Least General Generalizations in Description Logic: Verification and Existence

Jean Christoph Jung, Carsten Lutz, Frank Wolter

1 \{jeanjung,clu\}@uni-bremen.de, University of Bremen, Germany
2 wolter@liverpool.ac.uk, University of Liverpool, United Kingdom

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 Hitzer 2010; Lehmann and
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 $\text{CONP}$-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 $\text{PSPACE}$-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 $\mathcal{EL}$ is not guaranteed to exist. We prove that LCS and MSC existence and verification are $\text{PSPACE}$-hard and in $\text{ExpTIME}$. 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 $\text{PSPACE}$-hard.

We then switch to the case with TBoxes, starting with observing a connection to concept learning (Baader et al. 2017). Let $\mathcal{N}_C$ be a set of concept names and $\mathcal{N}_R$ 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

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

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) $C \subseteq D$, where $C$ and $D$ are $\mathcal{L}$ concepts. Let $\mathcal{N}$ 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 $a \in \mathcal{N}_C$, $r \in \mathcal{N}_R$, and $a, b \in \mathcal{N}$. We often use $r(a,b)$ to denote $r^-b(a)$ if $r$ is an inverse role. We use $\text{ind}(A)$ to denote the set of all individual names that occur in $A$. 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 $\mathcal{I} = (\Delta^\mathcal{I}, \mathcal{I}^\mathcal{I})$, where $\Delta^\mathcal{I}$ is a non-empty set and $\mathcal{I}^\mathcal{I}$ maps each concept name $A \in \mathcal{N}_C$ to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$ and each role name $r \in \mathcal{N}_R$ to a binary relation $r^\mathcal{I}$ on $\Delta^\mathcal{I}$. We refer to (Baader et al. 2017) for details on how to extend $\mathcal{I}$ to compound concepts. An interpretation $\mathcal{I}$ satisfies a CI $C \subseteq D$ 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 $C \subseteq D$ is a consequence of the TBox $\mathcal{T}$, in symbols $\mathcal{I} \models C \subseteq D$, 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 $a \in \text{ind}(A)$, 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 $a \in \text{ind}(A)$, and an $\mathcal{L}$ concept $C$, whether $\mathcal{K} \models C(a)$.

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

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 $G_\mathcal{I}^\mathcal{I}$. We thus apply graph theoretic terminology directly to interpretations, speaking for example about their
Lemma 1 For all ELI TBoxes T and ELI concepts C, D, T \models C \subseteq D iff (T, A_C) \models D(\rho_C).

We introduce simulations, universal models, and direct products. Let I_1 and I_2 be interpretations. A relation S \subseteq \Delta^{i_1} \times \Delta^{i_2} is an \( \mathcal{L}(\Sigma) \) simulation from I_1 to I_2 if for all d, d' \in \Delta^{i_1} and e \in \Delta^{i_2}:

1. d \in d_1^{i_1} and (d, e) \in S imply e \in \Delta^{i_2}, for all A \in \Sigma;

2. (d, d') \in \Delta^{i_1} and (d, e) \in S imply (d', e') \in S and (e, e') \in \Delta^{i_2}, for some e' \in \Delta^{i_2}, for all role names r in S.

S is an \( \mathcal{L}(\Sigma) \) simulation if Condition 2 also holds for inverse roles r^{-1} with r \in S. Let L \in \{\mathcal{L}, \mathcal{ELI}\} and (d, e) \in \Delta^{i_1} \times \Delta^{i_2}. We write \((I_1, d) \leq_{\mathcal{L}(\Sigma)} (I_2, e)\) if there exists an \( \mathcal{L}(\Sigma) \) simulation from I_1 to I_2 that contains (d, e). We omit \Sigma if it is the full signature \( \mathcal{N}_C \cup \mathcal{N}_E \), writing \( \leq_{\mathcal{L}} \) and speaking of \( \mathcal{L} \) simulations. It can be checked in polynomial time whether \((I_1, d) \leq_{\mathcal{L}(\Sigma)} (I_2, e)\). The following lemma shows that \( \mathcal{L}(\Sigma) \) simulations characterize preservation of \( \mathcal{L}(\Sigma) \) concepts.

Lemma 2 Let L \in \{\mathcal{L}, \mathcal{ELI}\}, let I_1, I_2 be interpretations with finite outdegree, and let \Sigma be a signature. The following are equivalent:

1. \((I_1, d) \leq_{\mathcal{L}(\Sigma)} (I_2, e)\);

2. for all \( \mathcal{L}(\Sigma) \) concepts C: if d \in C^{i_1}, then e \in C^{i_2}.

Let K = (T, A) be a KB and sub(T) be the set of all subconcepts of concepts that occur in T. A type for T is a subset \( t \subseteq \text{sub}(T) \) such that T \models T \models \top t \subseteq D implies D \models t \ for all D \subseteq \text{sub}(T). Denote by \( T \) the set of all types for T. When a \in \text{ind}(A), t \in T, and r \in A, we write

- \( a \sim^r_r t \) if \( K \models \exists r. \sqcap \text{t}(a) \) and t is maximal with this condition, and

- \( t \sim^T t' \) if \( T \models \sqcap r. \sqcap (a. t \land t') \) and t' is maximal with this condition.

A path p for K is a sequence \( a_0 t_1 \cdots r_{n-1} t_n \) such that a in \text{ind}(A), r_0, \ldots, r_{n-1} are roles, t_i, t_{i+1} \in T, a \sim^{r_i}_{r_{i+1}} t_i, and t_i \sim^T_{r_{i+1}} t_{i+1} for all i < n. Let tail(p) denote the last element of the path p. Define the universal model \( U_K \) of K by taking \( \Delta^{i_1 \Sigma} \) the set of all paths for K and setting for all concept names A and role names r:

\[ A^{i_1 \Sigma} = a \in \text{ind}(A) \mid T, A \models A(a) \cup \{ p \in \Delta^{i_1 \Sigma} \mid \text{tail}(p) \} \]

\[ i^{i_1 \Sigma} = \{(a, b) \in \text{ind}(A)^2 \mid r(a, b) \in A \cup \{(p, prt) \mid (p, rt) \in \Delta^{i_1 \Sigma} \cup \{(pr^{-1}t, p) \mid pr^{-1}t \in \Delta^{i_1 \Sigma} \} \}

The universal model \( U_K \) of an ELI TBox T and an ELI concept C is defined as \( U_{T, C} \) where \( K = (T, A_C) \).
Like the LCS, the MSC is unique up to equivalence w.r.t. $T$ (if it exists) and thus we speak of the MSC. We drop $\Sigma$ if $\Sigma \supseteq \text{sig}(K)$. 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 $A = \{A(a), r(a, a)\}$ the $\mathcal{EL}$-MSC of $A$ w.r.t. $K = \{\emptyset, A\}$ does not exist (use that $K \models \exists^2 r. T(a)$ for all $n \geq 0$). In contrast, the $\mathcal{EL}$-MSC of a w.r.t. $K'$ is $\{A \subseteq \exists r . A\}$. Then the $\mathcal{EL}$-MSC of a alone w.r.t. $K$ does not exist, and likewise for $b$. In contrast, the $\mathcal{EL}$-MSC of $a, b$ w.r.t. $K$ 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 Miliot 1999). It pays off, however, to directly apply the MSC to multiple individuals. Let, for example, $K = \{\emptyset, A\}$, $A = \{A(a), r(a, a), A(b), s(b, b)\}$. Then the $\mathcal{EL}$-MSC of $a$ alone w.r.t. $K$ does not exist, and likewise for $b$. In contrast, the $\mathcal{EL}$-MSC of $a, b$ w.r.t. $K$ is $A$.

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

**Theorem 1** Let $L \in \{\mathcal{EL}, \mathcal{EL}^+\}$, $T$ be an $L$ TBox, $C_1, \ldots, C_n$ $L$ concepts, and $\Sigma$ a signature. Then an $L(\Sigma)$ concept $D$ is the $L(\Sigma)$-LCS of $C_1, \ldots, C_n$ w.r.t. $T$ iff $D$ is the $L(\Sigma)$-MSC of $\rho_1, \ldots, \rho_n$ w.r.t. the $\mathcal{KB}$ $(T, A)$, $A = A(C_1) \cup \ldots \cup A(C_n)$. LCS and MSC give rise to the four decision problems studied in this paper. Let $L$ be a description logic. $L$-LCS existence w.r.t. $T$Boxes means to decide, given $L$ concepts $C_1, \ldots, C_n$, an $L$ TBox $T$, and a finite signature $\Sigma$, whether the $L(\Sigma)$-LCS of $C_1, \ldots, C_n$ w.r.t. $T$ exists. By the remark made after Definition 1, it is without loss of generality to consider only finite signatures. In particular, we can use $\text{sig}(T \cup \{C_1, \ldots, C_n\})$ instead of the full signature. $L$-MSC existence w.r.t. $T$Boxes is defined accordingly, the input consisting of a $\mathcal{KB}$ $(T, A)$ with $T$ an $L$ TBox, $a_1, \ldots, a_n \in \text{ind}(A)$, and a finite signature $\Sigma$. In $L$-LCS (resp. $L$-MSC) verification w.r.t. $T$Boxes, we are given as an additional input a candidate $L(\Sigma)$ concept $C$ and the question is whether $C$ is the $L(\Sigma)$-LCS of $C_1, \ldots, C_n$ w.r.t. $T$ (resp. the $L(\Sigma)$-MSC of $a_1, \ldots, a_n$ w.r.t. $K$).

Theorem 1 provides a reduction from $L$-LCS existence w.r.t. $T$Boxes to $L$-MSC existence w.r.t. $T$Boxes, 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 $L \in \{\mathcal{EL}, \mathcal{EL}^+\}$. Then $L$-MSC verification (resp. existence) w.r.t. $T$Boxes can be reduced in polynomial time to $L$-LCS verification (resp. existence). This also holds in the full signature case if there are at least two examples.

**Proof.** Let $T$ be an $L$ TBox, $A$ an ABox, $a_1, \ldots, a_n \in \text{ind}(A)$. We may assume w.l.o.g. that $A$ is the disjoint union of ABoxes $A_1, \ldots, A_n$ such that $a_i \in \text{ind}(A_i)$ for $i = 1, \ldots, n$. Let $X_a$ be a fresh concept name for every $a \in \text{ind}(A)$ and let $T'$ be the extension of $T$ with $X_a \subseteq A$ for all $A(a) \in A$, $X_a \subseteq \exists r . X_{a'}$ for all $r(a, a') \in A$.

(If $L = \mathcal{EL}$, then also add $X_a \subseteq \exists r^- . X_{a'}$ if $r(a', a) \in A_0$.) Then for every signature $\Sigma$ that does not contain $\{X_{a_1}, \ldots, X_{a_n}\}$ and every $L(\Sigma)$ concept $D$, $D$ is the $L(\Sigma)$-MSC of $a_1, \ldots, a_n$ w.r.t. $(T', A)$ iff $D$ is the $L(\Sigma)$-LCS of $X_{a_1}, \ldots, X_{a_n}$ w.r.t. $T'$.

In the case of the full signature, we have to consider the $L(\Sigma) \cup \{X_{a_1}, \ldots, X_{a_n}\}$-LCS in place of the $L(\Sigma)$-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 procedures 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 $L \in \{\mathcal{EL}, \mathcal{EL}^+\}$, $K = (T, A)$ be an $L$ KB, $a_1, \ldots, a_n \in \text{ind}(A)$, and $\Sigma$ a signature. An $L(\Sigma)$ concept $C$ is the $L(\Sigma)$-MSC of $a_1, \ldots, a_n$ w.r.t. $K$ iff the following conditions hold:

1. $\{a_1, \ldots, a_n\} \in C \cap \Pi_{i=1}^n (U_{\Sigma}, a_i)$.
2. $\Pi_{i=1}^n (U_{\Sigma}, a_i) \leq L(\Sigma) U_{\Sigma}, C, \rho_C$.

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

$$A((I, d_0)^{L(\Sigma), k}) = \{p \mid \text{tail}(p) \in A^T\}$$

$$\rho_{(I, d_0)^{L(\Sigma), k}} = \{(p, prt) \mid \text{prt} \in \Delta(I, a)^{L(\Sigma), k}\} \cup \{(pr^T, p) \mid p \in \Delta(I, a)^{L(\Sigma), k}\}.$$
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 $(a_1, \ldots, a_n) \in D_{\Pi_{i=1}^k} \mathcal{U}_k$ which implies $\rho_{C_k} \in D_{\Pi_{\rho \in \mathcal{R}_k} r_{\rho,k}}$. 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 $\mathcal{E} \in \{\mathcal{E}, \mathcal{E}_{\mathcal{L}}\}$, the $\mathcal{L}$-product simulation problem is to decide given $(I_1, d_1), \ldots, (I_n, d_n), (J, e)$, whether $\prod_{i=1}^n (I_i, d_i) \preceq_{\mathcal{L}} (J, e)$. 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 Çate and Dalmau 2015).

4 Without TBoxes

We start with studying least general generalizations in the case without TBoxes, beginning with verification in $\mathcal{E}$. Theorem 5 In $\mathcal{E}$, 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 $\mathcal{E}$ is in PTIME, and the observation that the $\mathcal{E}$-LCS-product simulation problem is in coNP if the interpretation $J$ is tree-shaped (here, it is even ditee-shaped). In fact, if $(I, d) \not\preceq_{\mathcal{E}, \Sigma} (J, e)$ with $J$-tree-shaped, then there is a subinterpretation $I_0$ of $I$ of polynomial size such that $(I_0, d) \not\preceq_{\Sigma} (J, e)$. The lower bound is proved by reducing the satisfiability problem for propositional logic to the complement of $\mathcal{E}$-LCS verification. It also establishes coNP-hardness of the $\mathcal{E}$-LCS-product simulation problem in the case that $J$ is tree-shaped.

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

Theorem 6 In $\mathcal{E}$, 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 $\mathcal{E}(\Sigma)$-MSC of $a_1, \ldots, a_n$ w.r.t. a KB $K = (\emptyset, A)$ exists if and only if there is no infinite $\Sigma$-path in $A^n = \prod_{i=1}^n A$ that starts at $(a_1, \ldots, a_n)$—we view $A$-boxes as finite interpretations here. We can thus decide existence of the $\mathcal{E}(\Sigma)$-MSC in polynomial space in the standard way: guess an element $a$ of $A^n$ and, proceeding step by step, a path through $A^n$ that starts at $(a_1, \ldots, a_n)$ and follows only role names from $\Sigma$. 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 $\mathcal{E}_{\mathcal{L}}$. In contrast to $\mathcal{E}$, here the LCS does not always exist even when the TBox is empty.

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

![Diagrams]

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

The next theorem summarizes our results regarding $\mathcal{E}_{\mathcal{L}}$.

Theorem 7 In $\mathcal{E}_{\mathcal{L}}$, 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 $\mathcal{E}_{\mathcal{L}}$-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 $\mathcal{E}$-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 $\mathcal{E}_{\mathcal{L}}$-product simulation on tree shaped interpretations.

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

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 $\mathcal{E}_{\mathcal{L}}$ simulations rather than $\mathcal{E}$ simulations and thus need to be more careful about the paths we consider. In fact, we use paths $d_0, d_0, d_1, d_2, \ldots$ through $A^n = \prod_{i=1}^n A$ that start at $d_0 = (a_1, \ldots, a_n)$, follow only $\Sigma$-roles, and satisfy the following for all $i \geq 0$: 1. if $r_i = r_{i+1}$, then $(A^n, d_{i+2}) \not\preceq_{\mathcal{E}_{\mathcal{L}}, \Sigma} (A^n, d_i)$; 2. there is no $e \neq d_{i+1}$ such that $r_i(d_i, e) \in A^n$, $(A^n, d_{i+1}) \preceq_{\mathcal{E}_{\mathcal{L}}, \Sigma} (A^n, e)$, and $(A^n, e) \not\preceq_{\mathcal{E}_{\mathcal{L}}, \Sigma} (A^n, d_{i+1})$.

