

Matching in the Description Logic $\mathcal{EL}$ without the Top Concept

Puneetha Jangir and Oliver Fernández Gil

Theoretical Computer Science, TU Dresden, Germany
puneethajangir297@gmail.com, oliver.fernandez@tu-dresden.de

Abstract

Matching and unification have been introduced in Description Logic as non-standard reasoning problems with applications in ontology engineering. For the DL $\mathcal{EL}$, it was shown that both problems are NP-complete, but some special cases of matching are tractable. Surprisingly, it turned out that for unification the complexity increases to PSpace if one disallows the use of the top concept to build concept descriptions. However, the effect of disallowing the top concept has not been investigated for matching. We show that a similar increase in complexity applies to matching when the top concept is not allowed.

1 Introduction

Matching of concept descriptions against concept patterns was introduced in Description Logics (DLs) as a non-standard reasoning problem almost 30 years ago. The original motivation for considering matching in DLs comes from [14], where it was used to filter out unimportant aspects of large concepts appearing in knowledge bases. For the DL $\mathcal{EL}$, which has the concept constructors conjunction ($\sqcap$), existential restriction ($\exists r.C$), and top concept ($\top$), matching was first investigated in [5, 18]. In these papers, two variants of matching were considered: *matching modulo subsumption* and *matching modulo equivalence*. Matching modulo subsumption is a problem of the form $C \sqsubseteq^? D$ where C is a concept description and D is a concept pattern (i.e., a concept containing variables). A matcher is a substitution σ satisfying $C \sqsubseteq \sigma(D)$, i.e., C is *subsumed* by $\sigma(D)$. For example, $\theta := \{X \mapsto A\}$ is a matcher of $\exists r.(A \sqcap B) \sqsubseteq^? \exists r.X$. As for matching modulo equivalence, it is a problem of the form $C \equiv^? D$, and a matcher is a substitution σ such that $C \equiv \sigma(D)$. In this case, θ is not a matcher of $\exists r.(A \sqcap B) \equiv^? \exists r.X$ since $\exists r.(A \sqcap B)$ and $\exists r.A$ are not equivalent concept descriptions.

In [5, 18], it was shown that deciding the existence of a matcher in $\mathcal{EL}$ is polynomial for matching modulo subsumption, but NP-complete for matching modulo equivalence. More recently, matching in $\mathcal{EL}$ has been investigated in the presence of a background $\mathcal{EL}$ -TBox [9], where the subsumption relation $\sqsubseteq$ and the equivalence relation $\equiv$ are interpreted modulo the axioms in the TBox. In addition, two new versions of matching were considered in [9]: the dual of matching modulo subsumption (i.e., the concept pattern appears on the left side of $\sqsubseteq^?$) and a generalization of all the versions of matching. For matching modulo subsumption and its dual, it is shown that the problem is tractable in the presence of a TBox, while the general version of matching w.r.t. TBoxes is shown to be NP-complete.

Unification of concept descriptions is the generalization of matching where C and D are both allowed to contain variables. In the context of DLs, unification was introduced in [11] as an inference task that can, for example, be used to detect redundancies in ontologies. For the DL $\mathcal{EL}$, unification was first investigated in [8], where it was proved that deciding the existence of a unifier is an NP-complete problem. Later, it was shown in [1] that unification in $\mathcal{EL}$ remains in NP in the presence of a background TBox, if such a TBox satisfies a condition called *cycle-restrictedness*. However, in contrast to matching, decidability of unification w.r.t. arbitrary

TBoxes is still an open problem. The unification problem has also been investigated for the DL $\mathcal{EL}^{-\top}$, i.e., the fragment of $\mathcal{EL}$ in which the top constructor is disallowed [12]. Surprisingly, it turned out that the complexity of deciding unification increases from NP-complete for $\mathcal{EL}$ to PSpace-complete for $\mathcal{EL}^{-\top}$. This result has been recently combined with the result in [1], to show that unification in $\mathcal{EL}^{-\top}$ remains in PSpace in the presence of cycle-restricted TBoxes [3].

Motivated by these results, we were curious to know if the absence of the top concept has a similar effect on the complexity of the matching problem. This is indeed the case, already for the case of matching modulo subsumption. More precisely, the complexity of matching increases from PTime for $\mathcal{EL}$ to NP-complete for $\mathcal{EL}^{-\top}$ (without TBoxes), and to PSpace-complete in the presence of arbitrary TBoxes. The purpose of this abstract is to report on these new results, which have been obtained in the master's thesis of Puneetha Jangir [16].

