\mathcal{U}_\mathcal{K}, a_i))^{\downarrow\mathcal{ELI}^{\mathsf{sf}}}$ using exponentially many states. Further, $\mathfrak{B}$ accepts interpretations of the form $\mathcal{U}^{\times N}_{\mathcal{T},D}$ for some D as in the lemma. We first construct an automaton that works over pairs of tree-shaped interpretations and verifies that the first component represents a suitable D and the second component represents $\mathcal{U}_{\mathcal{T},D}$. We then project to the latter and modify the automaton so as to accept all $\mathcal{I}^{\times N}$ with $\mathcal{I}$ accepted before.

7 Single Example MSC

We consider the MSC of a single example, which is the case traditionally studied in the literature. A PTime upper bound for $\mathcal{EL}$ was given in (Zarrieß and Turhan 2013). We show that adding a signature does not affect this result, and that it also holds for verification.

Theorem 14 *In $\mathcal{EL}$, single example MSC existence and verification are in* PTime.

Proof. (sketch) This is a consequence of the proof of Theorem 13. Applying the constructions from that proof to an $\mathcal{EL}$ TBox instead of an $\mathcal{ELI}$ TBox has two effects: first, all involved automata can be constructed in polynomial time and are of polynomial size; and second Theorem 12 implies that if the $\mathcal{ELI}^{\mathsf{sf}}$-MSC exists, it is actually an $\mathcal{EL}$ concept. ❑

We next show that the $\mathcal{ELI}$ case is dramatically different. In particular, the complexity is much higher and admitting non-full signatures causes an exponential jump in complexity.

Theorem 15 *In $\mathcal{ELI}$, single example MSC existence and verification are* 2ExpTime*-complete in general and* ExpTime*-complete when the signature is full.*

Proof. (sketch) In the full signature case, the lower bound is by reduction from the subsumption of concept names w.r.t. $\mathcal{ELI}$ TBoxes. For unrestricted signatures, we reduce the complement of single example $\mathcal{ELI}$ concept separability, shown 2ExpTime-hard in (Gutiérrez-Basulto, Jung, and Sabellek 2018), similar to the proof of Theorem 8.

The upper bounds are shown using an automata based approach that is in spirit similar to the approach taken in Section 6. The main difference is that the automaton $\mathfrak{A}$ has to be two-way since it checks for $\mathcal{ELI}$ simulations from $\mathcal{U}_\mathcal{K}, a$. In case of restricted signature, it has to store types in its states, while for the full signature ABox individuals suffice. ❑

8 Discussion

We have analyzed the complexity of LCS and MSC verification and existence in the DLs $\mathcal{EL}$ and $\mathcal{ELI}$, obtaining various complexity results and establishing a close link to concept separability. Topics for future research include tight bounds on the size of the LCS and MSC and studying cases in which the TBoxes is formulated in an expressive DL such as $\mathcal{ALC}$ while the LCS and MSC are formulated in $\mathcal{EL}$ or $\mathcal{ELI}$ (to avoid overfitting). It would also be interesting to study DLs that admit role constraints such as transitive roles and expressive forms of role inclusion. Finally, it would be of interest to study the data complexity, under which the TBox is not regarded as part of the input.

Acknowledgments. Carsten Lutz was supported by the DFG CRC EASE. Frank Wolter was partially supported by EPSRC grant EP/S032207/1.

References

Baader, F., and Küsters, R. 1998. Computing the least common subsumer and the most specific concept in the presence of cyclic aln-concept descriptions. In *Proc. of KI*, 129–140. Springer.

Baader, F.; Horrocks, I.; Lutz, C.; and Sattler, U. 2017. *An Introduction to Description Logic*. Cambridge University Press.

Baader, F.; Brandt, S.; and Lutz, C. 2008. Pushing the $\mathcal{EL}$ envelope further. In *Proc. of OWLED workshop*.

Baader, F.; Küsters, R.; and Molitor, R. 1999. Computing least common subsumers in description logics with existential restrictions. In *Proc. of IJCAI*, 96–103.