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 separabil-
ity 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 \((K, P, N)\) with \( K = (T, A) \) an \( \mathcal{L} \) KB and \( P, N \subseteq \text{ind}(A) \) sets of positive and negative examples. Let \( \Sigma \) be a signature. An \( \mathcal{L}(\Sigma) \) solution to \((K, P, N)\) is an \( \mathcal{L}(\Sigma) \) concept \( C\) such that \( K \models C(\alpha) \) for all \( \alpha \in P \) and \( K \not\models C(\alpha) \) for all \( \alpha \in N \).

This definition gives rise to the decision problem of \( \mathcal{L} \) concept separability: given an \( \mathcal{L} \) learning instance \((K, P, \{b\})\) and a signature \( \Sigma \), decide whether it admits an \( \mathcal{L}(\Sigma) \) solution. As the conjunction of \( \mathcal{L}(\Sigma) \) solutions to \((K, P, \{b\})\), \( b \in N \), is an \( \mathcal{L}(\Sigma) \) solution to \((K, P, N)\), 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 \((K, P, \{b\}) \) has an \( \mathcal{L}(\Sigma) \) solution iff \( \Pi_{a \in P} (U_K, a) \not\subseteq \Sigma \) \( U_K, b \). 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 \( K = (T, A) \), examples \( a_1, \ldots, a_n \in \text{ind}(A) \), and an \( \mathcal{L} \) concept \( D \) whether \( \Pi_{i=1}^n (U_K, a_i) \not\subseteq \Sigma \) \( U_K, D, \rho_D \), 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 \( K, a_1, \ldots, a_n, \) and \( D \), we extend \( K \) by adding assertions \( v(\pi_1, a_1), v(\pi_1, b_1), D(b_1) \), where \( \pi_1 \) and \( b_1 \) are fresh individuals, \( v \) a fresh role name, and \( D(b_1) \) stands for \( A_D \) rooted at \( b_1 \). Then \( \Pi_{i=1}^n (U_K, a_i) \not\subseteq \Sigma \) \( U_K, D, \rho_D \) iff \( \exists v, D \) is the \( \mathcal{EL} \)-MSC of \( \pi_1, \ldots, \pi_n \) w.r.t. the extended KB (under mild assumptions). For the reduction to \( \mathcal{L} \)MSC existence, we additionally generate infinite \( \tau \)-chains starting at \( a_i \) and \( b_i \) using CIs \( X \subseteq \exists \tau, X \) adding \( X(a_i) \) and \( X(b_i) \) to the ABox, where the concept names \( X \) are distinct for all \( a_i \) but coincide for all \( b_i \). If we assume w.l.o.g. that \( n \geq 2 \), then \( \Pi_{i=1}^n (U_K, a_i) \not\subseteq \Sigma \) \( U_K, D, \rho_D \) iff the \( \mathcal{EL} \)-MSC of \( \pi_1, \ldots, \pi_n \) w.r.t. the extended KB exists.

It is shown in (Funk et al. 2019) that \( \mathcal{EL} \) 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 \( \text{ExpTime} \)-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 \( \text{ExpTime} \)-complete. The lower bounds already apply when the signature is full.

**Proof.** (sketch) The lower bounds come from Theorems 8 and 2. \( \text{ExpTime} \) 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 \( P\text{TIME} \) (which was known for LCS existence (Zarrieß and Turhan 2013)).

We close this section with observing that \( \mathcal{EL} \)-MSC verification can be reduced to the complement of concept separability, and thus, by Theorem 8, to \( \mathcal{EL} \)-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 \( \forall r, c, \rho_C \subseteq \Sigma \) \( U_K, a_i \) for all \( j \). 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 \( \exists \tau (C \cap \exists \tau^{-}. D) \). 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 (\( \mathcal{ELI} \) concepts for short), that is, \( \mathcal{ELI} \) concepts that do not contain such subconcepts. With \( \mathcal{ELI} \)-LCS and MSC verification and existence w.r.t. \( \mathcal{ELI} \) TBoxes, we mean that the TBox is formulated in \( \mathcal{ELI} \) while we seek a least general generalization formulated in \( \mathcal{ELI} \). In the case of the LCS, also the examples are formulated in unrestricted \( \mathcal{ELI} \).

We start with providing a characterization of \( \mathcal{ELI} \) (\( \Sigma \))-MSC existence. To achieve this, we modify the notion of \( \mathcal{ELI} \), \( k \)-unfolding of an interpretation \( I \) at a \( d_0 \in \Delta^I \) 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 \( i \leq m \). We speak of the \( \mathcal{ELI} \)-\( k \)-unfolding of \( I \) at \( d_0 \), denoted \( (I, d_0) \) \( \mathcal{ELI} \)-\( k \). We further use \( (I, d_0) \) \( \mathcal{ELI} \)-\( k \) to denote the unbounded \( \mathcal{ELI} \)-\( k \)-unfolding of \( I \) at \( d_0 \), that is, the union of all \( (I, d_0) \) \( \mathcal{ELI} \)-\( k \) for \( k \geq 0 \). Now let \( \Sigma \) be a signature.

For an \( \mathcal{ELI} \) KB \( K \), we use \( (\Pi_{i=1}^n (U_K, d_i)) \) \( \mathcal{ELI} \)-\( k \) to denote the \( \mathcal{ELI} \)-\( k \)-unfolding of the \( \Sigma \)-reduct of \( \Pi_{i=1}^n (U_K, d_i) \) at \( (d_1, \ldots, d_n) \). As this interpretation is tree-shaped, it can be viewed as an \( \mathcal{ELI} \) concept which is even an \( \mathcal{ELI} \) concept.

**Theorem 12** (\( \mathcal{ELI} \)-MSC Existence w.r.t. \( \mathcal{ELI} \) TBoxes)

Let \( K = (T, A) \) be an \( \mathcal{ELI} \) KB, \( a_1, \ldots, a_n \in \text{ind}(A), \) and \( \Sigma \) a signature. The following are equivalent, for \( C_k = (\Pi_{i=1}^n (U_K, a_i)) \) \( \mathcal{ELI} \)-\( k \) :

1. the \( \mathcal{ELI} \) (\( \Sigma \))-MSC of \( a_1, \ldots, a_n \) w.r.t. \( K \) exists;
2. $C_k$ is the $ELT^d(\Sigma)$-MSC of $a_1, \ldots, a_n$ w.r.t. $K$, for a $k \geq 0$;
3. $\Pi^n_{i=1}(U_{K_1}, a_1)^{ELT^d} \preceq_{ELT, \Sigma} (U_{T, C_k}, \rho C_k)$ for a $k \geq 0$.

Since Theorem 1 extends to the case considered in this section, Theorem 12 also yields a characterization for $ELT^d$ LCS existence w.r.t. $ELT$ TBoxes. Theorems 8 and 11 can also be adapted using a version of concept separability where the separating concepts are formulated in $ELT^d$. 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** $ELT^d$-MSC and LCS existence and verification with respect to $ELT$ 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. $ELT$ TBoxes (Baader, Brandt, and Lutz 2008). For the upper bounds, we use an approach based on automata on infinite trees. Let $K = (T, A)$ be an $ELT$ KB, $a_1, \ldots, a_n \in \text{ind}(A)$, and $\Sigma$ a signature. Theorem 12 suggests to test emptiness of two tree automata $A$ and $B$ where $A$ accepts precisely the tree-shaped interpretations that admit an $ELT(\Sigma)$ simulation from $U := (\Pi^n_{i=1}(U_{K_1}, a_1))^{ELT^d}$ and $B$ accepts precisely the tree-shaped interpretations $U_{T, C_k}$, $\rho C_k$, $k \geq 0$. In particular, the automaton $A$ visits all elements of $U$ using its states, assigning to each of them a simulating element in the input interpretation. Elements in $U$ 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 $U$ 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 $I_1$ to $I_2$ is injective if for all $e \in \Delta^2$, there is at most one $d \in \Delta^2$ with $(d, e) \in S$. We write $(I_1, d_1) \preceq^\text{in}_{ELT, \Sigma} (I_2, d_2)$ if there is an injective $ELT(\Sigma)$-simulation from $I_1$ to $I_2$ that contains $(d_1, d_2)$. Let $I^{\times t}$ denote the interpretation that is obtained from a tree-shaped interpretation $I$ by duplicating every successor in the tree so that it occurs $t$ times.

**Lemma 5** Let $N$ be the outdegree of $\Pi^n_{i=1}(U_{K_1})$. Then the $ELT^d(\Sigma)$-MSC of $a_1, \ldots, a_n$ w.r.t. $K$ exists iff for some subconcept $D$ of $(\Pi^n_{i=1}(U_{K_1}, a_1))^{ELT^d}$, we have:

$$(\Pi^n_{i=1}(U_{K_1}, a_1))^{ELT^d} \preceq^\text{in}_{ELT, \Sigma} (U_{T, D}, \rho D).$$

Now, $A$ accepts the tree-shaped interpretations that admit injective $ELT(\Sigma)$ simulations from $(\Pi^n_{i=1}(U_{K_1}, a_1))^{ELT^d}$ using exponentially many states. Further, $B$ accepts interpretations of the form $U_{T, D}^N$ 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 $U_{T, D}$. We then project to the latter and modify the automaton so as to accept all $I^{\times N}$ with $I$ 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 $\mathcal{E}$ was given in (Zarríeß and Turhan 2013). We show that adding a signature does not affect this result, and that it also holds for verification.

**Theorem 14** In $\mathcal{E}$, 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 $\mathcal{E}$ TBox instead of an $ELT$ 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 $ELT^d$-MSC exists, it is actually an $ELT$ concept.

We next show that the $\mathcal{E}$ 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 $\mathcal{E}_I$, single example MSC existence and verification are 2-ExpTime-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. $ELT$ TBoxes. For unrestricted signatures, we reduce the complement of single example $ELT$ concept separability, shown 2-ExpTime-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 $A$ has to be two-way since it checks for $ELT$ simulations from $U_{K_1}$, $a$. 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 $ELT$ and $ELT^d$, 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 are formulated in an expressive DL such as $A\mathcal{C}$ while the LCS and MSC are formulated in $\mathcal{E}$ or $ELT$ (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.; 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.

Bühlmann, 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 Sciassio, 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 Sciassio, 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 Künnberger, 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 Sabellec, 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.

Jung, J.; Lutz, C.; Martel, M.; and Schneider, T. 2017. Query conservative extensions in Horn description logics with inverse roles. In Proc. of IJCAI-17.

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 refinement 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.

Vardi, M. Y. 1998. Reasoning about the past with two-way automata. In Proc. of ICALP ’98, 628–641.

Zarrieß, B., and Turhan, A. 2013. Most specific generalizations w.r.t. general $\mathcal{EL}$-TBoxes. In Proc. of IJCAI, 1191–1197.
Notes for Section 3

For the convenience of the reader we formulate the model-theoretic characterizations also for the verification and existence of the LCS. We start with LCS verification. The following characterization follows from Theorems 1 and 3.

**Theorem 16 (LCS Verification)** Let $L \in \{\Sigma, \Sigma I\}$, $T$ be an $L$ TBox, $C_1, \ldots, C_n$ $L$ concepts, and $\Sigma$ a signature. An $L(\Sigma)$ concept $C$ is the $L(\Sigma)$-LCS of $C_1, \ldots, C_n$ w.r.t. $T$ iff the following conditions hold:

1. $(p_{C_1}, \ldots, p_{C_n}) \in C \cap \Pi_{i=1}^{n} U_{T,C_1};$
2. $\Pi_{i=1}^{n}(U_{T,C_1}, p_{C_1}) \subseteq L, \Sigma U_{T,C_1}, p_{C}.$

For LCS existence, the following characterization follows from Theorems 1 and 4.

**Theorem 17 (LCS Existence)** Let $L \in \{\Sigma, \Sigma I\}$, $T$ be an $L$ TBox, $C_1, \ldots, C_n$ $L$ concepts, and $\Sigma$ a signature. The following are equivalent, for $D_k = (\Pi_{i=1}^{n}(U_{T,C_1}, p_{C_1}) \downarrow_{\Sigma}^k$:

1. the $L(\Sigma)$-LCS of $C_1, \ldots, C_n$ w.r.t. $T$ exists;
2. $D_k$ is the $L(\Sigma)$-LCS of $C_1, \ldots, C_n$ w.r.t. $T$, for some $k \geq 0$;
3. $\Pi_{i=1}^{n}(U_{T,C_1}, p_{C_1}) \subseteq L, \Sigma (U_{T,D_k}, p_{D_k})$, for some $k \geq 0$.

**Proofs for Section 4**

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

**Proof.** We reduce the word problem for polynomially space bounded Turing machines (TMs), that is, given such a TM $M$ with polynomial space bound $p(n)$, we construct an ABox $A$ with individuals $a_1, \ldots, a_n$, such that the $\Sigma$ MSC of $a_1, \ldots, a_n$ w.r.t. $A$ exists iff $M$ accepts an input $w$. It is well-known that there is a deterministic polynomially space bounded TM whose halting problem is PSPACE-hard.

For our purposes, a Turing machine $M = (Q, \Gamma, q_0, \delta, F)$ consists of a set of states $Q$, finite set of tape symbols $\Gamma$, an initial state $q_0$, a set of final states $F$, and a (partial) transition function $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$. There, $L$ and $R$ correspond to the head moving to the left and to the right, respectively. We assume that $M$ halts once it reaches a state $q \in F$, and always continues otherwise.

For the reduction, let $M = (Q, \Gamma, q_0, \delta, F)$ be a $p(n)$-space bounded deterministic TM, and $w$ an input of length $n$. We construct an ABox $A$ without any concept assertions. Let us first fix the following individuals:

$\text{ind}(A) = \{(q, a, i) \mid q \in Q, a \in \Gamma, 1 \leq i \leq p(n)\}$.

Intuitively, an individual $(q, a, i)$ represents that the content of cell $i$ is $a$, that the head of the TM is on cell $i$ and that the TM is in state $q$. Similarly, an individual $(a, i)$ represents that the content of cell $i$ is $a$ (and that the head is not at position $i$). In the following description the cases $i = 1$ and $i = p(n)$ are not treated in a special way, since we can assume that $M$ does not move its head beyond cell 1 or $p(n)$.

As role names, we use $r_{q,a,i}$ for all $(q, a, i) \in \text{ind}(A)$. Informally, a role assertion $r_{q,a,i}(e, e')$ is included in $A$ if in state $q$ with the head at cell $i$ and reading tape symbol $a$, $M$ will change the tape cell represented by $e$ to $e'$. Note that $e$ and $e'$ may be identical, meaning that the TM transition does not affect the tape cell.

