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

Franz Baader[1⋆], Nguyen Thanh Binh[2], Stefan Borgwardt[1⋆], and Barbara Morawska[1⋆]

[1] TU Dresden, Germany, {baader,stefborg,morawska}@tcs.inf.tu-dresden.de
[2] ETH Zürich, Switzerland, thannguy@inf.ethz.ch

Abstract. Unification in Description Logics has been proposed as a novel inference service that can, for example, be used to detect redundancies in ontologies. The inexpressive Description Logic $\mathcal{EL}$ is of particular interest in this context since, on the one hand, several large biomedical ontologies are defined using $\mathcal{EL}$. On the other hand, unification in $\mathcal{EL}$ has recently been shown to be NP-complete, and thus of considerably lower complexity than unification in other DLs of similarly restricted expressive power. However, $\mathcal{EL}$ allows the use of the top concept ($\top$), which represents the whole interpretation domain, whereas the large medical ontology SNOMED CT makes no use of this feature. Surprisingly, removing the top concept from $\mathcal{EL}$ makes the unification problem considerably harder. More precisely, we will show in this paper that unification in $\mathcal{EL}$ without the top concept is PSpace-complete.

1 Introduction

Description logics (DLs) [4] are a well-investigated family of logic-based knowledge representation formalisms. They can be used to represent the relevant concepts of an application domain using concept terms, which are built from concept names and role names using certain concept constructors. The DL $\mathcal{EL}$ offers the constructors conjunction ($\sqcap$), existential restriction ($\exists r.C$), and the top concept ($\top$). From a semantic point of view, concept names and concept terms represent sets of individuals, whereas roles represent binary relations between individuals. The top concept is interpreted as the set of all individuals. For example, using the concept names Male, Female, Person and the role names child, job, the concept of *persons having a son, a daughter, and a job* can be represented by the $\mathcal{EL}$-concept term $\mathsf{Person} \sqcap \exists \mathsf{child}.\mathsf{Male} \sqcap \exists \mathsf{child}.\mathsf{Female} \sqcap \exists \mathsf{job}.\top$.

In this example, the availability of the top concept in $\mathcal{EL}$ allows us to state that the person has some job, without specifying any further to which concept this job belongs. Knowledge representation systems based on DLs provide their users with various inference services that allow them to deduce implicit knowledge from the explicitly represented knowledge. For instance, the subsumption algorithm allows one to determine subconcept-superconcept relationships. For

⋆ Supported by DFG under grant BA 1122/14-1

example, the concept term $\exists \mathsf{job}.\top$ subsumes (i.e., is a superconcept of) the concept term $\exists \mathsf{job}.\mathsf{Boring}$ since anyone that has a boring job at least has some job. Two concept terms are called *equivalent* if they subsume each other, i.e., if they are always interpreted as the same set of individuals.

The DL $\mathcal{EL}$ has recently drawn considerable attention since, on the one hand, important inference problems such as the subsumption problem are polynomial in $\mathcal{EL}$ [1, 3]. On the other hand, though quite inexpressive, $\mathcal{EL}$ can be used to define biomedical ontologies. For example, the large medical ontology SNOMED CT[1] can be expressed in $\mathcal{EL}$. Actually, if one takes a closer look at the concept definitions in SNOMED CT, then one sees that they do not contain the top concept.

Unification in DLs has been proposed in [8] as a novel inference service that can, for example, be used to detect redundancies in ontologies. For example, assume that one knowledge engineer defines the concept of *female professors* as

$$\mathsf{Person} \sqcap \mathsf{Female} \sqcap \exists \mathsf{job}.\mathsf{Professor},$$

whereas another knowledge engineer represent this notion in a somewhat different way, e.g., by using the concept term

$$\mathsf{Woman} \sqcap \exists \mathsf{job}.(\mathsf{Teacher} \sqcap \mathsf{Researcher}).$$

These two concept terms are not equivalent, but they are nevertheless meant to represent the same concept. They can obviously be made equivalent by substituting the concept name $\mathsf{Professor}$ in the first term by the concept term $\mathsf{Teacher} \sqcap \mathsf{Researcher}$ and the concept name Woman in the second term by the concept term $\mathsf{Person} \sqcap \mathsf{Female}$. We call a substitution that makes two concept terms equivalent a *unifier* of the two terms. Such a unifier proposes definitions for the concept names that are used as variables. In our example, we know that, if we define Woman as $\mathsf{Person} \sqcap \mathsf{Female}$ and $\mathsf{Professor}$ as $\mathsf{Teacher} \sqcap \mathsf{Researcher}$, then the two concept terms from above are equivalent w.r.t. these definitions.