2 Matching in $\mathcal{EL}$ and $\mathcal{EL}^{-\top}$

Let N_C and N_R be sets of *concept* and *role* names, respectively. The set of $\mathcal{EL}$ -concept descriptions (for short, *concepts*) is built, starting with N_C and N_R , by using the constructors *conjunction* ($\sqcap$), *existential restriction* ($\exists r.C$), and *top* ($\top$). An $\mathcal{EL}$ -TBox is a finite set of *general concept inclusions* (GCIs) $C \sqsubseteq D$, where C and D are $\mathcal{EL}$ -concepts. The DL $\mathcal{EL}^{-\top}$ is the fragment of $\mathcal{EL}$ where the constructor $\top$ is not available to build concepts. In an $\mathcal{EL}^{-\top}$ -TBox, the concepts occurring in GCIs must be $\mathcal{EL}^{-\top}$ -concepts.

The semantics of $\mathcal{EL}$ -concepts and TBoxes is defined using the notion of an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, which has a non-empty domain $\Delta^{\mathcal{I}}$, and assigns subsets $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ to each concept name $A \in N_C$ and binary relations $r^{\mathcal{I}}$ over $\Delta^{\mathcal{I}}$ to each role name $r \in N_R$. The function $\cdot^{\mathcal{I}}$ is extended to arbitrary $\mathcal{EL}$ -concepts in the usual way (see [2] for details). The interpretation $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all $C \sqsubseteq D \in \mathcal{T}$.

Given an $\mathcal{EL}$ -TBox $\mathcal{T}$ and $\mathcal{EL}$ -concepts C, D , we say that C is *subsumed* by D w.r.t. $\mathcal{T}$ (written as $C \sqsubseteq_{\mathcal{T}} D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$. They are *equivalent* w.r.t. $\mathcal{T}$ (written as $C \equiv_{\mathcal{T}} D$) if $C \sqsubseteq_{\mathcal{T}} D$ and $D \sqsubseteq_{\mathcal{T}} C$. In the presence of the empty TBox, we will simply write $C \sqsubseteq D$ and $C \equiv D$. Subsumption (and thus also equivalence) between $\mathcal{EL}$ -concepts w.r.t. arbitrary TBoxes can be decided in polynomial time [2].

To define matching, we consider a new set N_V of concept names, disjoint from N_C and N_R . The symbols in N_V are called *concept variables* and the symbols in N_C are called *concept constants*. Given $\mathcal{L} \in \{\mathcal{EL}, \mathcal{EL}^{-\top}\}$, an $\mathcal{L}$ -*concept pattern* is an $\mathcal{L}$ -concept defined over $N_C \cup N_V$ and N_R . An $\mathcal{L}$ -*substitution* σ is a mapping from a finite subset of N_V to the set of $\mathcal{L}$ -concept patterns. The extension of σ to arbitrary concept patterns is defined inductively in the usual way. An $\mathcal{L}$ -concept pattern is *ground* if it does not contain variables.

Definition 1 ([9]). *Let $\mathcal{T}$ be an $\mathcal{L}$ -TBox.¹ An $\mathcal{L}$ -matching problem w.r.t. $\mathcal{T}$ is a finite set $\Gamma = \{C_1 \sqsubseteq^? D_1, \dots, C_n \sqsubseteq^? D_n\}$ of subsumption constraints between $\mathcal{L}$ -concept patterns, where for each $i, 1 \leq i \leq n$, C_i or D_i is ground. An $\mathcal{L}$ -substitution σ is an $\mathcal{L}$ -matcher of Γ w.r.t. $\mathcal{T}$ if σ solves all constraints in Γ w.r.t. $\mathcal{T}$, i.e., if $\sigma(C_1) \sqsubseteq_{\mathcal{T}} \sigma(D_1), \dots, \sigma(C_n) \sqsubseteq_{\mathcal{T}} \sigma(D_n)$.*

We also consider the following special cases of the matching problem introduced above:

- The $\mathcal{L}$ -matching problem Γ is a *matching problem modulo equivalence* if $C \sqsubseteq^? D \in \Gamma$ implies that $D \sqsubseteq^? C \in \Gamma$.²

¹Note that the GCIs in $\mathcal{T}$ are built using $\mathcal{L}$ -concepts. Thus, $\mathcal{T}$ does not contain variables.