Formally, we include the following role assertions for every $q \in Q, a \in \Gamma$, and $i \in \{1, \ldots, p(n)\}$ such that $\delta(q, a) = (q', b, D)$ is defined:

1. Role assertions that affect the direct environment of the head position $i$:

   \begin{align*}
   r_{q,a,i}((q, a, i), (b, i)) & \quad \text{if } D = L \\
   r_{q,a,i}((a', i - 1), (q', a', i - 1)) & \quad \text{if } D = L \\
   r_{q,a,i}((a', i + 1), (a', i + 1)) & \quad \text{if } D = L \\
   r_{q,a,i}((q, a, i), (b, i)) & \quad \text{if } D = R \\
   r_{q,a,i}((a', i - 1), (a', i - 1)) & \quad \text{if } D = R \\
   r_{q,a,i}((a', i + 1), (q', a', i + 1)) & \quad \text{if } D = R
   \end{align*}

2. Role assertions that do not affect the direct environment of the head position $i$:

   \[ r_{q,a,i}((b, j), (b, j)) \quad \text{for all } b \in \Gamma, j \notin \{i - 1, i, i + 1\} \]

This finishes the construction of $A$. It remains to specify the individuals $a_1, \ldots, a_{p(n)}$ for the input $w = b_1 \ldots b_n$:

\[
\begin{align*}
   a_1 & = (q_0, b_1, 1) \\
   a_i & = (b_i, i) \quad \text{for all } i \in \{2, \ldots, n\} \\
   a_i & = (\square, i) \quad \text{for all } i \in \{n + 1, \ldots, p(n)\},
\end{align*}
\]

where $\square$ denotes the blank symbol.

**Claim.** $M$ accepts $w$ iff the $\Sigma$-MSC of $a_1, \ldots, a_n$ w.r.t. $A$ exists.

**Proof of the Claim.** We provide some insight into the construction of $A$. For this purpose, let us denote with $A^{p(n)}$ the $p(n)$-fold product of $A$. For a configuration $\alpha$ of $M$, let $x_\alpha$ denote the element of $A^{p(n)}$ corresponding to this configuration in the natural way. The construction of $A$ ensures the following:

(*) if $\alpha'$ is a successor configuration of $\alpha$ then $x_\alpha$ has precisely one successor in $A^{p(n)}$, namely $x_{\alpha'}$.

Thus, paths in $A^{p(n)}$ starting in $(a_1, \ldots, a_{p(n)})$ directly correspond to computations of $M$ on input $w$.

The claim now follows from (*) and the fact that the MSC exists iff all paths starting from $(a_1, \ldots, a_{p(n)})$ in $A^{p(n)}$ are finite. This finishes the proof of the claim, and in fact of the Theorem.

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

For the proof of Theorem 5 we require the following lemma. For a tree-shaped interpretation $J$ and $e \in \Delta_J$ we denote by $J_e$ the subinterpretation of $J$ induced by the subtree of $J$ rooted at $e$.

**Lemma 6** Let $I$ and $J$ be interpretations with $J$ tree-shaped. If $(I, d) \not\preceq_{\Sigma, \Sigma} (J, e)$, then there exists a set $X$ with $d \in X \subseteq \Delta^I$ such that $|X| \leq |\Delta^J_e| + 1$ and $(I[X], d) \not\preceq_{\Sigma, \Sigma} (J, e)$. 

Proof. The proof is by induction on the depth of $\mathcal{J}$. Assume first that $\mathcal{J}$ has depth 0. If there exists a concept name $A \in \Sigma$ with $d \in A^2$ but $e \notin A^2$, then $X = \{d\}$ is as required. Otherwise there exists a role name $r \in \Sigma$ and $d'$ with $(d, d') \in r^2$. Then $X = \{d, d'\}$ is as required. Now suppose that $e$ has depth $k + 1$ and the lemma has been proved for all $e'$ with $\mathcal{J}_{e'}$ of depth $\leq k$. Assume $\langle I, d \rangle \not\preceq_{\mathcal{E}\mathcal{L}\Sigma} (\mathcal{J}, e)$. If there exists a concept name $A \in \Sigma$ with $d \in A^2$ but $e \notin A^2$, then $X = \{d\}$ is as required. Otherwise there exists a role name $r \in \Sigma$ and $d'$ with $(d, d') \in r^2$ such that for all $e'$ with $(e, e') \in r^3, (I, d') \not\preceq_{\mathcal{E}\mathcal{L}\Sigma} (\mathcal{J}, e')$. Fix $d'$. By induction hypothesis, we can take for every $e'$ with $(e, e') \in r^3$ a set $X_{e'}$ with $d' \in X_{e'} \subseteq \Delta^2$ such that $|X_{e'}| \leq |\Delta^2| + 1$ and $(I_{|X_{e'}}, d') \not\preceq_{\mathcal{E}\mathcal{L}\Sigma} (\mathcal{J}, e')$. Let $X$ be the union of $\{d\}$ and the sets $X_{e'}$, $(e, e') \in r^3$. Then $X$ is as required.

We now give the proof of Theorem 5.

Proof. By Theorem 1, it suffices to give the cONP upper bound for $\mathcal{E}\mathcal{L}(\Sigma)$-MSC verification with empty TBox. Assume an ABox $A, a_1, \ldots, a_n \in \text{ind}(A)$, a signature $\Sigma$, and an $\mathcal{E}\mathcal{L}(\Sigma)$ concept $C$ are given. It can be checked in $\text{PTIME}$ where $A \models C(a_i)$ for $i = 1, \ldots, n$. Thus it suffices to show that

$$\prod_{i=1}^{n} (A, a_i) \not\preceq_{\mathcal{E}\mathcal{L}\Sigma} A_C, \rho_C$$

is in $\text{NP}$, where we regard $A, \prod_{i=1}^{n} A_i$, and $A_C$ as interpretations in the obvious way. But this follows directly since

1. $\varphi$ is unsatisfiable;
2. $\prod_{i=1}^{n} (A_C, \rho_C) \not\preceq_{\mathcal{E}\mathcal{L}} (A_D, \rho_D);$
3. the LCS of $C_1', \ldots, C_n'$ is $\exists_s D$, where $C_i' = \exists s. C_i \cap \exists s. D$ for $i = 1, \ldots, n$.

For better readability, we define concepts $C_1, \ldots, C_n$ in terms of interpretations $I_1, \ldots, I_n$ as follows:

- $\Delta^2 = \{d_1, d_0, d_i\};$
- $s^2 = \{(d_i, d_0), (d_i, d_1)\};$
- $X^2_j = X^2_j = \{d_0, d_1, d_i\}$, for all $j \neq i;$
- $X^2_i = \{d_i\}$ and $X^2_i = \{d_0\}$.

Intuitively, each $I_i$ has a root $d_i$ with two successors $d_0, d_1$ which “choose” a variable for $x_i$. Note that every successor of the root of the product $\prod_{i=1}^{n} I_i$ corresponds to a variable assignment. We define $D$ via the interpretation $\mathcal{J}$ defined as follows:

- $\Delta^2 = \{d_1, \ldots, m\};$
- $s^2 = \{(d, j) \mid j \in \{1, \ldots, m\}\};$
- $j \in X^2_j$ if $x_i$ does not occur positively in clause $j$, for all $j \in \{1, \ldots, m\}$ and $i \in \{1, \ldots, n\};$
- $j \in X^2_i$ if $x_i$ does not occur negatively in clause $j$, for all $j \in \{1, \ldots, m\}$ and all $i \in \{1, \ldots, n\}$.

Note that every element $j$ in $\mathcal{J}$ corresponds to clause $j$ in $\varphi$ and is labeled with all negated literals from the clause, that is, a successor of the root in $\prod_{i=1}^{n} I_i$ maps to $j$ if the corresponding assignment makes the clause false. The concepts $C_1, \ldots, C_n, D$ thus satisfy the equivalence $\Leftrightarrow 2”$ above. For the equivalence $\Leftrightarrow 2”, we use Theorem 3 (note that it applies to the LCS since the constructed ABoxes are essentially $\mathcal{E}\mathcal{L}$ concepts). For “2 $\Rightarrow 3”, note that

- $\exists_s D$ satisfies Condition 1 of Theorem 3, since it is a conjunction in every $C_i'$;
- $\exists_s D$ satisfies Condition 2 of Theorem 3: first note that every $s$-successor in the product $\prod_{i=1}^{n} I_i$ that involves $D$ is trivially simulated by $D$; second note that the the $s$-successor corresponding to $\prod_{i=1}^{n} I_i$ is simulated by $D$ by Point 2 above.

Conversely, that is, from “3 $\Rightarrow 2”, Point 2 above follows from Condition 2 of Theorem 3.

Theorem 7 In $\mathcal{E}\mathcal{L}$, LCS and MSC existence and verification w.r.t. the empty TBox are $\text{PSpace}$-hard and in $\text{EXPTime}$. The lower bounds apply when the signature is full.

The $\text{EXPTime}$ upper bound for MSC existence in $\mathcal{E}\mathcal{L}$ is established in the following.

Lemma 7 In $\mathcal{E}\mathcal{L}$, MSC existence w.r.t. the empty TBox is decidable in $\text{EXPTime}$.

Proof. Assume $K = (\emptyset, A), a_1, \ldots, a_n \in \text{ind}(A)$, and a signature $\Sigma$ are given. We show that the $\mathcal{E}\mathcal{L}(\Sigma)$-MSC of $a_1, \ldots, a_n$ w.r.t. $K$ exists iff there is no infinite path $d_0, r_0, d_1, r_1, d_2, \ldots$ in $Z = \prod_{i=1}^{n} U_K$ satisfying

$$d_0 = (a_1, \ldots, a_n) \text{ and for all } i \geq 0: \text{sg}(r_i) \in \Sigma \text{ and }$$

1. if $r_i = r_{i+1}$, then $(I, d_{i+2}) \not\preceq_{\mathcal{E}\mathcal{L}, \Sigma} (I, d_i);$
2. there is no $e \neq d_{i+1}$ such that $(d_i, e) \in r^2_i, (I, d_{i+1}) \not\preceq_{\mathcal{E}\mathcal{L}, \Sigma} (I, e)$, and $(I, e) \not\preceq_{\mathcal{E}\mathcal{L}} (I, d_{i+1}).$

To prove this characterization, recall that by Theorem 4 the $\mathcal{E}\mathcal{L}(\Sigma)$-MSC of $a_1, \ldots, a_n$ with respect to $K$ exists iff there exists $k \geq 0$ such that for $C_k = \prod_{i=1}^{n} (U_K, a_i) \not\preceq_{\mathcal{E}\mathcal{L}, \Sigma} (U_0, C_i, \rho_{C_i})$. Assume first that there is an infinite path satisfying (1). Then clearly the path cannot be $\mathcal{E}\mathcal{L}(\Sigma)$-simulated by $(U_0, C_i, \rho_{C_i})$ for any $k$ because in any $U_0, C_i$ the length of such paths starting at $\rho_{C_i}$ does not exceed $k$. Conversely, assume there are no infinite paths satisfying (1). Then let $k$ be the length of the longest path satisfying (1). It is readily shown that $(S_k)$ holds, as required. As the universal model $U_K$ can be constructed in exponential time and the existence of infinite paths in $\prod_{i=1}^{n} U_K$ satisfying (1) can also be checked in exponential time, the existence of the $\mathcal{E}\mathcal{L}(\Sigma)$-MSC can be decided in $\text{EXPTime}$. 

$\square$
For the lower bound, we first prove lower bounds for the \( \mathcal{ELT} \)-product simulation problem for the case of tree-shaped interpretations.

**Theorem 18** The \( \mathcal{ELT} \)-product simulation problem on tree-shaped interpretations is \( \text{PSPACE} \)-hard.

We reduce from a tiling problem where the input is a tiling system \((T, H, V)\), an initial tiling \(\theta = t_1, \ldots, t_n\) with tiles from \(T\), and a final tile \(t_F \in T\). The goal is to tile a finite rectangle of size \(m \times n\), \(m, n \geq 1\) arbitrary, such that the first row is tiled with \(\theta\) and \(t_F\) occurs in the tiling. Formally, a solution to a tiling instance \((T, H, V, \theta, t_F)\), \(\theta\) of length \(n\), is a mapping \(\tau : \{1, \ldots, m\} \times \{1, \ldots, n\} \to T\), for some \(m \geq 2\), such that the following conditions are satisfied:

1. \((\tau(i, j), \tau(i, j + 1)) \in H\) for \(1 \leq i \leq m\) and \(1 \leq j < n\);
2. \((\tau(i, j), \tau(i + 1, j)) \in V\) for \(1 \leq i < m\) and \(1 \leq j \leq n\);
3. \((\tau(1, 1), \ldots, \tau(1, n)) = \theta\);
4. \(t_F\) is in the range of \(\tau\).

Let a tiling instance \((T, H, V, \theta, t_F)\) be given. We construct tree interpretations \(I_1, \ldots, I_{3n}, M\) such that for suitably chosen \(d_0\) and \(e_0\), \((\prod_{i=1}^{3n} I_i, d_0) \not\leq_{\mathcal{ELT}} (M, e_0)\) iff there is no solution.

We use the following signature:

1. a single role name \(\tau\);
2. concept names \(T_{i,j}, t \in T, i \in \{1, \ldots, n\}, j \in \{1, 2\}\) to express that position \(i\) is tiled with \(t\);
3. concept names \(M_1, \ldots, M_5\) representing different ‘phases’ we go through when following a path through the product;
4. concept names \(M_{ij}\) with \(1 \leq i, j \leq 5\) and \(j \in \{i-1, i+1\}\), representing transitions between these phases.

Let \(T_H\) be the set of triples \((t_1, t_2, t_3)\) with \((t_1, t_2) \in H\) and \((t_2, t_3) \in H\). For every \(\tau = (t_1, t_2, t_3) \in T_H\), \(i \in \{1, \ldots, n\}\), and \(\ell \in \{1, 2\}\), let \(S_{\tau, \ell}\) be the set of concept names that contains \(T_{i,\ell}, T_{i,\ell}^{-1}, T_{i,\ell}, T_{i,\ell}^{-1}\), and \(T_{i,\ell}\) for every \(j \in \{1, \ldots, i-2, i+2, \ldots, n\}\) and \(\ell \in T\). We admit superscripts \(-1\) and \(+1\) for uniformity purposes.

The interpretations \(I_i, 1 \leq i \leq n\) are defined as follows:

- \(I_i\) is a tree of depth two that branches only at the root \(d_0\);
- for all \(\tau \in T_H\), \(d_0^{\tau}\) has an \(r^-\)-successor \(e_\tau^{i}\) which has an \(r^-\)-successor \(d_\tau^{i}\);
- \(d_0^{\tau}\) has further \(r^-\)-successors \(d_0^{3,4}, d_0^{-4,3}, d_0^{-4,5}, d_0^{-5,4}\);
- \(d_0^{\tau}\) is labeled with \(M_3, M_4, M_5\) and with \(T_{i,\ell}\) for every \(i \in \{1, \ldots, n\}\), \(t \in T\), and \(\ell \in \{1, 2\}\);
- each \(d_0^{i}\) is labeled with \(M_1, M_2, M_3, M_4, M_5\), with every concept name from \(S_{\tau,1}\) and with every concept name \(T_{i,2}\), \(i \in \{1, \ldots, n\}\) and \(t \in T\);
- each \(e_\tau^{i}\) is labeled with \(M_{12}, M_{21}, M_{32}, M_{32}\);
- each \(d_{i,j}^{k}\) is labeled with \(M_{jk}\).

The interpretations \(I_i, n+1 \leq i \leq 2n\) are defined as follows:

- \(I_i\) is a tree of depth two that branches only at the root \(d_0^n\);
- for all \(\tau_1, \tau_2 \in T_H\), \(d_0^n\) has an \(r^-\)-successor \(e_{\tau_1,\tau_2}^{i}\) which has an \(r^-\)-successor \(d_{\tau_1,\tau_2}^{i}\);
- \(d_0^n\) is labeled with \(M_1, M_5\) and with \(T_{i,\ell}\) for every \(i \in \{1, \ldots, n\}\), \(t \in T\), and \(\ell \in \{1, 2\}\);
- each \(d_{\tau_1,\tau_2}^{i}\) is labeled with \(M_2, M_3, M_4, M_5\), with every concept name from \(S_{\tau_1,1}\) and from \(S_{\tau_2,2}\);
- each \(e_{\tau_1,\tau_2}^{i}\) is labeled with all concept names \(M_{jk}\).