In [8] it was shown that, for the DL $\mathcal{FL}_0$, which differs from $\mathcal{EL}$ by offering value restrictions ($\forall r.C$) in place of existential restrictions, deciding unifiability is an ExpTime-complete problem. In [5], we were able to show that unification in $\mathcal{EL}$ is of considerably lower complexity: the decision problem is "only" NP-complete. The original unification algorithm for $\mathcal{EL}$ introduced in [5] was a brutal "guess and then test" NP-algorithm, but we have since then also developed more practical algorithms. On the one hand, in [7] we describe a goal-oriented unification algorithm for $\mathcal{EL}$, in which non-deterministic decisions are only made if they are triggered by "unsolved parts" of the unification problem. On the other hand, in [6], we present an algorithm that is based on a reduction to satisfiability in propositional logic (SAT), and thus allows us to employ highly optimized state-of-the-art SAT solvers for implementing an $\mathcal{EL}$-unification algorithm.

As mentioned above, however, SNOMED CT is not formulated in $\mathcal{EL}$, but rather in its sub-logic $\mathcal{EL}^{-\top}$, which differs from $\mathcal{EL}$ in that the use of the top

[1] see http://www.ihtsdo.org/snomed-ct/

Name	Syntax	Semantics	$\mathcal{EL}$	$\mathcal{EL}^{-\top}$
concept name	A	$A^{\mathcal{I}} \subseteq \mathcal{D}_{\mathcal{I}}$	×	×
role name	r	$r^{\mathcal{I}} \subseteq \mathcal{D}_{\mathcal{I}} \times \mathcal{D}_{\mathcal{I}}$	×	×
top-concept	$\top$	$\top^{\mathcal{I}} = \mathcal{D}_{\mathcal{I}}$	×	
conjunction	$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$	×	×
existential restriction	$\exists r.C$	$(\exists r.C)^{\mathcal{I}} = \{x \mid \exists y : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$	×	×
subsumption	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$	×	×
equivalence	$C \equiv D$	$C^{\mathcal{I}} = D^{\mathcal{I}}$	×	×

Table 1. Syntax and semantics of $\mathcal{EL}$ and $\mathcal{EL}^{-\top}$.

concept is disallowed. If we employ $\mathcal{EL}$-unification to detect redundancies in (extensions of) SNOMED CT, then a unifier may introduce concept terms that contain the top concept, and thus propose definitions for concept names that are of a form that is not used in SNOMED CT. Apart from this practical motivation for investigating unification in $\mathcal{EL}^{-\top}$, we also found it interesting to see how such a small change in the logic influences the unification problem. Surprisingly, it turned out that the complexity of the problem increases considerably (from NP to PSpace). In addition, compared to $\mathcal{EL}$-unification, quite different methods had to be developed to actually solve $\mathcal{EL}^{-\top}$-unification problems. In particular, we will show in this paper, that—similar to the case of $\mathcal{FL}_0$-unification—$\mathcal{EL}^{-\top}$-unification can be reduced to solving certain language equations. In contrast to the case of $\mathcal{FL}_0$-unification, these language equations can be solved in PSpace rather than ExpTime, which we show by a reduction to the emptiness problem for alternating automata on finite words. Complete proofs of the results presented in this paper can be found in [2].