²As described in the introduction, the original definition of matching modulo equivalence defined Γ as a finite set of constraints of the form $C \equiv^? D$ [6, 5].

- The $\mathcal{L}$ -matching problem Γ is a *left-ground matching problem* if $C \sqsubseteq^? D \in \Gamma$ implies that C is ground. This corresponds to the original notion of matching modulo subsumption.
- The $\mathcal{L}$ -matching problem Γ is a *right-ground matching problem* if $C \sqsubseteq^? D \in \Gamma$ implies that D is ground.

We are interested in the following decision problem: Given an $\mathcal{L}$ -matching problem Γ and an $\mathcal{L}$ -TBox $\mathcal{T}$, does Γ has an $\mathcal{L}$ -matcher w.r.t. $\mathcal{T}$? The complexity of deciding matching in $\mathcal{EL}$ has been determined for the four classes of matching problems introduced above. The first results showed that, w.r.t. the empty TBox, left-ground matching can be decided in PTime and matching modulo equivalence is an NP-complete problem [5, 18]. These results were later extended in [9] to take into account the other two classes of matching problems, as well as non-empty $\mathcal{EL}$ -TBoxes. It was shown that, w.r.t. arbitrary TBoxes, left-ground matching and right-ground matching are both decidable in PTime, whereas NP-completeness of matching modulo equivalence also applies to the general case.

For right-ground matching, we can reuse the arguments presented in [9] to show that for $\mathcal{EL}^{-\top}$ the problem is also in PTime. The next two sections give an overview of our complexity results for the remaining cases of matching in $\mathcal{EL}^{-\top}$. In Section 3, we show that the complexity of matching increases (already for the class of left-ground matching problems), if one disallows the use of $\top$. In Section 4, we sketch a general decision procedure for matching in $\mathcal{EL}^{-\top}$ and obtain matching complexity upper bounds.

Theorem 2. *Deciding right-ground matching in $\mathcal{EL}^{-\top}$ w.r.t. arbitrary TBoxes is in PTime.*

3 Lower bounds

We show that deciding left-ground matching in $\mathcal{EL}^{-\top}$ becomes NP-hard w.r.t. $\mathcal{T} = \emptyset$ and PSpace-hard w.r.t. arbitrary TBoxes. Note that left-ground matching can be reduced (in linear time) to matching modulo equivalence. That is, a substitution σ solves the constraint $C \sqsubseteq^? D$ w.r.t. $\mathcal{T}$ iff it solves the constraints $C \sqsubseteq^? C \sqcap D$ and $C \sqcap D \sqsubseteq^? C$ w.r.t. $\mathcal{T}$.³ Hence, the lower bounds transfer to matching modulo equivalence and the general case.

We first sketch how to show NP-hardness. The reduction is from the NP-complete problem 3SAT [15]. Given a 3CNF formula φ , we construct a left-ground matching problem Γ_φ such that φ is satisfiable iff Γ_φ has an $\mathcal{EL}^{-\top}$ -matcher. The matching problem Γ_φ is defined as the union $\Gamma_1 \cup \Gamma_2$. The idea is to define Γ_1 such that its set of $\mathcal{EL}^{-\top}$ -matchers captures all possible truth assignments over the variables occurring in φ . As for Γ_2 , its purpose is to simulate the satisfaction relation between a truth assignment and the formula φ .

Let $\varphi = c_1 \wedge \dots \wedge c_m$ be a 3CNF formula and let $x_1, \dots, x_n$ be the propositional variables of φ . For each x_i , we introduce two concept variables X_i and $\bar{X}_i$. They represent the literals x_i and $\neg x_i$, respectively. Further, we use the concept constants A_t and A_f to represent the truth values *true* and *false*, respectively. Then, we define Γ_1 as the set of constraints $\{C \sqsubseteq^? D_1, \dots, C \sqsubseteq^? D_n, E \sqsubseteq^? F_1, \dots, E \sqsubseteq^? F_n\}$, where:

$$\begin{aligned} C &:= \exists r.(A_t \sqcap \exists r.A_f) \sqcap \exists r.(A_f \sqcap \exists r.A_t) & E &:= A_t \sqcap A_f \\ D_i &:= \exists r.(X_i \sqcap \exists r.\bar{X}_i) & F_i &:= X_i \sqcap \bar{X}_i. \end{aligned}$$

³This equivalence was pointed out in [6] for $\mathcal{T} = \emptyset$. Note that it does not apply to right-ground matching. In such a case, D is ground and C may contain variables. Hence, the resulting constraint $C \sqsubseteq^? C \sqcap D$ need not be as required in Definition 1.