The interpretations \(I_i, 2n+1 \leq i \leq 3n\) are defined as follows:

- \(I_i\) is a tree of depth two that branches only at the root \(d_0^n\);
- for all \(\tau \in T_H\), \(d_0^n\) has an \(r^-\)-successor \(e_\tau^n\) which has an \(r^-\)-successor \(d_\tau^n\);
- \(d_0^n\) has further \(r^-\)-successors \(d_1^{12}, d_2^{12}, d_2^{23}, d_3^{23}\);
- \(d_0^n\) is labeled with \(M_1, M_2, M_3, M_4, M_5\) and with \(T_{i,\ell}\) for every \(i \in \{1, \ldots, n\}\), \(t \in T\), and \(\ell \in \{1, 2\}\);
- each \(d_1^{i}\) is labeled with \(M_1, M_2, M_3, M_4, M_5\), with every concept name from \(S_{\tau,2}\) and with every concept name \(T_{i,1}\), \(i \in \{1, \ldots, n\}\) and \(t \in T\);
- each \(e_{\tau}^{i}\) is labeled with \(M_{34}, M_{43}, M_{45}, M_{54}\);
- each \(d_{i,j}^{k}\) is labeled with \(M_{jk}\).

We are mainly interested in paths through \(\prod_{i=1}^{3n} I_i\) that are marked with the following pattern:

\[
M_1, r^- M_{12}, r, M_2, r^- M_{23}, r, M_3, r^- M_{34}, r, M_4, r^- M_{45}, r, M_5, r^- M_{54}, r, \ldots \quad (*)
\]

\[
M_4, r^- M_{43}, r, M_3, r^- M_{32}, r, M_2, r^- M_{21}, r, M_1, r^- M_{12}, r, M_2.
\]

We give an informal description of how the mentioned paths are related to rectangle tilings. Note first that, if an element of \(\prod_{i=1}^{3n} I_i\) satisfies some \(M_i\), then this has implications regarding its components. For instance, if an element \((d_1, \ldots, d_{3n})\) satisfies \(M_3\), then \(d_1, \ldots, d_n\) are all roots of their respective interpretations and so are \(d_{2n+1}, \ldots, d_{3n}\), while \(d_{n+1}, \ldots, d_{2n}\) are leaves. We sketch how to obtain a path through \(\prod_{i=1}^{3n} I_i\) that follows pattern \((*)\) and represents any rectangle tiling. Let \(\theta_1, \theta_2, \ldots\) be an enumeration of the rows of some the tiling.

- In \(\prod_{i=1}^{n} I_i\), start at those \(d_i^{\theta_1}\) that represent \(\theta_1\) by concept names \(T_{i,1}\). In \(\prod_{i=n+1}^{3n} I_i\), we start at \(d_{n+1}^{\theta_1}, \ldots, d_0^{\theta_1}\). This point in the product is labeled \(M_1\).
• Then proceed via an element labeled $M_{1,2}$ to an element labeled $M_2$ that represents $\theta_2$ in the $T_{t,1}$ and $\theta_1$ in the $T_{t,1}^i$.
  The choice is in components $\prod_{i=1}^{2n+1} I_i$ and in $\prod_{i=1}^{2n+1} I_i$.

• Next proceed to the roots in $\prod_{i=1}^{n} I_i$, remaining stationary in $\prod_{i=n+1}^{2n+1} I_i$ (label $M_3$, via $M_{23}$). We still represent $\theta_1$ and $\theta_2$ as before. As explained earlier, this transition serves to verify the vertical matching condition.

• Next proceed to leaves in $\prod_{i=1}^{2n} I_i$, remaining stationary in $\prod_{i=n+1}^{2n+1} I_i$ (label $M_4$, via $M_{34}$). Once more, $\theta_1$ and $\theta_2$ are represented as before. This transition serves no purpose as we move ‘upwards’ (towards higher indexes) in the $M_1, \ldots, M_5$ sequence. When moving downwards, this transition checks the vertical matching condition while the transition in the previous item serves no purpose.

• Then proceed to the roots in $\prod_{i=1}^{2n} I_i$, remaining stationary in all other components (label $M_5$, via $M_{45}$); this preserves the representation of $\theta_2$ via $T_{t,2}$, but ‘forgets’ the representation of $\theta_1$ via $T_{t,1}$.

• Now do everything backwards, from $M_5$ towards $M_1$; first proceed via an element labeled $M_{54}$ to an element labeled $M_4$ that represents $\theta_3$ in the $T_{t,1}$ and $\theta_2$ in the $T_{t,2}$. The choice is in components $\prod_{i=1}^{2n+1} I_i$ and we remain stationary in all other components; then move to $M_3$ via $M_{34}$, and so on.

• After reaching $M_1$, proceed again in the forward direction, representing $\theta_3$, and so forth.

Of course, there are paths through the product that do not follow this ideal pattern, for different reasons. For instance, the desired sequence of the $M_i$ is not followed, some element does not correspond to a valid row in the tiling, or the vertical matching condition is not met. These undesired paths are captured by ‘traps’ in the interpretation $\mathcal{M}$ that we construct next.

We assemble $\mathcal{M}$ by starting with a path of length nine, connected by alternating between $r^-$ and $r$:

\[ e_1 \rightarrow e_{12} \rightarrow e_{23} \rightarrow e_{34} \rightarrow e_{45} \rightarrow e_5 \]

such that each $e_i$ is labeled with $M_i$ and with all concept names $T_{i,\ell}$, $1 \leq i \leq n$, $\ell \in \{1,2\}$, and $t \in T \setminus \{t_F\}$; the missing $t_F$ means that any ‘proper’ path reaching $t_F$ will result in non-simulation. Also, each $e_{ij}$ is labeled with $M_{ij}$ and $M_{ji}$.

We now add traps to make sure that undesired paths are simulated by $\mathcal{M}$. We start with the case that the desired sequence of the $M_i$ is not followed:

1. To every $e_{ij}$, we attach an $r$-successor that is labeled with $M_i$ or every $k \notin \{i,j\}$ and with all concept names $T_{k,\ell}$, and that has an $r^-$-successor which has an $r$-successor that makes true all concept names (including all $T_{i,\ell}^i$), acting as a well of positivity.

2. To every $e_i$, we attach an $r^-$-successor that is labeled with $M_{jk}$ whenever $jk \notin \{i-1, i+1\}$, and that has an $r$-successor which is a well of positivity.

Next, we add traps that address defects which concern a single row of the tiling:

3. To each $e_{ij}$, we attach an $r$-successor for each $k \in \{1, \ldots, n\}$ and $l \in \{1,2\}$. No concept name $T_{i,\ell}$ is true there, but all concept names $T_{j,\ell}$ with $(j,m) \neq (k,l)$ and $t \in T$, and of course there is a well below it. This has two effects:

   (a) it enforces synchronization of the tiling of each $i$-th column across the $\prod_{i=1}^{n} I_i$ resp. $\prod_{i=n+1}^{2n+1} I_i$; so the horizontal tiling condition is satisfied;

   (b) in $M_2$ and $M_3$-configurations, it enforces that the tiling of the row that is represented twice, once in $\prod_{i=1}^{n} I_i$ and once in $\prod_{i=n+1}^{2n+1} I_i$, and once in $\prod_{i=2n+1}^{2n+1} I_i$, is identical.

It remains to deal with paths that violate the vertical matching condition:

4. at $e_{23}$, we attach a trap for each $(t_1, t_2) \notin V$ and each $i \in \{1, \ldots, n\}$, labeled with $M_2$, with $T_{t_1,1}$ and $T_{t_2,2}$, and with $T_{j,\ell}$ for all $j \neq i$, $\ell \in \{1,2\}$, and $t \in T$;

5. at $e_{34}$, we attach a trap for each $(t_2, t_1) \notin V$ and each $i \in \{1, \ldots, n\}$, labeled with $M_4$, with $T_{t_2,1}$ and $T_{t_1,2}$, and with $T_{j,\ell}$ for all $j \neq i$, $\ell \in \{1,2\}$, and $t \in T$.

The initial tiling $\theta = t_1, \ldots, t_n$ gives rise to a sequence of triples $\tau_1, \ldots, \tau_n$ in the obvious way. We are going to use

\[ d_0 = (d_1^{0}, \ldots, d_n^{0}, d_0^{n+1}, \ldots, d_0^{3n}) \]

as the starting point for the simulation.

**Lemma 8** $\prod_{i=1}^{2n} I_i, d_0 \preceq_{\mathcal{E}M} (\mathcal{M}, e_1)$ iff there is no solution for $(T, H, V, \theta, t_F)$.

**Proof.** “if”. Assume that there is no solution for $(T, H, V, \theta, t_F)$. We prove the existence of a simulation from $(\prod_{i=1}^{2n} I_i, d_0)$ to $(\mathcal{M}, e_1)$.

Let us first introduce some notation. We call a tuple $t_1, \ldots, t_n \in T$ possible if there is a mapping $\tau$ with $\tau(i, 1) = t_1, \ldots, \tau(i, n) = t_n$, for some $\tau$ which satisfies Condition 1–3 of a solution (but does not necessarily mention $t_F$). As there is no solution for $(T, H, V, \theta, t_F)$, no tuple that is possible mentions $t_F$.

Now, we say that $d$ is $k$-proper if $d$ satisfies $M_k$ and

- if $k = 1$, then the $T_{t,1}$ represent a possible row at $d$;
- if $k = 2$, then the $T_{i,1}$ represent a possible row, and the $T_{i,2}$ represent some row satisfying the horizontal tiling condition;
- if $k = 3$, then the $T_{i,1}$ represent possible rows, for $\ell \in \{1,2\}$;
- if $k = 4$, then the $T_{i,2}$ represent a possible row, and the $T_{i,1}$ represent some row satisfying the horizontal tiling condition;
- if $k = 5$, then the $T_{i,2}$ represent a possible row.
We claim that there is a simulation $S$ from $\prod_{i=1}^{3n} \mathcal{I}_i, d_0$ to $(\mathcal{M}, e_1)$ which relates all $k$-proper elements in $\prod_{i=1}^{3n} \mathcal{I}_i$ with $e_k$. The statement then follows, because the initial tuple $d_0$ is 1-proper by construction.

We show the arguments only for $k = 2$ because the other cases are similar. Thus, take any $d$ that is 2-proper, and assume $(d, e_2) \in S$. We show how to continue the simulation from $(d, e_2)$. To this end, let $d'$ be an $r$-successor of $d$. We distinguish several cases that can arise by the construction of the $\mathcal{I}_i$:

- if $d'$ does not satisfy one of $M_{21}$ or $M_{23}$, then $d'$ is simulated by the trap of type 2 at $e_2$.
- if $d'$ satisfies $M_{21}$, then we add $(d', e_{12})$ to the simulation. Now, let $d''$ be any $r$-successor of $d'$. Since $d'$ satisfies $M_{21}$, we have to be in $d_{21}$ for all $i \in \{2n + 1, \ldots, 3n\}$ and thus $d''$ satisfies one of $M_1, M_2, M_3$ or no $M_i$ at all. We again distinguish cases:
  - if $d''$ satisfies $M_3$ or no $M_i$ at all, then $d''$ is simulated by a trap of type 1 at $e_{12}$;
  - if $d''$ satisfies $M_2$, then, by construction of the $\mathcal{I}_i$, $d''$ is actually $d$ and it is simulated by $e_2$;
  - if $d''$ satisfies $M_1$, the construction of the $\mathcal{I}_i$ implies that the rows represented by $T^1_{t, \ell}$ at $d''$ are the same as these rows at represented by $T^1_{t, \ell}$ at $d$. Since the latter is possible by assumption, so is the former. Thus, $d''$ is 1-proper and we know that $(d'', e_1) \in S$.
- if $d'$ satisfies $M_{23}$, then we add $(d', e_{23})$ to the simulation and continue as in the previous case. More precisely, let $d''$ be any $r$-successor of $d'$. Since $d'$ satisfies $M_{23}$, we have to be in $d_{23}$ for all $i \in \{2n + 1, \ldots, 3n\}$ and thus $d''$ satisfies one of $M_1, M_2, M_3$ or no $M_i$ at all. We again distinguish cases:
  - if $d''$ satisfies $M_3$ or no $M_i$ at all, then $d''$ is simulated by a trap of type 1 at $e_{23}$;
  - if $d''$ satisfies $M_2$, then, by construction of the $\mathcal{I}_i$, $d''$ is actually $d$ and it is simulated by $e_{23}$;
  - if $d''$ satisfies $M_1$, the construction of the $\mathcal{I}_i$ implies that the rows represented by $T^1_{t, \ell}$ at $d''$ are the same as these rows at represented by $T^1_{t, \ell}$ at $d$. Let $t_1, \ldots, t_n$ and $t_1', \ldots, t_n'$ be the rows represented by $T^1_{t, \ell_0}$ and $T^1_{t_0, \ell_0}$, respectively. If $(t_j, t_j') \notin V$, for some $j$, then $d''$ is simulated by a trap of type 4 at $e_{23}$. Otherwise, the row represented by the $T^1_{t, \ell_0}$ at $d''$ is a valid successor of the row represented by the $T^1_{t, \ell_0}$ at $d$. Overall, $d''$ is 3-proper and we know that $(d'', e_{13}) \in S$.

"only if". Assume that there is a solution $\tau$ for $(T, H, V, \theta, t_F)$, and let $t_F$ occur in the last row. Let $d_0, r_0, d_0, r, d_1, \ldots, d_n$ be the path which follows the pattern (…) that is contained in the product and which reflects the solution $\tau$. (We can obtain this path by letting $\tau$ guide the selection of successors as described above). By construction, the path satisfies the following properties, for all $i \geq 0$:

- if $d_i$ satisfies $M_1$, then the $T^i_{t, \ell}$ represent a row of $\tau$;
- if $d_i$ satisfies $M_2$, then the $T^i_{t, \ell}$ represent rows $\theta_{t, \ell}$, for $\ell \in \{1, 2\}$, respectively, and
  - if $d_{i-1}$ satisfies $M_1$, $\theta_2$ is a successor row of $\theta_1$ in $\tau$ and $\theta_1$ is represented by the $T^i_{1, \ell}$ at $d_{i-1}$, and
  - if $d_{i-1}$ satisfies $M_3$, $\theta_1$ is a successor row of $\theta_2$ in $\tau$ and $\theta_2$ is represented by the $T^i_{1, \ell}$ at $d_{i-1}$;
- if $d_i$ satisfies $M_3$, then the $T^i_{t, \ell}$ and the $T^i_{t, \ell}$ represent rows of $\tau$, and in fact the same rows as the $T^i_{t, \ell}$ and $T^i_{t, \ell}$ at $d_{i-1}$;
- if $d_i$ satisfies $M_4$, then the $T^i_{t, \ell}$ represent rows $\theta_{t, \ell}$, for $\ell \in \{1, 2\}$, respectively, and:
  - if $d_{i-1}$ satisfies $M_2$, then $\theta_1$ and $\theta_2$ are also represented by the $T^i_{t, \ell}$ and $T^i_{t, \ell}$ at $d_{i-1}$;
  - if $d_{i-1}$ satisfies $M_5$, then $\theta_1$ is a successor row of $\theta_2$ in $\tau$, and $\theta_2$ is represented by the $T^i_{1, \ell}$ at $d_{i-1}$;
- if $d_i$ satisfies $M_5$, then the $T^i_{t, \ell}$ represent a row of $\tau$, and the $T^i_{t, \ell}$ represent the same row at $d_{i-1}$;
- if $d_i$ satisfies $M_{jk}$ such that $d_i$ satisfies $M_j$ and $d_{i+1}$ satisfies $M_k$ or $d_{i+1}$ satisfies $M_j$ and $d_i$ satisfies $M_k$.