2 The Description Logics $\mathcal{EL}$ and $\mathcal{EL}^{-\top}$

Starting with a set N_C of concept names and a set N_R of role names, $\mathcal{EL}$-*concept terms* are built using the concept constructors *top-concept* ($\top$), *conjunction* ($C \sqcap D$), and *existential restriction* ($\exists r.C$ for every $r \in N_R$). The $\mathcal{EL}$-concept term C is an $\mathcal{EL}^{-\top}$-*concept term* if $\top$ does not occur in C. Since $\mathcal{EL}^{-\top}$-concept terms are special $\mathcal{EL}$-concept terms, many definitions and results transfer from $\mathcal{EL}$ to $\mathcal{EL}^{-\top}$, and thus we only formulate them for $\mathcal{EL}$. We will explicitly mention it if this is not the case.

The semantics of $\mathcal{EL}$ and $\mathcal{EL}^{-\top}$ is defined in the usual way, using the notion of an interpretation $\mathcal{I} = (\mathcal{D}_{\mathcal{I}}, \cdot^{\mathcal{I}})$, which consists of a nonempty domain $\mathcal{D}_{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ that assigns binary relations on $\mathcal{D}_{\mathcal{I}}$ to role names and subsets of $\mathcal{D}_{\mathcal{I}}$ to concept terms, as shown in the semantics column of Table 1. The concept term C *is subsumed by* the concept term D (written $C \sqsubseteq D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all interpretations $\mathcal{I}$. We say that C *is equivalent to* D (written $C \equiv D$) iff $C \sqsubseteq D$ and $D \sqsubseteq C$, i.e., iff $C^{\mathcal{I}} = D^{\mathcal{I}}$ holds for all interpretations $\mathcal{I}$.

An $\mathcal{EL}$-concept term is called an *atom* iff it is a concept name $A \in N_C$ or an existential restriction $\exists r.D$. Concept names and existential restrictions $\exists r.D$, where D is a concept name or $\top$, are called *flat atoms*. The set $\mathrm{At}(C)$ of *atoms of an $\mathcal{EL}$-concept term C* consists of all the subterms of C that are atoms. For example, $C = A \sqcap \exists r.(B \sqcap \exists r.\top)$ has the atom set $\mathrm{At}(C) = \{A, \exists r.(B \sqcap \exists r.\top), B, \exists r.\top\}$. Obviously, any $\mathcal{EL}$-concept term C is a conjunction $C = C_1 \sqcap \ldots \sqcap C_n$ of atoms and $\top$. We call the atoms among $C_1, \ldots, C_n$ the *top-level atoms* of C. The $\mathcal{EL}$-concept term C is called *flat* if all its top-level atoms are flat. Subsumption in $\mathcal{EL}$ and $\mathcal{EL}^{-\top}$ can be characterized as follows [7]:

Lemma 1. *Let $C = A_1 \sqcap \ldots \sqcap A_k \sqcap \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_m.C_m$ and $D = B_1 \sqcap \ldots \sqcap B_l \sqcap \exists s_1.D_1 \sqcap \ldots \sqcap \exists s_n.D_n$ be two $\mathcal{EL}$-concept terms, where $A_1, \ldots, A_k, B_1, \ldots, B_l$ are concept names. Then $C \sqsubseteq D$ iff $\{B_1, \ldots, B_l\} \subseteq \{A_1, \ldots, A_k\}$ and for every $j \in \{1, \ldots, n\}$ there exists an $i \in \{1, \ldots, m\}$ such that $r_i = s_j$ and $C_i \sqsubseteq D_j$.*

In particular, this means that $C \sqsubseteq D$ iff for every top-level atom D' of D there is a top-level atom C' of C such that $C' \sqsubseteq D'$.

Modulo equivalence, the subsumption relation is a partial order on concept terms. In $\mathcal{EL}$, the top concept $\top$ is the greatest element w.r.t. this order. In $\mathcal{EL}^{-\top}$, there are many incomparable maximal concept terms. We will see below that these are exactly the $\mathcal{EL}^{-\top}$-concept terms of the form $\exists r_1.\cdots\exists r_n.A$ for $n \geq 0$ role names $r_1, \ldots, r_n$ and a concept name A. We call such concept terms *particles* . The set $\mathrm{Part}(C)$ of all particles of a given $\mathcal{EL}^{-\top}$-concept term C is defined as

- $\mathrm{Part}(C) := \{C\}$ if C is a concept name,
- $\mathrm{Part}(C) := \{\exists r.E \mid E \in \mathrm{Part}(D)\}$ if $C = \exists r.D$,
- $\mathrm{Part}(C) := \mathrm{Part}(C_1) \cup \mathrm{Part}(C_2)$ if $C = C_1 \sqcap C_2$.