The correspondence between matchers of Γ_1 and truth assignments of $\{x_1, \dots, x_n\}$ is established by showing that the set of $\mathcal{EL}^{-\top}$ -matchers of Γ_1 consists of all substitutions σ such that:

$$\text{for all } i \in \{1, \dots, n\} : (\sigma(X_i) \equiv A_t \wedge \sigma(\bar{X}_i) \equiv A_f) \quad \text{or} \quad (\sigma(X_i) \equiv A_f \wedge \sigma(\bar{X}_i) \equiv A_t).$$

Let us now consider Γ_2 . We represent each clause $c_j = \{\ell_{j1}, \ell_{j2}, \ell_{j3}\}$ of φ with a pattern H_j of the form $\exists r.(Z_{j1} \sqcap \exists r.(Z_{j2} \sqcap \exists r.Z_{j3}))$, where each symbol Z_{jh} is defined as X_i if $\ell_{jh} = x_i$, and as $\bar{X}_i$ if $\ell_{jh} = \neg x_i$. Then, $\Gamma_2 := \{G \sqsubseteq^? H_1, \dots, G \sqsubseteq^? H_m\}$ and $G := G_1 \sqcap G_2 \sqcap G_3$, where:

$$\begin{aligned} G_1 &:= \exists r.(A_t \sqcap \exists r.(A_t \sqcap A_f \sqcap \exists r.(A_t \sqcap A_f))), & G_2 &:= \exists r.(A_t \sqcap A_f \sqcap \exists r.(A_t \sqcap \exists r.(A_t \sqcap A_f))), \\ G_3 &:= \exists r.(A_t \sqcap A_f \sqcap \exists r.(A_t \sqcap A_f \sqcap \exists r.A_t)). \end{aligned}$$

Intuitively, for each variable Z_{jh} in H_j , the concept G_h encodes all possible ways to assign A_t to Z_{jh} , and either A_t or A_f to the other variables in H_j . For instance, let σ be a matcher of Γ_1 . Then, to have $G_1 \sqsubseteq \sigma(H_j)$, the structure of G_1 demands that $\sigma(Z_{j1}) \equiv A_t$, while giving σ the freedom to assign A_t or A_f to the other two variables in H_j . Therefore, we have that $G \sqsubseteq \sigma(H_j)$ iff σ maps at least one variable in H_j to A_t . This provides us with a correspondence between satisfying assignments of φ and substitutions that simultaneously solve the constraints in Γ_2 . This concludes the sketch of the reduction.

Finally, Γ_φ can be constructed in time polynomial in the size of φ . Thus,

Theorem 3. *The problem of deciding left-ground matching in $\mathcal{EL}^{-\top}$ is NP-hard.*

We conclude this section by briefly sketching how to show PSpace-hardness of left-ground matching in $\mathcal{EL}^{-\top}$ w.r.t. arbitrary TBoxes. The PSpace lower bound is established by a reduction from the intersection emptiness problem for *deterministic finite automata (DFA)*, which is PSpace-complete [17]. The *intersection emptiness problem* asks, given a finite sequence $\mathcal{A}_1, \dots, \mathcal{A}_n$ of DFAs over an alphabet Σ , whether $\mathcal{L}(\mathcal{A}_1) \cap \dots \cap \mathcal{L}(\mathcal{A}_n) \neq \emptyset$.