In order to show that there is no simulation, we show that:

1. $(i)$ if $d_i$ satisfies $M_k$ it can only be simulated by $e_k$.

Note that this is a contradiction since $d_n$ satisfies $T^i_{t, \ell}$ for some $i, \ell$, but none of the $e_i$ does.

We argue inductively. The induction base is given by the fact that $d_0$ has to be simulated by $e_0$ in the lemma. In the inductive step, we suppose that $d_i$ satisfies some $M_k$ and show that $d_{i+1}$ can only be simulated by $e_{k-1}$ or $e_{k+1}$, respectively, depending on whether $d_{i+1}$ satisfies $M_{k-1}$ or $M_{k+1}$.

We show how to argue for $k = 2$, assuming that we are in the downward phase of the construction of the path, that is, $d_{i+1}$ will be labeled with $M_3$. Based on the invariants given above, it can be verified that $d_{i'}$ and $d_{i+1}$ cannot be simulated by any trap, and thus have to be simulated by $e_{23}$ and $e_{33}$, respectively.

To show PSPACE hardness of LCS and MSC verification and existence we first observe that the tiling problem used in the proof of Theorem 18 is still PSPACE hard if one only considers tiling instances $(T, H, V, \theta, t_F)$ such that the initial tiling $\theta$ only occurs in the first row for any mapping $\tau: \{1, \ldots, m\} \times \{1, \ldots, n\} \rightarrow T$, $m \geq 2$, such that the first three conditions for a solution are satisfied:

1. $(\pi(i, j), \pi(i, j+1)) \in H$ for $1 \leq i \leq m$ and $1 \leq j < n$;
2. $(\pi(i, j), \pi(i+1, j)) \in V$ for $1 \leq i < m$ and $1 \leq j < n$;
3. $(\pi(1, 1), \pi(1, n)) = \theta$.

If this is the case then the pair $(\prod_{i=1}^{3n} \mathcal{I}_i, d_0)$, $(\mathcal{M}, e_1)$ constructed above has the following property which we require in the reduction to LCS and MSC verification and existence. Let $\mathcal{I}$ and $\mathcal{J}$ be interpretations and $d \in \Delta^\mathcal{J}$. An $\mathcal{ELI}$ simulation $S$ from $\mathcal{I}$ to $\mathcal{J}$ is called d-injective if there exists exactly one $e \in \Delta^\mathcal{J}$ with $(d, e) \in S$. We write $(\mathcal{I}, d) \leq_{\mathcal{ELI}} (\mathcal{J}, e)$ if there exists a d-injective $\mathcal{ELI}$ simulation $S$ from $\mathcal{I}$ to $\mathcal{J}$ that contains $(d, e)$. We say that a pair $(\mathcal{I}, d)$, $(\mathcal{J}, e)$ is oblivious.
to d-injectivity if \((\mathcal{I}, d) \preceq^{d,\text{inj}}_{\mathcal{E}LI} (\mathcal{J}, e)\) iff \((\mathcal{I}, d) \preceq_{\mathcal{E}LI} (\mathcal{J}, e)\). It is easy to show the following.

**Lemma 9** The pair \((\prod_{i=1}^{3n} I_i, A_0, (M, e_1))\) is oblivious to d-injectivity if the input tiling instance \((T, H, V, \theta_1, t_F)\) is such that the initial tiling \(\theta_1\) can only occur in the first row for any mapping \(\tau\) satisfying the first three conditions of solutions.

Now let \(C_1, \ldots, C_n\) and \(D\) be \(\mathcal{E}LI\) concepts and assume that \(n \geq 2\). Consider the concepts \(D_1, D_2\) constructed in Example 3 and let

\[
D_3 = A_1 \cap A_2 \sqcap \exists r. (A_3 \sqcap A_4),
\]

where we assume that the signature of \(D_1, D_2\) is disjoint from the signature of \(C_1, \ldots, C_n\). Set \(C'_i = C_i \cap D_i\) if \(i\) is even, \(C''_i = C_i \cap D_i\) if \(i\) is odd, and let \(D' = D \sqcap D_3\). Let \(v\) be a fresh role name and \(E_i = \exists v. C'_i \sqcap \exists v. D'\) for \(i = 1, \ldots, n\). PSPACE-hardness of LCS and MSC verification and existence now follow directly from the following reduction.

**Lemma 10** Assume that \(\prod_{i=1}^{n} (U_{C_i}, \rho_{C_i})\) is oblivious to \((\rho_{C_1}, \ldots, \rho_{C_n})\)-injectivity. Then the following conditions are equivalent:

1. \(\prod_{i=1}^{n} (U_{C_i}, \rho_{C_i}) \preceq_{\mathcal{E}LI} (U_D, \rho_D)\);
2. \(\exists v. D'\) is the \(\mathcal{E}LI\)-LCS of \(E_1, \ldots, E_n\);
3. The \(\mathcal{E}LI\)-LCS of \(E_1, \ldots, E_n\) exists.

**Proof.** "1 \implies 2.\" By Theorem 3, it suffices to show

1. \((\rho_{E_1}, \ldots, \rho_{E_n}) \in (\exists v. D') \prod_{i=1}^{n} U_{E_i}\);
2. \(\prod_{i=1}^{n} (U_{E_i}, \rho_{E_i}) \preceq_{\mathcal{E}LI} (U_{\exists v. D'}, \rho_{\exists v. D'})\).

Condition 1 follows directly from the construction. For Condition 2, we construct for every \(v\) successor \(d = (d_1, \ldots, d_n)\) of \((\rho_{E_1}, \ldots, \rho_{E_n})\) a \(d\)-injective \(\mathcal{E}LI\) simulation \(S_{d_v}\) between \(\prod_{i=1}^{n} (U_{E_i}, d_i)\) and \((U_{\exists v. D'}, \rho_{\exists v. D'})\).

Then we are done: let \(V\) be the set of \(v\)-successors of \((\rho_{E_1}, \ldots, \rho_{E_n})\).

Then

\[
S = \{(d_1, \ldots, d_n), \rho_{E_1}, \ldots, \rho_{E_n})\} \cup \bigcup_{d \in V} S_{d_v}
\]

witnesses \(\prod_{i=1}^{n} (U_{E_i}, \rho_{E_i}) \preceq_{\mathcal{E}LI} (U_{\exists v. D'}, \rho_{\exists v. D'})\), as required. Assume \(d = (d_1, \ldots, d_n)\) is given.

Case 1. \((d_1, \ldots, d_n) = (\rho_{C_1}, \ldots, \rho_{C_n})\).

To construct a \(\rho_{C_1}', \ldots, \rho_{C_n}'\)-injective \(\mathcal{E}LI\) simulation let \(S_0\) be a \(\rho_{C_1}', \ldots, \rho_{C_n}'\)-injective \(\mathcal{E}LI\) simulation between \(\prod_{i=1}^{n} (U_{C_i}, \rho_{C_i})\) and \((U_D, \rho_D)\).

It exists since \(\prod_{i=1}^{n} (U_{C_i}, \rho_{C_i})\) is oblivious to \((\rho_{C_1}, \ldots, \rho_{C_n})\)-injectivity. To define the simulating nodes for the remaining elements of \(\prod_{i=1}^{n} (U_{C_i \sqcap D_i})\) (where we set \(D'_i = D_i\) if \(i\) is even and \(D'_i = D_i\) if \(i\) is odd) simply choose the projection to a fixed component \(D'_i\) composed with the obvious \(\rho_{D'}\)-injective \(\mathcal{E}LI\) simulation between \((U_{D'}, \rho_{D'})\) and \((U_{D'_3}, \rho_{D'_3})\).

The resulting relation is a \(\rho_{C_1}', \ldots, \rho_{C_n}'\)-injective \(\mathcal{E}LI\) simulation.

Case 2. There exists \(d_i\) with \(d_i = \rho_{D \sqcap D_3}\). Then we take the projection to the component \(U_{D \sqcap D_3}\) and obtain a \((d_1, \ldots, d_n)\)-injective \(\mathcal{E}LI\) simulation.

“2 \implies 3” is trivial.

“3 \implies 1.” We use Theorem 4. There exists \(k \geq 0\) such that for \(G_k = \prod_{i=1}^{n} (U_{E_i}, \rho_{E_i}) \preceq_{\mathcal{E}LI} (U_{G_k}, \rho_{G_k})\).

Consider the \(v\) successor \((\rho_{C_1'}, \ldots, \rho_{C_n'})\) of \((\rho_{E_1}, \ldots, \rho_{E_n})\) in \(\prod_{i=1}^{n} (U_{E_i})\). There exists a \(v\)-successor \((d_1, \ldots, d_n)\) of \(\rho_{G_k}\) such that

\[
\prod_{i=1}^{n} (U_{C_i'}, \rho_{C_i'}) \preceq_{\mathcal{E}LI} (U_{G_k}, (d_1, \ldots, d_n))
\]

Note that \((d_1, \ldots, d_n) \neq (\rho_{C_1'}, \ldots, \rho_{C_n'})\) because \(n \geq 2\) and the chain constructed in Example 3 shows that there is no \(\mathcal{E}LI\) simulation between \(\prod_{i=1}^{n} (U_{D_i'}, \rho_{D_i'})\) and any \((U_{G_k}, (\rho_{C_1'}, \ldots, \rho_{C_n'}))\), where \(D'_i = D_i\) if \(i\) is even and \(D'_i = D_i\) if \(i\) is odd. It follows that there exists \(i \leq n\) such that \(d_i = \rho_{D \sqcap D_3}\).

Then

\[
(\rho_{G_k}, (d_1, \ldots, d_n)) \preceq_{\mathcal{E}LI} (U_{D \sqcap D_3}, \rho_{D \sqcap D_3})
\]

as the projections onto components are \(\mathcal{E}LI\) simulations. By taking the composition of \(\mathcal{E}LI\) simulations we obtain

\[
\prod_{i=1}^{n} (U_{C_i'}, \rho_{C_i'}) \preceq_{\mathcal{E}LI} (U_{D \sqcap D_3}, \rho_{D \sqcap D_3})
\]

Condition 1 follows by construction.

□

**Proofs for Section 5**

**Theorem 8** For \(\mathcal{L} \in \{\mathcal{E}, \mathcal{E}LI\}\), 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.** Let \(\mathcal{L} \in \{\mathcal{E}, \mathcal{E}LI\}\). Similarly to the proof of Theorem 2, one can show that it suffices to provide a reduction of the following problem: given an \(\mathcal{L}\) TBox \(T\), an ABox \(A\) with assertions \(A_1(a_1), \ldots, A_n(a_n), B(b)\), where \(A_1, \ldots, A_n, B\) are concept names, and a signature \(\Sigma\) containing \(B\), is it the case that

\[
\prod_{i=1}^{n} (\rho_{K_i}, a_i) \preceq_{\mathcal{E}LI} (K, b),
\]

where \(K = (T, A)\)? We start with the reduction for \(\mathcal{E}\).

Assume an \(\mathcal{E}\) TBox \(T\) and an ABox \(A\) with assertions \(A_1(a_1), \ldots, A_n(a_n), B(b)\), where \(A_1, \ldots, A_n, B\) are concept names, and a signature \(\Sigma\) containing \(B\), is it the case that

\[
\prod_{i=1}^{n} (\rho_{K_i}, a_i) \preceq_{\mathcal{E}LI} (K, b),
\]

where \(K = (T, A)\)? We start with the reduction for \(\mathcal{E}\).

Assume an \(\mathcal{E}\) TBox \(T\) and an ABox \(A\) with assertions \(A_1(a_1), \ldots, A_n(a_n), B(b)\), where \(A_1, \ldots, A_n, B\) are concept names, and a signature \(\Sigma\) containing \(B\), is it the case that

\[
\prod_{i=1}^{n} (\rho_{K_i}, a_i) \preceq_{\mathcal{E}LI} (K, b),
\]
The relativisation $T_E$ of a TBox $T$ to a concept name $E$ is defined by setting
$$T_E = \{ C | E \subseteq D | E \subseteq D \in T \}$$

Construct a new TBox
$$T' = T_{B_1} \cup T_{B_2} \cup T_{B_3} \cup \{ B_i \subseteq \exists w.B_i \mid i = 1, 2, 3 \},$$
where $w$ is a fresh role name and $B_1, B_2, B_3$ are fresh concept names. Take a fresh role name $v$ and construct an ABox $A'$ with individuals $\rho_i, a_i, b_i, i = 1, \ldots, n$, and assertions

- $v(\rho_i, a_i), v(\rho_i, b_i), i = 1, \ldots, n$;
- $A_i(a_i), B_i(b_i), i = 1, \ldots, n$;
- $B_2(a_i)$ if $i$ is even;
- $B_2(a_i)$ if $i$ is odd;
- $B_3(b_i)$ for $i = 1, \ldots, n$.

Let $C_0 = \exists w. (B \cap B_3)$, $K' = (T', A')$, and $\Sigma' = \Sigma \cup \{ B_1, B_2, B_3, v, w \}$. We show that the following conditions are equivalent:

(a) $\Pi_n^{\Sigma' \setminus \Sigma} (U_{K'}, a_i) \preceq_{\Sigma', \Sigma} (U_K, b)$;
(b) $C_0$ is the $\mathcal{E}\mathcal{L}(\Sigma')$-MSC of $\rho_1, \ldots, \rho_n$ w.r.t. $K'$;
(c) The $\mathcal{E}\mathcal{L}(\Sigma')$-MSC of $\rho_1, \ldots, \rho_n$ w.r.t. $K'$ exists.

(a) $\Rightarrow$ (b). Assume (a) holds. By Theorem 3, it suffices to show
1. $\exists n \in \mathbb{N}. (\rho_1, \ldots, \rho_n) \in C_0^{\Pi_n^{\Sigma' \setminus \Sigma} U_{K'}}$;
2. $\Pi_n^{\Sigma' \setminus \Sigma} (U_{K'}, \rho_i) \preceq_{\Sigma', \Sigma} (U_{T'}, C_0, \rho_c, c_0)$.

Condition 1 follows from $((\rho_1, \ldots, \rho_n), (b_1, \ldots, b_n)) \in v^{\Pi_n^{\Sigma' \setminus \Sigma} U_{K'}}$ and
$$(b_1, \ldots, b_n) \in (B \cap B_3)^{\Pi_n^{\Sigma' \setminus \Sigma} U_{K'}}$$

since $C_0 = \exists w. (B \cap B_3)$. For Condition 2, it follows from the construction of $K'$ that it suffices to show that
$$\Pi_n^{\Sigma' \setminus \Sigma} (U_{K'}, \rho_i) \preceq_{\Sigma', \Sigma} (U_{T'}, (B \cap B_3), \rho_{B \cap B_3})$$

for every $v$-successor $(d_1, \ldots, d_n)$ of $(\rho_1, \ldots, \rho_n)$ in $\Pi_n^{\Sigma' \setminus \Sigma} U_{K'}$. If there exists $i \leq n$ with $d_i = b_i$ then this follows from the fact that projections are simulations and the construction of $K'$. Otherwise $d_i = a_i$ for all $i = 1, \ldots, n$. Denote by

- $U_{a_i}$ the tree-shaped subinterpretation rooted at $a_i$ in $U_{K'}$;
- $U_{b_i}$ the tree-shaped subinterpretation rooted at $b_i$ in $U_{K'}$;
- $U_{B \cap B_3}$ the tree-shaped subinterpretation rooted at $\rho_{B \cap B_3}$ in $U_{T', C_0}$;
- $U_B$ the tree-shaped subinterpretation rooted at $b$ in $U_{K'}$.