Baader, F.; Sertkaya, B.; and Turhan, A. 2007. Computing the least common subsumer w.r.t. a background terminology. *J. Applied Logic* 5(3):392–420.

Baader, F. 2003. Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In *Proc. of IJCAI*, 319–324.

Badea, L., and Nienhuys-Cheng, S. 2000. A refinement operator for description logics. In *Proc. of ILP*, 40–59.

Barceló, P., and Romero, M. 2017. The complexity of reverse engineering problems for conjunctive queries. In *Proc. of ICDT*, 7:1–7:17.

Borgida, A.; Toman, D.; and Weddell, G. E. 2016. On referring expressions in query answering over first order knowledge bases. In *Proc. of KR*, 319–328.

Bühmann, L.; Lehmann, J.; Westphal, P.; and Bin, S. 2018. DL-learner structured machine learning on semantic web data. In *Proc. of WWW*, 467–471.

Cohen, W. W.; Borgida, A.; and Hirsh, H. 1992. Computing least common subsumers in description logics. In *Proc. of AAAI*, 754–760.

Colucci, S.; Donini, F. M.; Giannini, S.; and Sciascio, E. D. 2016. Defining and computing least common subsumers in RDF. *J. Web Semant.* 39:62–80.

Donini, F. M.; Colucci, S.; Noia, T. D.; and Sciascio, E. D. 2009. A tableaux-based method for computing least common subsumers for expressive description logics. In *Proc. of IJCAI*, 739–745.

Eppe, M.; Maclean, E.; Confalonieri, R.; Kutz, O.; Schorlemmer, M.; Plaza, E.; and Kühnberger, K. 2018. A computational framework for conceptual blending. *Artif. Intell.* 256:105–129.

Fauconnier, G., and Turner, M. 2008. *The way we think: Conceptual blending and the mind's hidden complexities*. Basic Books.

Funk, M.; Jung, J. C.; Lutz, C.; Pulcini, H.; and Wolter, F. 2019. Learning description logic concepts: When can positive and negative examples be separated. In *Proc. of IJCAI*.

Gutiérrez-Basulto, V.; Jung, J. C.; and Sabellek, L. 2018. Reverse engineering queries in ontology-enriched systems: The case of expressive horn description logic ontologies. In *Proc. of IJCAI-ECAI*.

Harel, D.; Kupferman, O.; and Vardi, M. Y. 2002. On the complexity of verifying concurrent transition systems. *Inf. Comput.* 173(2):143–161.

Küsters, R., and Borgida, A. 2001. What's in an attribute? consequences for the least common subsumer. *J. Artif. Intell. Res.* 14:167–203.

Küsters, R., and Molitor, R. 2001. Approximating most specific concepts in description logics with existential restrictions. In *Proc. of KI*, 33–47.

Lehmann, J., and Haase, C. 2009. Ideal downward refinement in the $\mathcal{EL}$ description logic. In *Proc. of ILP*, 73–87.

Lehmann, J., and Hitzler, P. 2010. Concept learning in description logics using refine-ment operators. *Machine Learning* 78:203–250.

Lisi, F. A. 2012. A formal characterization of concept learning in description logics. In *Proc. of DL*.

Muggleton, S. 1991. Inductive logic programming. *New Generation Comput.* 8(4):295–318.

Nebel, B. 1990. *Reasoning and Revision in Hybrid Representation Systems*. Springer.

Plotkin, G. 1970. A note on inductive generalizations. *Edinburgh University Press*.

Sarker, M. K., and Hitzler, P. 2019. Efficient concept induction for description logics. In *Proc. of AAAI*, 3036–3043.

ten Cate, B., and Dalmau, V. 2015. The product homomorphism problem and applications. In *Proc. of ICDT*, 161–176.

Zarrieß, B., and Turhan, A. 2013. Most specific generalizations w.r.t. general $\mathcal{EL}$-TBoxes. In *Proc. of IJCAI*, 1191–1197.