Our reduction is an adaptation of the one provided in [12] to show that unification in $\mathcal{EL}^{-\top}$ is PSpace-hard (w.r.t. the empty TBox). It translates a given sequence $\mathcal{A}_1, \dots, \mathcal{A}_n$ of DFAs into an $\mathcal{EL}^{-\top}$ -TBox $\mathcal{T}$ and a left-ground matching problem Γ such that $\mathcal{L}(\mathcal{A}_1) \cap \dots \cap \mathcal{L}(\mathcal{A}_n) \neq \emptyset$ iff Γ has an $\mathcal{EL}^{-\top}$ -matcher w.r.t. $\mathcal{T}$. We first set $\mathbb{N}_R := \Sigma$ and introduce for each automaton $\mathcal{A}_i$ and each state q of $\mathcal{A}_i$ a concept name $B_{q,i}$. Then, we use the transition function of $\mathcal{A}_i$ to define a TBox $\mathcal{T}_i$, whose purpose is to simulate the behavior of $\mathcal{A}_i$ via the relation $\sqsubseteq_{\mathcal{T}_i}$. That is, we can show that the following holds for all $w = r_1 r_2 \dots r_k \in \Sigma^*$ ($k \geq 0$):

$$B_{q_0,i} \sqsubseteq_{\mathcal{T}_i} \exists r_1. \exists r_2. \dots \exists r_k. A \quad \text{iff} \quad w \in \mathcal{L}(\mathcal{A}_i), \quad (1)$$

where q_0 is the initial state of $\mathcal{A}_i$ and A is a distinguished (and the only) concept name shared by $\mathcal{T}_1, \dots, \mathcal{T}_n$. Finally, we define the TBox $\mathcal{T}$ as $\mathcal{T} := \mathcal{T}_1 \cup \dots \cup \mathcal{T}_n$ and the left-ground matching problem Γ as $\Gamma := \bigcup_{i=1}^n \{B_{q_0,i} \sqsubseteq^? Y\}$, where Y is a concept variable. The common use of Y on the right-hand side of all constraints in Γ and the equivalence in (1) ensure that the reduction is correct.

Theorem 4. *Deciding left-ground matching in $\mathcal{EL}^{-\top}$ w.r.t. a TBox is PSpace-hard.*

4 A decision procedure for matching in $\mathcal{EL}^{-\top}$

In [9], it was shown that matching in $\mathcal{EL}$ w.r.t. $\mathcal{EL}$ -TBoxes is in NP by introducing a goal-oriented decision procedure that runs in non-deterministic polynomial time. Let us denote this

algorithm by **Alg**. On input $(\Gamma_0, \mathcal{T})$, **Alg** starts by initializing $\Gamma := \Gamma_0$ and then continues by applying deterministic and non-deterministic rules to the constraints in Γ . A non-failing application of a rule may add new subsumption constraints to Γ . A successful run of **Alg** is an exhaustive sequence of non-failing rule applications. Such a sequence generates a final matching problem Γ that uses only symbols from $(\Gamma_0, \mathcal{T})$ and contains the initial problem Γ_0 .

We cannot directly use this algorithm for matching in $\mathcal{EL}^{-\top}$. For instance, on input $(\Gamma_0 = \{A \sqsubseteq^? X, B \sqsubseteq^? X\}, \emptyset)$, **Alg** returns “success” since $\sigma := \{X \mapsto \top\}$ is an $\mathcal{EL}$ -matcher of Γ_0 . However, Γ_0 has no $\mathcal{EL}^{-\top}$ -matchers. To repair this, we extend **Alg** into a decision procedure for matching in $\mathcal{EL}^{-\top}$. We came up with the idea for extending **Alg**, by analyzing the soundness proof of **Alg** in [9]. Let us briefly explain this. Consider a matching problem Γ obtained after termination of a non-failing run of **Alg**. From Lemma 8 of [9], we can directly infer that Γ satisfies the following property:

$$\text{If } C \sqsubseteq^? X \in \Gamma \text{ and } X \sqsubseteq^? D \in \Gamma \text{ then } C \sqsubseteq_{\mathcal{T}} D. \quad (2)$$

Based on this property, soundness of **Alg** is established in [9], by demonstrating that a specific $\mathcal{EL}$ -substitution σ_{Γ} is a matcher of Γ w.r.t. $\mathcal{T}$. This substitution is defined as:

$$\sigma_{\Gamma}(X) := \sqcap \{D \mid X \sqsubseteq^? D \in \Gamma\} \text{ for all variables } X \text{ occurring in } \Gamma,$$

where the empty conjunction corresponds to $\top$. The full proof showing that σ_{Γ} is a matcher of Γ w.r.t. $\mathcal{T}$ can be found in Lemma 5.4 of [10] (the long version of [9]). This proof consists of two main steps:

- First, it is shown that σ_{Γ} is a matcher of $\Gamma_{\mathbf{V}} := \{C \sqsubseteq^? X \in \Gamma \mid X \in \mathbf{N}_{\mathbf{V}}\} \cup \{X \sqsubseteq^? D \in \Gamma \mid X \in \mathbf{N}_{\mathbf{V}}\}$ w.r.t. $\mathcal{T}$. This is an easy consequence of (2) and the definition of σ_{Γ} .
- Second, taking the subsumption constraints in $\Gamma_{\mathbf{V}}$ as the base case, an appropriate inductive argument is used to show that σ_{Γ} also solves the remaining constraints in $\Gamma \setminus \Gamma_{\mathbf{V}}$.

Interestingly, the proof of the second step does not depend on the particular structure of σ_{Γ} , but only requires that σ_{Γ} solves the constraints in $\Gamma_{\mathbf{V}}$. Hence, the proof of Lemma 5.4 in [10] also shows a slightly more general result. That is,

Remark 5. *If a substitution σ is a matcher of $\Gamma_{\mathbf{V}}$ w.r.t. $\mathcal{T}$, then σ is a matcher of Γ w.r.t. $\mathcal{T}$.*

As observed above, if $\{D \mid X \sqsubseteq^? D \in \Gamma\} = \emptyset$, then $\sigma_{\Gamma}(X) = \top$. In addition, since Γ uses only symbols from $(\Gamma_0, \mathcal{T})$, we have that if $(\Gamma_0, \mathcal{T})$ does not contain $\top$ and $\{D \mid X \sqsubseteq^? D \in \Gamma\} \neq \emptyset$, then $\sigma_{\Gamma}(X)$ is an $\mathcal{EL}^{-\top}$ -concept. These observations, together with Remark 5, suggest how to extend **Alg** into a decision procedure for matching in $\mathcal{EL}^{-\top}$. Basically, we only need to consider the variables X such that $\{D \mid X \sqsubseteq^? D \in \Gamma\} = \emptyset$, and find out appropriate $\mathcal{EL}^{-\top}$ -concepts for them. Note that although $\{D \mid X \sqsubseteq^? D \in \Gamma\} = \emptyset$, Γ can still contain constraints $C_1 \sqsubseteq^? X, \dots, C_k \sqsubseteq^? X$ for some $k > 0$. Hence, such an appropriate $\mathcal{EL}^{-\top}$ -concept for $\sigma_{\Gamma}(X)$ must be a common subsumer of $C_1, \dots, C_k$ not containing $\top$. The new algorithm extending **Alg**, denoted by Alg^* , is as follows:

1. Apply the rules used by **Alg** exhaustively, and fail whenever the application of a rule fails;
2. If no more rule applies and no failure has occurred, then check for each variable X with $\{D \mid X \sqsubseteq^? D \in \Gamma\} = \emptyset$ whether the concepts $C_1, \dots, C_k$ have a common subsumer without $\top$, where $\{C_1 \sqsubseteq^? X, \dots, C_k \sqsubseteq^? X\} = \{C \sqsubseteq^? X \in \Gamma \mid X \in \mathbf{N}_{\mathbf{V}}\}$. If such common subsumers exist for all the considered variables, then return “success”, otherwise return “failure”.

One can show that Alg^* is a decision procedure for matching in $\mathcal{EL}^{-\top}$, provided that the existence of the common subsumers can be effectively decided. Soundness can be shown based on the previous discussions. As for completeness, it can be derived from the following lemma, which shows that Alg is complete for matching in $\mathcal{EL}$.

Lemma 6. (Lemma 11 of [9]) *Let σ be a matcher of Γ_0 w.r.t. $\mathcal{T}$. Then there is a non-failing and terminating run of Alg generating a matching problem Γ such that σ is a matcher of Γ w.r.t. $\mathcal{T}$.*

In fact, since $\mathcal{EL}^{-\top}$ -substitutions are also $\mathcal{EL}$ -substitutions, Lemma 6 implies that Alg^* has a non-failing and terminating run if Γ_0 has an $\mathcal{EL}^{-\top}$ -matcher w.r.t. $\mathcal{T}$.