Observe that $U'_{a_i}$ is obtained from $U_{a_i}$ by adding infinite $w$-chains starting from each node in $U_{a_i}$ and making $B_1$ true in every node if $i$ is even and $B_2$ in every node if $i$ is odd. Similarly, $U'_{B \cap B_3}$ is obtained from $U_B$ by adding infinite $w$-chains starting from each node in $U_B$ and making $B_3$ true in every node. By definition, it suffices to show that
$$\Pi_n^{\Sigma' \setminus \Sigma} (U'_{a_i}, a_i) \preceq_{\Sigma', \Sigma} (U'_{B \cap B_3}, \rho_{B \cap B_3}).$$

But this follows directly from the fact that
$$\prod_{i=1}^{n} (U_{a_i}, a_i) \preceq_{\Sigma, \Sigma} (U_K, b)$$

and the assumption that $n \geq 2$ which implies that no $B_i, i = 1, 2, 3$ is satisfiable in $\Pi_n^{\Sigma' \setminus \Sigma} (U'_{a_i}, a_i)$.

(b) $\Rightarrow$ (c) is trivial.

(c) $\Rightarrow$ (a) We use Theorem 4. There exists $k \geq 0$ such that
$$\Pi_n^{\Sigma' \setminus \Sigma} (U_{K'}, \rho_i) \preceq_{\Sigma', \Sigma} (U_{T'}, G_k, \rho_{G_k}).$$

Consider $(\rho_1, \ldots, \rho_n)$ and the product $\Pi_n^{\Sigma' \setminus \Sigma} (U'_{a_i}, a_i)$ of which it is the root. There exists a $v$-successor $(d_1, \ldots, d_n)$ of $\rho_{G_k}$ in $U_{T', G_k}$ such that
$$\prod_{i=1}^{n} (U'_{a_i}, a_i) \preceq_{\Sigma', \Sigma} (U_{T'}, G_k, (d_1, \ldots, d_n))$$

Note that $(d_1, \ldots, d_n) \neq (a_1, \ldots, a_n)$ because there is an infinite $w$-chain starting at $(a_1, \ldots, a_n)$ in $\prod_{i=1}^{n} U'_{a_i}$ but there is no such $w$-chain starting at $(a_1, \ldots, a_n)$ in $U_{T', G_k}$ because no $B_i, i = 1, 2, 3$, is satisfied in any node of $G_k$ in the subtree generated by $(a_1, \ldots, a_n)$ and $w$-chains are only generated by the TBox $T'$ from nodes satisfying at least one $B_i$. It follows that there exists $i \leq n$ such that $d_i = b_i$. Then
$$(U_{T'}, G_k, (d_1, \ldots, d_n)) \preceq_{\Sigma', \Sigma} (U_{K'}, b_i)$$
as the projections onto components are $\mathcal{E}\mathcal{L}$ simulations. By taking the composition of $\mathcal{E}\mathcal{L}$ simulations we obtain
$$\prod_{i=1}^{n} (U'_{a_i}, a_i) \preceq_{\Sigma', \Sigma} (U_{K'}, b_i)$$

(a) follows by construction of $K'$.

We now sketch the proof for $\mathcal{E}\mathcal{L}T$. The construction given above for $\mathcal{E}\mathcal{L}$ almost works except that in the proof of (a) $\Rightarrow$ (b), for $\mathcal{E}\mathcal{L}T$ simulations we have to consider the $v$-predecessor $(\rho_1, \ldots, \rho_n)$ of $(d_1, \ldots, d_n)$ when proving
$$\Pi_n^{\Sigma' \setminus \Sigma} (U_{K'}, \rho_i) \preceq_{\Sigma', \Sigma} (U_{T', (B \cap B_3)}, \rho_{B \cap B_3})$$to show that Condition 2 for $\mathcal{E}\mathcal{L}T$-MSC verification in Theorem 3 holds. In the $\mathcal{E}\mathcal{L}$ case, the nodes $\mathcal{E}\mathcal{L}$ simulating $(d_1, \ldots, d_n)$ might not have $v$-predecessors simulating $(\rho_1, \ldots, \rho_n)$ to ensure the existence of appropriate $v$-predecessors we modify the construction as follows. Given an $\mathcal{E}\mathcal{L}T$ TBox $T$ and an ABox $A$ with assertions $A_1(a_1), A_2(a_2), B(b)$, with $A_1, A_2, B$ concept names, and a signature $\Sigma$, we construct $T'$ and $A'$ as before but add to $T'$ the CIs $B_3 \subseteq \exists v \cdot B_4$ and
$$B_4 \subseteq X \cap \bigcap_{s \in \text{role}(T')} \exists s.B_4,$$where $\text{role}(T')$ denotes the set of all role names and their inverses used in $T'$ and $X = \bigcap_{A \in \text{sub}(T') \setminus A}$ and $\Sigma'$ also contains $B_4$. Then one can prove that the following conditions are equivalent:

(a) $\Pi_n^{\Sigma' \setminus \Sigma} (U_{K'}, a_i) \preceq_{\Sigma', \Sigma} (U_K, b)$;

(b) $C_0$ is the $\mathcal{E}\mathcal{L}(\Sigma')$-MSC of $\rho_1, \ldots, \rho_n$ w.r.t. $K'$;

(c) The $\mathcal{E}\mathcal{L}(\Sigma')$-MSC of $\rho_1, \ldots, \rho_n$ w.r.t. $K'$ exists.

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

**Proof.** Let $\mathcal{L} \in \{\mathcal{EL}, \mathcal{ELT}\}$. Let $\mathcal{K} = (T, A)$ be an $\mathcal{L}$ knowledge base, $a_1, \ldots, a_n \in \text{ind}(A)$ individuals, $\Sigma$ a signature, and $C$ an $\mathcal{L}(\Sigma)$ concept. We construct a new ABox $A'$ as follows:

- start with $A$ extended with a disjoint copy of $A$ where every individual $a \in \text{ind}(A)$ is replaced with $a'$;
- take a fresh role name $s$, and let $A_{ij}$, with $i, j \in \{1, \ldots, n\}$, be (disjoint) copies of $A_C$ with roots $\rho_{ij}$. Then add the ABoxes $B_i$, for every $i \in \{1, \ldots, n\}$, $A_C$ (with root $\rho_C$), and $B'$ (also with root $\rho_C$) defined as follows:

$$B_i = \bigcup_{j \in \{1, \ldots, n\}} A_{ij} \cup \{s(a_i, \rho_{i1})\} \cup \{s(\rho_{ij}, \rho_{i(j+1)}) \mid j \in \{1, \ldots, n-1\}\}$$

$$B' = \{s(\rho_C, a'_i)\} \cup \{s(a'_i, a'_{i+1}) \mid i \in \{1, \ldots, n-1\}\}$$

Intuitively, $B$ adds an $s$-chain of length $n$ to every $a_i$ in which every element satisfies $C$, and $B'$ adds an $s$-chain to the copies of the individuals $a_i$.

Let $\mathcal{K}' = (T, A')$ and $\Sigma' = \Sigma \cup \{s\}$. Moreover, let $U$ denote the interpretation that is obtained by taking the union of $\prod_{i=1}^n (\mathcal{U}_K, a_i)$ and $\mathcal{U}_{T, B_1}$ (the index is not important), identifying the root $a_1$ of $\mathcal{U}_{T, B_1}$ with the root $(a_1, \ldots, a_n)$ of the product. Let $\rho$ denote the new root of $U$. Note that we have:

$$\prod_{i=1}^n (\mathcal{U}_{K'}, a_i) \preceq_{\mathcal{L}, \Sigma'} (U, \rho) \preceq_{\mathcal{L}, \Sigma'} \prod_{i=1}^n (\mathcal{U}_{K'}, a_i). \quad (1)$$

The second simulation exists since $U$ is a sub-structure of the product. The first simulation exists because

$$\prod_{i=1}^n (\mathcal{U}_{K'}, a_i) \preceq_{\mathcal{L}, \Sigma} (U, \rho) \preceq_{\mathcal{L}, \Sigma} \prod_{i=1}^n (\mathcal{U}_{K'}, a_i)$$

and for elements in the product reachable via some $s$-successor of $(a_1, \ldots, a_n)$, any projection is an $\mathcal{L}(\Sigma)$-simulation to $\mathcal{U}_{T, B_1}$.

**Claim.** $C$ is the $\mathcal{L}(\Sigma)$-MSC of $a_1, \ldots, a_n$ w.r.t. $\mathcal{K}$ iff $\prod_{i=1}^n (\mathcal{U}_{K'}, a_i) \preceq_{\mathcal{L}, \Sigma} (U, \rho, \rho_C)$.

**Proof of the Claim.** For the “if” direction, suppose $\prod_{i=1}^n (\mathcal{U}_{K'}, a_i) \preceq_{\mathcal{L}, \Sigma} (U, \rho, \rho_C)$. By Equation (1), it follows that

$$(U, \rho) \preceq_{\mathcal{L}, \Sigma} (\mathcal{U}_{K'}, \rho_C).$$

Since $s$ is fresh, we have that $\prod_{i=1}^n (\mathcal{U}_{K'}, a_i) \preceq_{\mathcal{L}, \Sigma} (\mathcal{U}_{T, A_C}, \rho_C)$ and $\mathcal{U}_{T, B_1} \preceq_{\mathcal{L}, \Sigma} (\mathcal{U}_{T, B'}, \rho_C).$ The former is just Condition 2 of Theorem 3 is satisfied. Moreover, the latter implies that $(\mathcal{U}_{T, A_i}, \rho_{ij}) \preceq_{\mathcal{L}, \Sigma} (\mathcal{U}_{K'}, a'_i)$, for all $i \in \{1, \ldots, n\}$. Thus, we also have $\mathcal{K} \models C(a_i)$, for all $i$ and hence also Condition 1 of Theorem 3 holds.

For “only if”, suppose that $C$ satisfies Conditions 1 and 2 of Theorem 3. The former implies that $\mathcal{K} \models C(a_i)$ for all $i \in \{1, \ldots, n\}$, and thus $(\mathcal{U}_{T, A_C}, \rho_C) \preceq_{\mathcal{L}, \Sigma} (\mathcal{U}_{K'}, a_i)$, for all $i$.

It can be verified that $(\mathcal{U}_{T, B}, a_1) \preceq_{\mathcal{L}, \Sigma} (\mathcal{U}_{T, B'}, \rho_C)$. Moreover, Condition 2 reads $\prod_{i=1}^n (\mathcal{U}_{K'}, a_i) \preceq_{\mathcal{L}, \Sigma} (\mathcal{U}_{T, A_C}, \rho_C)$. Together, we have $(U, \rho) \preceq_{\mathcal{L}, \Sigma} (\mathcal{U}_{K'}, \rho_C)$, and the claim follows from Equation (1).

The Claim establishes correctness of the reduction, so it remains to note that the construction of $A'$ can be implemented in polynomial time.

**Proofs for Section 6**

To show Theorem 12, we first observe the following easily proved relationship between $\mathcal{ELT}$ simulations between $(I_1, d)^{\mathcal{ELT}}$ and $(I_2, e)$ and preservation of $\mathcal{ELT}^f$ concepts from $(I_1, d)$ to $(I_2, e)$.

**Lemma 11** Let $I_1, I_2$ have finite outdegree, and let $\Sigma$ be a signature. The following conditions are equivalent:

1. $(I_1, d)^{\mathcal{ELT}} \preceq_{\mathcal{ELT}, \Sigma} (I_2, e)$;
2. for all $\mathcal{ELT}^f(\Sigma)$ concepts $C$: if $d \in C^{I_1}$, then $e \in C^{I_2}$.

We also state and prove the characterization for MSC verification in $\mathcal{ELT}^d$.

**Theorem 19** Let $\mathcal{K} = (T, A)$ be an $\mathcal{ELT} KB$, $a_1, \ldots, a_n \in \text{ind}(A)$, and $\Sigma$ a signature. An $\mathcal{ELT}^d(\Sigma)$ concept $C$ is the $\mathcal{ELT}^d(\Sigma)$-MSC of $a_1, \ldots, a_n$ with respect to $\mathcal{K}$ if, and only if, the following conditions hold:

1. $(a_1, \ldots, a_n) \in C^{\prod_{i=1}^n (\mathcal{U}_{K}, a_i)}$;
2. $(\prod_{i=1}^n (\mathcal{U}_{K}, a_i))^{\mathcal{ELT}^d} \preceq_{\mathcal{ELT}, \Sigma} (\mathcal{U}_{T, C}, \rho_C)$.

**Proof.** The proof if similar to the proof of Theorem 3. By Lemmas 3 and 4, Condition 1 is equivalent to Condition 1 of the definition of MSCs. For Condition 2 observe that by Lemmas 3, 4, and 11, Condition 2 is equivalent to Condition 2 of the definition of MSCs.

**Theorem 12** ($\mathcal{ELT}^d$-MSC Existence w.r.t. $\mathcal{ELT}$ TBoxes) Let $\mathcal{K} = (T, A)$ be an $\mathcal{ELT} KB$, $a_1, \ldots, a_n \in \text{ind}(A)$, and $\Sigma$ a signature. The following are equivalent, for $C_k = (\prod_{i=1}^n (\mathcal{U}_{K}, a_i))^{\mathcal{ELT}^d,s,k}$:

1. the $\mathcal{ELT}^d(\Sigma)$-MSC of $a_1, \ldots, a_n$ w.r.t. $\mathcal{K}$ exists;
2. $C_k$ is the $\mathcal{ELT}^d(\Sigma)$-MSC of $a_1, \ldots, a_n$ w.r.t. $K$, for a $k \geq 0$.
3. $\Pi_{i=1}^n (U_{K_i}, a_i)_{\mathcal{ELT}^d} \preceq_{\mathcal{ELT}^d} (U_{T,K}, c_k)$ for all $k \geq 0$.

The proof is similar to the proof of Theorem 4.

Proof. “$2 \Rightarrow 1$” is trivial. “$3 \Rightarrow 2$” is an immediate consequence of Theorem 19. For “$1 \Rightarrow 3$”, let the $\mathcal{L}(\Sigma)$-MSC $D$ be of depth $k$. It follows from Theorem 19 that
\[
(a_1, \ldots, a_n) \in D_{\Pi_{i=1}^n (U_{K_i}, a_i)}
\]
which implies
\[
(a_1, \ldots, a_n) \in D_{(U_{T,K}, \rho_c)_k}
\]
since $D$ is an $\mathcal{ELT}^d$ concept. As $D$ has depth $k$ and has signature $\Sigma$, $\rho_c \in D_{(U_{T,K}, \rho_c)_k}$.

Now, Point 3 follows from the definition of the MSC and Lemmas 3, 4, and 11.

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

We show hardness for LCS verification and existence at the same time, by reducing from concept subsumption relative to general $\mathcal{ELT}$ TBoxes (Baader, Brandt, and Lutz 2008). Hardness for MSC verification and existence then follows from Theorem 1. Let $T, A, B$ be an input to the subsumption problem. We define a TBox $T'$ by taking fresh role names $r, s$ and fresh concept names $E, F$ and setting for $C := \exists r A \land \exists r A$, $D_0 := \exists r A \land \exists r A$, and $C_1 := \exists r A \land \exists r A$:
\[
T' = T \cup \{ C \sqsubseteq \exists s D_1, D_1 \sqsubseteq \exists s D_1, D_0 \sqsubseteq F, F \sqsubseteq \exists s (\exists r T \sqcap F) \}.
\]
Based on Theorems 1, 19, and 12, one can verify that the following conditions are equivalent:
(a) $T \models A \subseteq B$;
(b) $\exists r T \sqcap F$ is the $\mathcal{ELT}^d$-LCS of $C$ w.r.t. $T'$;
(c) the $\mathcal{ELT}^d$-LCS of $C$ w.r.t. $T'$ exists.

This establishes the claimed lower bounds.

Establishing the upper bounds requires more work. We start with Lemma 5.

Lemma 5 Let $N$ be the outdegree of $\Pi_{i=1}^n U_{K_i}$. Then the $\mathcal{ELT}^d(\Sigma)$-MSC of $a_1, \ldots, a_n$ w.r.t. $K$ exists if, for some subconcept $D$ of $\Pi_{i=1}^n (U_{K_i}, a_i)_{\mathcal{ELT}^d}$, we have:
\[
\Pi_{i=1}^n (U_{K_i}, a_i)_{\mathcal{ELT}^d} \preceq_{\Sigma} (U_{T,D}, \rho_D).
\]

Proof. In the “if”-direction, assume that
\[
\Pi_{i=1}^n (U_{K_i}, a_i)_{\mathcal{ELT}^d} \preceq_{\Sigma} (U_{T,D}, \rho_D)
\]
for some subconcept $D$ of $\Pi_{i=1}^n (U_{K_i}, a_i)_{\mathcal{ELT}^d}$, and let $S$ be a witnessing injective simulation. Let $h$ be the homomorphism from $U_{T,D}$ to $U_{T,D}$ which maps every element to its “original”. It should be clear the relation $S'$ defined by
\[
S' = \{(d, h(e)) \mid (d, e) \in S\}
\]
is an $\mathcal{ELT}(\Sigma)$-simulation between $\Pi_{i=1}^n (U_{K_i}, a_i)_{\mathcal{ELT}^d}$ and $U_{T,D, \rho_D}$. Since $D$ is a subconcept of $\Pi_{i=1}^n (U_{K_i}, a_i)_{\mathcal{ELT}^d}$, it is also a subconcept of $C_k$ where $k$ is the role depth of $D$, and thus
\[
\Pi_{i=1}^n (U_{K_i}, a_i)_{\mathcal{ELT}^d} \preceq_{\Sigma} (U_{T,D}, \rho_D),
\]
By Theorem 12, the MSC exists.

Conversely, suppose the $\mathcal{ELT}^d(\Sigma)$-MSC exists, and thus, there is a $k \geq 0$ with
\[
\Pi_{i=1}^n (U_{K_i}, a_i)_{\mathcal{ELT}^d} \preceq_{\Sigma} (U_{T,D}, \rho_D).
\]
Take $D = C_k$ and let $S$ be the witnessing simulation. It is crucial to observe that, by the definition of $\mathcal{ELT}^d$ unfolding we have the following property:
\[
(*) \text{ for all } (d, e)(d', e') \in S: \text{ if } d' \text{ is a successor of } d \text{ in the tree } \Pi_{i=1}^n (U_{K_i}, a_i)_{\mathcal{ELT}^d} \text{ then } e' \text{ is an successor of } e \text{ in the tree } U_{T,D, \rho_D}.
\]

Intuitively, the simulation always goes “downwards” in the right tree. Based on this insight, we construct an injective simulation $S'$ to $U_{T,D, \rho_D}$ inductively. During the construction, we maintain the invariant that $(d, e) \in S'$ implies $(d, h(e)) \in S$, where $h$ is the homomorphism from $U_{T,D}^N$ to $U_{T,D}$ which maps every element to its “original”.

- Start with $S' = \{(a_1, \ldots, a_n), \rho_D\}$;
- For the inductive step, do the following for every $(d, e) \in S'$: let $d_1, \ldots, d_n$ be the $\Sigma$-successors of $d$ in $\Pi_{i=1}^n (U_{K_i}, a_i)_{\mathcal{ELT}^d}$, then $e'$ is a successor of $e$ in the tree $U_{T,D, \rho_D}$, and add $(d_i, e_i) \in S'$, for all $i$.

It can be verified that the invariant is preserved and that $S'$ is an injective simulation because of $(*)$.

We give now the automata-based approach to deciding the criterion in Lemma 5. We start with providing the necessary preliminaries. An $n$-ary tree is the set $T = \{1, \ldots, n\}^*$. For a node $wi \in T$, we identify $wi - 1$ with $w$. For an alphabet $\Theta$, a $\Theta$-labeled tree is a pair $(T, L)$ with $T$ a tree and $L : T \rightarrow \Theta$ a node labeling function. We also recall the notion of non-deterministic parity tree automata (NTA). An NTA over $n$-ary trees is a tuple $A = (Q, \Theta, q_0, Q, \Delta)$ where $Q$ is a set of states, $\Theta$ is the input alphabet, $q_0 \in Q$ is the initial state, $\Delta \subseteq Q \times \Theta \times Q^N$ is the transition relation, and $Q : Q \rightarrow N$ is the priority function. The semantics of NTAs is defined as usual via runs. A run of an NTA $A = (Q, \Theta, q_0, Q, \Delta)$ over an $n$-ary input $(T, L, r)$ is a $Q$-labeled tree $(T, r)$ such that:
- $r(e) = q_0$, and
- for all $w \in T$, $(r(w), L(w), r(w), \ldots, r(wN)) \in \Delta$.

Let $\gamma = i_0 j_1 \cdots$ be an infinite path in $(T, r)$ and denote, for all $j \geq 0$, with $q_j$ the state such that $r(i_j) = (x, q_j)$. The path $\gamma$ is accepting if the largest number $m$ such that $\Omega(q_j) = m$ for infinitely many $j$ is even. A run $(T, r)$ is
accepting, if all infinite paths in $T_r$ are accepting. The language accepted by $\mathfrak{A}$, denoted $L(\mathfrak{A})$, is the set of all trees $(T, L)$ for which there is an accepting run.

To encode interpretations we use the alphabet $\Theta = 2^{\Theta_0}$ where

$$\Theta_0 = \text{sub}(T) \cup \{r, r^- \mid r \text{ occurs in } T\}.$$ 

A $\Theta$-labeled tree $(T, L)$ represents the interpretation $I_L = (T, \mathcal{I}_L)$ given by

$$A^{\mathcal{I}_L} = \{u \mid A \in L(u)\},$$

$$r^{\mathcal{I}_L} = \{(u, u, -1) \mid r^- \in L(u)\} \cup \{(u, -1, u) \mid r \in L(u)\}$$

for every concept name $A \in \text{sub}(T)$ and role name $r$ that occurs in $T$. Note that the interpretation $I_L$ is not necessarily connected; however, we usually identify $I_L$ with its sub-interpretation induced by all elements reachable from the root. It should be clear that, conversely, for every tree-shaped interpretation $I$ of outdegree $\leq n$, there is an $n$-ary labeled tree $(T, L)$ such that $I$ and $I_L$ are isomorphic.

We also use NTAs over the alphabet $\Theta^2$ in which case an input tree $(T, L)$ is treated as two trees $(T, L_1), (T, L_2)$ and thus encodes two interpretations $I_{L_1}$ and $I_{L_2}$. Finally, we treat $I_L$ as a concept if it is finite and has no multiedges.

**Lemma 12** Let $N$ be the outdegree of $\Pi^n_{i=1} U_k$.

1. There is an NTA $\mathfrak{A}$ such that, for all $N^2$-ary $\Theta$-labeled trees $(T, L)$, we have:

   $$(T, L) \in L(\mathfrak{A}) \iff (\Pi^n_{i=1} (U_k, a_i))^{\mathcal{I}_L} \models_{\mathcal{I}_L} \models_{\mathcal{I}_L} (T, L)\varepsilon.$$ 

2. There is an NTA $\mathfrak{B}_0$ over $N^2$-ary $\Theta^2$-labeled trees such that for every $(T, L) \in L(\mathfrak{B}_0)$, there is a subconcept $D$ of $(\Pi^n_{i=1} (U_k, a_i))^{\mathcal{I}_L} \varepsilon$ such that:

   (a) $I_{L_1}$ is the concept $D$, and
   (b) $I_{L_2}$ is isomorphic to $\mathcal{U}_{T, D}$.

   Conversely, for every subconcept $D$ of $(\Pi^n_{i=1} (U_k, a_i))^{\mathcal{I}_L} \varepsilon$ there is some $(T, L) \in L(\mathfrak{B}_0)$ such that $I_{L_1}$ is $D$.

Moreover, $\mathfrak{A}$ and $\mathfrak{B}_0$ can be constructed in time exponential in $|K|$.

In order to prove this lemma, we need a concrete definition of $(\Pi^n_{i=1} (U_k, a_i))^{\mathcal{I}_L} \varepsilon$. Let $\text{tp}(T)$ denote the set of all types for $T$, and $\text{rol}(T)$ be the set of roles that occur in $T$. For each $r \in \text{rol}(T)$, we define a relation $\rightarrow_r$ on the set $U := D^n \times (\{\varepsilon\} \cup \text{rol}(T))$ for

$$D = \text{ind}(A) \cup \text{tp}(T)$$

by taking $(x_1, \ldots, x_n, x) \rightarrow_r (y_1, \ldots, y_n, y)$ iff

- $y = r$ and $x \neq r^-$, and
- for every $i \in \{1, \ldots, n\}$, one of the following is satisfied:
  (i) $x_i, y_i \in \text{ind}(A)$ and $r(x_i, y_i) \in A$
  (ii) $x_i \in \text{ind}(A), y_i \in \text{tp}(T)$, and $a_i \rightarrow_r^{\mathcal{I}_L} y_i$;
  (iii) $x_i, y_i \in \text{tp}(T)$ and $x_i \rightarrow_r^{\mathcal{I}_L} y_i$.

Some element $(x_1, \ldots, x_n, x) \in U$ satisfies a concept name $A$ if for every $i \in \{1, \ldots, n\}$, either $x_i \in \text{tp}(T)$ and $A \models (x_i, x)$, or $x_i \in \text{ind}(A)$ and $A \models (x_i, x)$. A path is a sequence $\pi_0 \pi_1 \cdots \pi_{n-1} \pi_n$ such that $\pi_0 = (a_1, \ldots, a_n, \varepsilon), \pi_{i-1} \rightarrow_r \pi_i$ for all $i \in \{1, \ldots, n\}$. We denote with PATHS the set of all paths, and with tail($p$) the last element in the sequence $p$. It can be verified that the interpretation $\langle U, (a_1, \ldots, a_n, \varepsilon) \rangle$ with $U$ defined below is isomorphic to $(\Pi^n_{i=1} (U_k, a_i))^{\mathcal{I}_L} \varepsilon$:

$$\Delta'' = \text{PATHS}$$

$$A = \{p \mid \text{tail}(p) \text{ satisfies } A\}$$

$$r = \{(p, pr_u) \mid pr_u' \in \text{PATHS}\} \cup \{(pr^{-u}, p) \mid pr^{-t} \in \text{PATHS}\}.$$ 

**Construction of $\mathfrak{A}$** Informally, the automaton $\mathfrak{A}$ simulates the definition of $\mathcal{U}$ by keeping in its state only the tail of the current path.

More formally, the set $Q$ of states is the smallest set that contains $q_0^T$ and $(a_1, \ldots, a_n, \varepsilon)$ and is closed under the relations $\rightarrow_r$ defined above, that is, if $u \in Q$ and $u \rightarrow_r u'$ for some $r \in \text{rol}(T)$, then $u' \in Q$. The initial state is $(a_1, \ldots, a_n, \varepsilon)$, and the transition relation contains $(q_{\varepsilon}^T, \theta, q_1^T)$, for all $\theta \in \Theta$, and

$$((x_1, \ldots, x_n, x), \theta, q_1, \ldots, q_{N\varepsilon})$$

whenever:

- if $(x_1, \ldots, x_n, x)$ satisfies $A$ then $A \in \theta$, for all $A \in \Sigma$;
- if $x \neq \varepsilon$, then $x \in \theta$;
- each $(y_1, \ldots, y_n, r)$ such that $(x_1, \ldots, x_n, x) \rightarrow_r (y_1, \ldots, y_n, r)$ for some $r \in \text{rol}(T)$ occurs precisely once in $q_{1}, \ldots, q_{N\varepsilon}$; all other $q_i$ are $q_{\varepsilon}^T$.

It is routine to verify that $\mathfrak{A}$ as is required.

**Construction of $\mathfrak{B}_0$** The automaton $\mathfrak{B}_0$ is the intersection of three automata $\mathfrak{A}_0, \mathfrak{A}_1, \mathfrak{A}_2$.

Automaton $\mathfrak{A}_0$ verifies Condition (a) from the Lemma by ensuring that $I_{L_1}$ is a $\Sigma$-concept, that is, finite and without multiedges, and in fact, a subconcept of $(\Pi^n_{i=1} (U_k, a_i))^{\mathcal{I}_L} \varepsilon$.

Realizing this condition as an NTA is relatively straightforward; details are thus omitted.

The automata $\mathfrak{A}_1, \mathfrak{A}_2$ together verify Condition (b) from the Lemma, assuming that $I_{L_1}$ is some concept $C$. The first, $\mathfrak{A}_1$, ensures that for all $n \in \Delta^{L_2}$ with $L_1(n) \neq \emptyset$ and all $D \in \text{sub}(T)$, we have

$$D(a) \in L_2(n) \iff T, T \models D(a).$$

$$r \in L_2(n) \iff r \in L_1(n).$$

Thus, on the elements in $I_{L_1}$, the interpretation $I_{L_2}$ is the universal model of $I_{L_1}$ and $T$. Based on this, $\mathfrak{A}_2$ just generates (in $L_2$) the trees induced in the universal model below the elements in $\Delta^{L_1}$ by simulating $\rightarrow^T_r$, which is again standard and omitted.
It is rather tedious to specify the automaton $A_1$ ensuring (2) directly as an NTA. Instead, we specify $A_1$ as a two-way alternating tree automata (TWAPA) relying on the fact that every TWAPA can be transformed into an equivalent NTA under an exponential blowup (Vardi 1998).

**Two-way Alternating Tree Automata** A two-way alternating parity tree automaton over $k$-ary trees (TWAPA) is a tuple $A = (Q, \Theta, q_0, \delta, \Omega)$ where $Q$ is a finite set of states, $\Theta$ is the input alphabet, $q_0 \in Q$ is the initial state, $\delta$ is a transition function, and $\Omega : Q \rightarrow \mathbb{N}$ is a priority function (Vardi 1998). The transition function $\delta$ maps every state $q$ and input letter $\theta \in \Theta$ to a positive Boolean formula $\delta(q, \theta)$ over the truth constants true and false and transition atoms of the form $(i, q) \in [k] \times Q$, where $[k] = \{ -1, 0, 1, \ldots, k \}$. The semantics is given in terms of runs. More precisely, let $(T, L)$ be a $\Theta$-labeled tree and $A = (Q, \Theta, q_0, \delta, \Omega)$ a TWAPA. A run of $A$ over $(T, L)$ is a $(T \times Q)$-labeled tree $(T_r, r)$ such that:

1. $r(\varepsilon) = (\varepsilon, q_0)$, and
2. for all $y \in T_r$, with $r(y) = (x, q)$, there is a subset $S \subseteq [k] \times Q$ such that $S \models \delta(q, L(x))$ and for every $(i, q') \in S$, there is some successor $y'$ of $y$ in $T_r$ with $r(y') = (x', q')$.

Let $\gamma = i_0i_1 \cdots$ be an infinite path in $T_r$ and denote, for all $j \geq 0$, with $q_j$ the state such that $r(i_j) = (x, q_j)$. The path $\gamma$ is accepting if the largest number $m$ such that $\Omega(q_j) = m$ for infinitely many $j$ is even. A run $(T_r, r)$ is accepting, if all infinite paths in $T_r$ are accepting. $A$ accepts a tree if $A$ has an accepting run over it.