Thus, it remains to see how to decide the existence of a common subsumer without $\top$ for k $\mathcal{EL}$ -concepts. By using model theoretic characterizations of $\mathcal{EL}$, we can show that deciding the existence of such a common subsumer can be reduced to solving a reachability problem in a certain graph. This graph can be of size exponential in k , but it can actually be generated *on-the-fly*. Hence, the corresponding reachability problem can be decided in PSpace. When $\mathcal{T} = \emptyset$, reachability in this graph can be decided in NP, since the graph becomes a tree of polynomial depth. Therefore, Alg^* is a non-deterministic polynomial space procedure for matching in $\mathcal{EL}^{-\top}$, which runs in non-deterministic polynomial time if $\mathcal{T} = \emptyset$. Thus, by Savitch's theorem [19] and the lower bounds shown in Theorem 3 and Theorem 4, we have the following results.

Theorem 7. *The problem of deciding matching in $\mathcal{EL}^{-\top}$ is NP-complete w.r.t. the empty TBox and PSpace-complete w.r.t. arbitrary TBoxes. The lower bounds hold even for the class of left-ground matching problems.*

5 In the future

We plan to investigate whether considering cycle-restricted TBoxes can lower the complexity of matching in $\mathcal{EL}^{-\top}$. In fact, the TBox built to show PSpace-hardness in Section 3 does not satisfy the cycle-restrictedness condition defined in [1]. Another topic for future research is to consider other interesting restrictions on the set of admissible matchers. Examples of these restrictions have been considered for unification in [4, 13]. Finally, we would like to develop more practical algorithms and integrate their implementations into the system UEL [7].

References

- [1] Franz Baader, Stefan Borgwardt, and Barbara Morawska. Extending unification in $\mathcal{EL}$ towards general tboxes. In Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*. AAAI Press, 2012.
- [2] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In Leslie Pack Kaelbling and Alessandro Saffioti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 364–369. Professional Book Center, 2005. URL: <http://ijcai.org/Proceedings/05/Papers/0372.pdf>.
- [3] Franz Baader and Oliver Fernández Gil. Unification in the description logic $\mathcal{ELH}_{\mathcal{R}^+}^{-\top}$ without the top concept modulo cycle-restricted ontologies. In Christoph Benzmüller, Marijn J. H. Heule, and Renate A. Schmidt, editors, *Automated Reasoning - 12th International Joint Conference, IJCAR 2024, Nancy, France, July 3-6, 2024, Proceedings, Part II*, Lecture Notes in Computer Science, pages 279–297. Springer, 2024. doi:10.1007/978-3-031-63501-4_15.

- [4] Franz Baader, Oliver Fernández Gil, and Maryam Rostamigiv. Restricted unification in the DL $\mathcal{FL}_0$. In Boris Konev and Giles Rege, editors, *Frontiers of Combining Systems - 13th International Symposium, FroCoS 2021, Birmingham, UK, September 8-10, 2021, Proceedings*, Lecture Notes in Computer Science, pages 81–97. Springer, 2021. doi:[10.1007/978-3-030-86205-3_5](https://doi.org/10.1007/978-3-030-86205-3_5).
- [5] Franz Baader and Ralf Küsters. Matching concept descriptions with existential restrictions. In Anthony G. Cohn, Fausto Giunchiglia, and Bart Selman, editors, *KR 2000, Principles of Knowledge Representation and Reasoning Proceedings of the Seventh International Conference, Breckenridge, Colorado, USA, April 11-15, 2000*, pages 261–272. Morgan Kaufmann, 2000.
- [6] Franz Baader, Ralf Küsters, Alexander Borgida, and Deborah L. McGuinness. Matching in description logics. *J. Log. Comput.*, 9(3):411–447, 1999. doi:[10.1093/LOGCOM/9.3.411](https://doi.org/10.1093/LOGCOM/9.3.411).
- [7] Franz Baader, Julian Mendez, and Barbara Morawska. UEL: unification solver for the description logic $\mathcal{EL}$ - system description. In Bernhard Gramlich, Dale Miller, and Uli Sattler, editors, *Automated Reasoning - 6th International Joint Conference, IJCAR 2012, Manchester, UK, June 26-29, 2012. Proceedings*, Lecture Notes in Computer Science, pages 45–51. Springer, 2012. doi:[10.1007/978-3-642-31365-3_6](https://doi.org/10.1007/978-3-642-31365-3_6).
- [8] Franz Baader and Barbara Morawska. Unification in the description logic $\mathcal{EL}$. In Ralf Treinen, editor, *Rewriting Techniques and Applications, 20th International Conference, RTA 2009, Brasília, Brazil, June 29 - July 1, 2009, Proceedings*, Lecture Notes in Computer Science, pages 350–364. Springer, 2009. doi:[10.1007/978-3-642-02348-4_25](https://doi.org/10.1007/978-3-642-02348-4_25).
- [9] Franz Baader and Barbara Morawska. Matching with respect to general concept inclusions in the description logic $\mathcal{EL}$. In Carsten Lutz and Michael Thielscher, editors, *KI 2014: Advances in Artificial Intelligence - 37th Annual German Conference on AI, Stuttgart, Germany, September 22-26, 2014. Proceedings*, Lecture Notes in Computer Science, pages 135–146. Springer, 2014. doi:[10.1007/978-3-319-11206-0_14](https://doi.org/10.1007/978-3-319-11206-0_14).
- [10] Franz Baader and Barbara Morawska. Matching with respect to general concept inclusions in the description logic $\mathcal{EL}$. LTCS-Report 14-03, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany, 2014. See <http://lat.inf.tu-dresden.de/research/reports.html>. doi:[10.25368/2022.205](https://doi.org/10.25368/2022.205).
- [11] Franz Baader and Paliath Narendran. Unification of concept terms in description logics. In Henri Prade, editor, *13th European Conference on Artificial Intelligence, Brighton, UK, August 23-28 1998, Proceedings*, pages 331–335. John Wiley and Sons, 1998.
- [12] Franz Baader, Thanh Binh Nguyen, Stefan Borgwardt, and Barbara Morawska. Deciding unifiability and computing local unifiers in the description logic $\mathcal{EL}$ without top constructor. *Notre Dame J. Formal Log.*, 57(4):443–476, 2016. doi:[10.1215/00294527-3555507](https://doi.org/10.1215/00294527-3555507).
- [13] Franz Baader and Maryam Rostamigiv. Restricted unification in the DL $\mathcal{EL}$. In Martin Homola, Vladislav Ryzhikov, and Renate A. Schmidt, editors, *Proceedings of the 34th International Workshop on Description Logics (DL 2021) part of Bratislava Knowledge September (BAKS 2021), Bratislava, Slovakia, September 19th to 22nd, 2021*, CEUR Workshop Proceedings. CEUR-WS.org, 2021. URL: <https://ceur-ws.org/Vol-2954/paper-4.pdf>.
- [14] Alexander Borgida and Deborah L. McGuinness. Asking queries about frames. In Luigia Carlucci Aiello, Jon Doyle, and Stuart C. Shapiro, editors, *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96), Cambridge, Massachusetts, USA, November 5-8, 1996*, pages 340–349. Morgan Kaufmann, 1996.
- [15] Stephen A. Cook. The complexity of theorem-proving procedures. In Michael A. Harrison, Ranan B. Banerji, and Jeffrey D. Ullman, editors, *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, May 3-5, 1971, Shaker Heights, Ohio, USA*, pages 151–158. ACM, 1971. doi:[10.1145/800157.805047](https://doi.org/10.1145/800157.805047).
- [16] Puneetha Jangir Lok Ram Jangir. Complexity of matching in the description logic $\mathcal{EL}$ without the top concept. Master's thesis, Technische Universität Dresden, Dresden, Germany, 2025. Available at <https://lat.inf.tu-dresden.de/research/theses/2025/Jangir-Mas-25.pdf>.

- [17] Dexter Kozen. Lower bounds for natural proof systems. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 254–266. IEEE Computer Society, 1977. doi:[10.1109/SFCS.1977.16](https://doi.org/10.1109/SFCS.1977.16).
- [18] Ralf Küsters. *Non-Standard Inferences in Description Logics*, volume 2100 of *Lecture Notes in Computer Science*. Springer, 2001. doi:[10.1007/3-540-44613-3](https://doi.org/10.1007/3-540-44613-3).
- [19] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. Syst. Sci.*, 4(2):177–192, 1970. doi:[10.1016/S0022-0000\(70\)80006-X](https://doi.org/10.1016/S0022-0000(70)80006-X).