Before we can give the TWAPA, we need some preliminary notions, in particular a syntactic characterization of whether $T, A \models C(a)$, which can be easily be implemented in a TWAPA. Similar characterizations have been used before, e.g., in (Jung et al. 2017).

**Derivation Trees** Fix an $ELI$ knowledge base $K = (T, A)$, $a_0 \in \text{ind}(A)$, and $C \subseteq \text{sub}(T)$. A derivation tree for an assertion $C_0(a_0) \in A$ w.r.t. $T$ is a finite $\text{ind}(A) \times \text{sub}(T)$-labeled tree $(T, V)$ such that:

- $V(\varepsilon) = (\varepsilon, C_0)$;
- if $V(n) = (a, C)$ and neither $C(a) \in A$ nor $T \models C \subseteq C$, one of the following holds:
  (i) $n$ has successors $n_1, \ldots, n_k$, $k \geq 1$ with $V(n_i) = (a, C_i)$, for $1 \leq i \leq k$ and $T \models C_1 \sqcap \ldots \sqcap C_k \subseteq C$;
  (ii) $n$ has a single successors $n'$ with $V(n') = (b, C')$ such that $r(a, b) \in A$ and $T \models \exists r. C' \subseteq C$.

**Lemma 13** $T, A \models C_0(a_0)$ iff there is a derivation tree for $C_0(a_0)$ in $A$ w.r.t. $T$.

**Proof.** (⇒) is clear.

For (⇒), we construct a sequence of ABoxes $A = A_0, A_1, \ldots$, by obtaining $A_{i+1}$ from $A_i$ by applying one of the following two rules:

1. if $C_1(a), \ldots, C_k(a) \in A$ and $T \models C_1 \sqcap \ldots \sqcap C_k \subseteq C$ for some $C \in \text{sub}(T)$, then add $C(a)$ to $A$;
2. if $r(a, b), C'(b) \in A$, and $T \models \exists r. C' \subseteq C$ for some $C \in \text{sub}(T)$, then add $C(a)$.

Note that the sequence is finite, and denote with $A^*$ the final ABox.

**Claim.** There is a model $I$ of $A^*$ and $T$ such that $a \in C^I$ implies $C(a) \in A^*$, for all $a \in \text{ind}(A)$ and $C \in \text{sub}(T)$.

**Proof of the Claim.** Start with an interpretation $I_0$ defined by:

$$\Delta_{I_0}^\emptyset = \text{ind}(A)$$

$$A_{I_0}^\emptyset = \{ a \mid A(a) \in A^* \}$$

$$r_{I_0}^\emptyset = \{ (a, b) \mid r(a, b) \in A \}$$

Denote with $A_n^*$ the set of all $C(a)$ in $A^*$. Now extend $I_0$ as follows: For every $a \in \text{ind}(A)$ and every $C \subseteq \exists r. C' \in T$ such that $T, A_n^* \models C(a)$, add the $r$-successor of $a$ satisfying $C'$ in $U_{T, A^*}$ as an $r$-successor of $a$ to $I$.

This finishes the construction of $I$, and it can be verified that it indeed satisfies the requirements of the claim. This finishes the proof of the claim.

Now suppose $T, A \models C_0(a_0)$. By the Claim, we have $C_0(a_0) \in A^*$. Exploiting that the two rules to construct $A_0, A_1, \ldots$ are in one-to-one correspondence with Conditions (i) and (ii) from the definition of derivation trees, we can inductively construct a derivation tree for $C_0(a_0)$ in $A$ w.r.t. $T$.

We are now in the position to construct the automaton $A_1$. It ensures that when a $\Theta^2$-labeled tree $(T, L)$ is accepted, then for all $n \in \Delta T^2_h$ with $L_1(n) \neq \emptyset$, Condition (3) is satisfied, and for all concepts $D \in \text{sub}(T)$:

$$D \in L_2(n) \text{ iff there is a derivation tree for } D(n) \text{ in } L_1,$$

viewed as ABox.

By Lemma 13, this condition ensure that Equation (2) above is satisfied. We take $A_1 = (Q, \Theta, q_0, \delta, \Omega)$ where

$$Q = \{ q_0, q_0' \} \cup \{ q_D, \overline{q}_D \mid D \in \text{sub}(T) \} \cup \{ q_r, \overline{q}_r, q_{r, D}, \overline{q}_{r, D} \mid r \in \text{rol}(T), D \in \text{sub}(T) \}$$

and $\Omega$ assigns zero to all states, except for states of the form $q_D$, to which it assigns one.

For Condition (∗), we use states $q_D$ for the “⇒” part, and states $\overline{q}_D$ for the “⇐” part. Intuitively, a state $q_D$ assigned to some node $n$ is an obligation to verify the existence of a derivation tree for $D(n)$. Conversely, $\overline{q}_D$ is the obligation that there is no such derivation tree. The automaton starts with the following transitions for every $\theta = (\theta_1, \theta_2)$:

$$\delta(q_0, \theta) = \begin{cases} 
\text{true} & \text{if } \theta_1 = \emptyset \\
q_0' & \text{if } \theta_1 \neq \emptyset \text{ and } \theta_2 \cap \text{rol}(T) = \emptyset \cap \text{rol}(T) \\
\text{false} & \text{otherwise}
\end{cases}$$

$$\delta(q_0, \theta) = \bigwedge_{i=1}^{N^2} (i, q) \wedge \bigwedge_{D \in \theta_2 \cap \text{sub}(T)} q_D \wedge \bigwedge_{D \in \text{sub}(T) \setminus \theta_2} \overline{q}_D$$

For states $q_D$, we implement Conditions (i) and (ii) of derivation trees as transitions. Finiteness of the derivation tree is ensured by the priority assigned to the states. More
\[ \delta(q_D, \theta) = \bigvee_{T|\models D} ((0, q_{D_1}) \land \cdots \land (0, q_{D_N})) \lor \bigvee_{T|\models \exists r.D|\models D} (((-1, q_{D'}) \land q_r) \lor \bigvee_{i=1}^{N^2} (i, q_{r.D'})) \]

Finally, the transitions for \( q_r \) and \( q_{r.D} \) are as follows:

\[ \delta(q_r, \theta) = \begin{cases} \text{true} & \text{if } r \in \theta_1 \\ \text{false} & \text{otherwise} \end{cases} \]

\[ \delta(q_{r.D}, \theta) = (0, q_r) \land (0, q_D) \]

The transitions for \( \eta_D, \eta_r, \) and \( \eta_{r.D} \) are obtained by dualizing the ones for \( q_D, q_r, q_{r.D} \). More precisely, for every such \( q \), we define \( \delta(\eta, \theta) = \overline{\delta(q, \theta)} \), where \( \overline{\gamma} \) is obtained from \( \varphi \) by exchanging \( \land \) with \( \lor \), true with false, and replacing every state \( p \) with \( \overline{p} \). This finishes the construction of \( \mathfrak{A}_1 \). Based on the provided explanations it can be shown that it verifies Condition (2) from above, that is, it computes locally \( \phi \) from \( \theta \).

Observe first that \( (b) \implies (c) \) is trivial. For \( (a) \implies (b) \), suppose \( T \models A \subseteq B \). We show via Theorem 3 that \( A \) is the \( \mathcal{ELI} \)-MSC of \( a \) w.r.t. \( K \). Condition 1 is satisfied since \( A(a) \in A \). For Condition 2, note first that the universal model \( \mathcal{U}_{T',A} \) has the following structure:

- its root \( \rho_A \) satisfies \( A, B \) and has an \( r \)-successor \( b \) satisfying \( E \);
- \( b \) is the root of an infinite binary tree in which each node has an \( r \)-successor and an \( r^{-} \)-successor satisfying \( E \);
- every node in the binary tree has an \( r^{-} \)-successor satisfying \( A \);
- below each element satisfying \( A \) we find a copy of \( \mathcal{U}_{T',A} \) itself.

It is now straightforward to define an \( \mathcal{ELI} \) simulation between \( (\mathcal{U}_K, a) \) and \( (\mathcal{U}_{T',A}, \rho_A) \).

For \( (c) \implies (a) \), suppose \( C \) is an \( \mathcal{ELI} \)-MSC of \( a \) w.r.t. \( K \). By Theorem 4, there is some \( k \geq 0 \) such that \( (\mathcal{U}_K, a) \preceq \mathcal{ELI} (\mathcal{U}_{T',C_k}, \rho_{C_k}) \). Note that there is an infinite \( r \)-path starting from \( a \) in \( \mathcal{U}_K \). However, if \( T \not\models A \subseteq B \), there is no infinite \( r \)-path starting at \( \rho_{C_k} \) in any \( \mathcal{U}_{T',C_k} \), a contradiction.

For 2-ExpTime-hardness in the general case, we reduce the complement of single example \( \mathcal{ELI} \) concept separability which has been shown to be 2-ExpTime-hard in (Gutiérrez-Basulto, Jung, and Sabellek 2018).

**Theorem 20.** Let \( \mathcal{L} \in \{ \mathcal{ELI}, \mathcal{ELC} \} \). Then the complement of single individual \( \mathcal{L} \) concept separability can be reduced in polynomial time to single example \( \mathcal{L} \)-MSC verification and to single example \( \mathcal{L} \)-MSC existence.

**Proof.** The basic argument is very similar to the proof of Theorem 8. The difference is that the proof of Theorem 8 relies on the fact that at least two positive examples are given in the learning instance which are then used to generate \( w \)-chains using distinct concept names \( B_1, B_2 \) which cannot occur in the MSC. Here, instead of using two positive examples we make use of the signature restriction: the concept name generating the \( w \)-chains is not in the signature \( \Sigma \).

Let \( \mathcal{L} \in \{ \mathcal{ELI}, \mathcal{ELC} \} \). We reduce single individual \( \mathcal{L} \) concept separability. We can assume without loss of generality that it is formulated as follows: given an \( \mathcal{L} \) TBox \( T \), a signature \( \Sigma \), and an ABox \( A = \{ A(a), B(b) \} \), where \( A, B \) are concept names with \( B \in \Sigma \), decide whether it is the case that \( (\mathcal{U}_K, a) \preceq \mathcal{ELI} (\mathcal{U}_K, b) \), where \( K = (T, A) \). We use the notation introduced in the proof of Theorem 8 and start with \( \mathcal{E} \). Construct a new TBox

\[ T' = T \cup \{ B_i \subseteq \exists v.E_i \mid i = 1, 2 \} \]

where \( v \) is a fresh role name \( v \) and \( B_1, B_2 \) are fresh concept names. Take a fresh role name \( v \) and construct an ABox \( A' \) with individuals \( \rho, a, b \), and assertions

\[ \forall \rho, a, \forall \rho, b, A(a), B(b), B_1(a), B_2(b) \]
Let \( C_0 = \exists v.(B \cap B_2) \), set \( \Sigma' = \Sigma \cup \{ w, v, B_3 \} \), and let \( K' = (T', A') \). We show that the following conditions are equivalent:

(a) \( (U_K, a) \preceq_{\mathcal{EL}, \Sigma} (U_K, b) \);
(b) \( C_0 \) is the \( \mathcal{EL}(\Sigma') \)-MSC of \( \rho \) w.r.t. \( K' \);
(c) The \( \mathcal{EL}(\Sigma') \)-MSC of \( \rho \) w.r.t. \( K' \) exists.

(a) \( \Rightarrow \) (b). Assume (a) holds. By Theorem 3, it suffices to show

1. \( \rho \in C_0^\mathcal{EL}_K \);
2. \( (U_K', \rho) \preceq_{\mathcal{EL}, \Sigma'} (U_T', G_0, \rho G_0) \).

Condition 1 is by construction. For Condition 2, it suffices to show that

\[
(\mathcal{EL}, d) \preceq_{\mathcal{EL}, \Sigma'} (\mathcal{EL}, (B \cap B_3), (B \cap B_3))
\]

for \( d \in \{ a, b \} \). For \( d = b \) this is trivial and for \( d = a \) this follows directly from (a) and the construction of \( K' \) and \( \Sigma' \).

(b) \( \Rightarrow \) (c) is trivial.

(c) \( \Rightarrow \) (a) We use Theorem 4. There exists \( k \geq 0 \) such that for \( G_k = (U_K', \rho) \preceq_{\mathcal{EL}, \Sigma'} (U_T', G_k, \rho G_k) \).

Then, \( a, b \) are in the domain of \( G_k \) and

\[
(\mathcal{EL}, a) \preceq_{\mathcal{EL}, \Sigma'} (\mathcal{EL}, (B \cap B_3), (B \cap B_3))
\]

for either \( d = a \) or \( d = b \). But the assumption \( d = a \) leads to a contradiction as there is an \( w \)-chain starting at \( a \) in \( U_K \) but there is no such \( w \)-chain starting at \( b \) in \( U_T \).

The transition function \( \delta \) of \( \mathcal{EL} \) assigns \( \delta(q_x, \theta) = \) false whenever \( t \cap \Sigma \not\subseteq \theta \) and \( t \cap \Sigma \not\subseteq \theta \), respectively. Otherwise, we set:

\[
\delta(q_x, \theta) = \bigwedge_{t \in \text{cIs}(\mathcal{EL})} (0, p_{r,t}) \vee \bigvee_{i \leq \theta_i} (i, g_{r,t})
\]

The transition function for states of the form \( p_{r,x}, q_{r,x} \) is defined by taking, for all \( \theta \):

\[
\delta(p_{r,x}, \theta) = \begin{cases} \text{false} & \text{if } r^+ \not\subseteq \theta \\ \text{true} & \text{if } r^- \subseteq \theta \end{cases}
\]

\[
\delta(q_{r,x}, \theta) = \begin{cases} \text{false} & \text{if } r^- \not\subseteq \theta \\ \text{true} & \text{if } r^+ \subseteq \theta \end{cases}
\]

It is not hard to verify that \( (T, L) \in L(\mathcal{EL}) \) if \( U_{K,a} \preceq_{\mathcal{EL}, \Sigma} I_{L}, \varepsilon \), for all \( (T, L) \), even without the assumption \( I_L = T \).

Moreover, the size of \( \mathcal{EL} \) is exponential in \( |K| \) and it can be computed in exponential time.

For the full signature case, we obtain \( \mathcal{G} \) from \( \mathcal{EL} \) by dropping the states of the shape \( q_x, q_{r,t}, p_{r,t} \), for all types \( t \). The transition function \( \delta' \) is obtained from \( \delta \) given above by replacing every atom \( (i, q) \) for a dropped state \( q \) with true. It is routine to verify that \( (T, L) \in L(\mathcal{G}) \) if \( U_{K,a} \preceq_{\mathcal{EL}, \Sigma} I_{L}, \varepsilon \), whenever \( I_L = T \).

Moreover, the size of \( \mathcal{G} \) is polynomial in \( |K| \) and it can be computed in exponential time.

For the full signature case, we obtain \( \mathcal{N} \) from \( \mathcal{EL} \) by the standard translation of TWA PAs to NTA s, incurring an exponential blow-up (Vardi 1998). 

To finish the upper bound from Theorem 15, let \( \mathcal{B} \) be the projection of \( \mathcal{B}_0 \) from Lemma 12 to its second component. It is routine to verify that the \( \mathcal{EL}(\Sigma') \)-MSC of \( \rho \) w.r.t. \( K' \) exists if \( \mathcal{I}_L \models T \) with \( \mathcal{I}_L \models T \).

Since all automata can be computed in double exponential time, and since intersection (non-)emptiness can be verified in PTIME, the \( 2 \) \text{-EXPTIME} upper bound follows. For the full signature case, we obtain an \( \text{EXPTIME} \) upper bound since in this case \( \mathcal{L} \) and \( \mathcal{B} \) can be computed in exponential time.