For example, the particles of $C = A \sqcap \exists r.(A \sqcap \exists r.B)$ are $A, \exists r.A, \exists r.\exists r.B$. Such particles will play an important role in our $\mathcal{EL}^{-\top}$-unification algorithm. The next lemma states that particles are indeed the maximal concept terms w.r.t. to subsumption in $\mathcal{EL}^{-\top}$, and that the particles subsuming an $\mathcal{EL}^{-\top}$-concept term C are exactly the particles of C.

Lemma 2. *Let C be an $\mathcal{EL}^{-\top}$-concept term and B a particle.*

1. *If $B \sqsubseteq C$, then $B \equiv C$.*
2. *$B \in \mathrm{Part}(C)$ iff $C \sqsubseteq B$.*

3 Unification in $\mathcal{EL}$ and $\mathcal{EL}^{-\top}$

To define unification in $\mathcal{EL}$ and $\mathcal{EL}^{-\top}$ simultaneously, let $\mathcal{L} \in \{\mathcal{EL}, \mathcal{EL}^{-\top}\}$. When defining unification in $\mathcal{L}$, we assume that the set of concepts names is partitioned into a set N_v of concept variables (which may be replaced by substitutions) and a set N_c of concept constants (which must not be replaced by substitutions). An *$\mathcal{L}$-substitution* σ is a mapping from N_v into the set of all

$\mathcal{L}$-concept terms. This mapping is extended to concept terms in the usual way, i.e., by replacing all occurrences of variables in the term by their σ-images. An $\mathcal{L}$-concept term is called *ground* if it contains no variables, and an $\mathcal{L}$-substitution σ is called *ground* if the concept terms $\sigma(X)$ are ground for all $X \in N_v$.

Unification tries to make concept terms equivalent by applying a substitution.

Definition 1. *An* $\mathcal{L}$-unification problem *is of the form* $\Gamma = \{C_1 \equiv^? D_1, \ldots, C_n \equiv^? D_n\}$*, where* $C_1, D_1, \ldots C_n, D_n$ *are* $\mathcal{L}$*-concept terms. The* $\mathcal{L}$*-substitution* σ *is an* $\mathcal{L}$-unifier *of* Γ *iff it solves all the equations* $C_i \equiv^? D_i$ *in* Γ*, i.e., iff* $\sigma(C_i) \equiv \sigma(D_i)$ *for* $i = 1, \ldots, n$*. In this case,* Γ *is called* $\mathcal{L}$-unifiable.

In the following, we will use the subsumption $C \sqsubseteq^? D$ as an abbreviation for the equation $C \sqcap D \equiv^? C$. Obviously, σ solves this equation iff $\sigma(C) \sqsubseteq \sigma(D)$.

Clearly, every $\mathcal{EL}^{-\top}$-unification problem Γ is also an $\mathcal{EL}$-unification problem. Whether Γ is $\mathcal{L}$-unifiable or not may depend, however, on whether $\mathcal{L} = \mathcal{EL}$ or $\mathcal{L} = \mathcal{EL}^{-\top}$. As an example, consider the problem $\Gamma := \{A \sqsubseteq^? X, B \sqsubseteq^? X\}$, where A, B are distinct concept constants and X is a concept variable. Obviously, the substitution that replaces X by $\top$ is an $\mathcal{EL}$-unifier of Γ. However, Γ does not have an $\mathcal{EL}^{-\top}$-unifier. In fact, for such a unifier σ, the $\mathcal{EL}^{-\top}$-concept term $\sigma(X)$ would need to satisfy $A \sqsubseteq \sigma(X)$ and $B \sqsubseteq \sigma(X)$. Since A and B are particles, Lemma 2 would imply $A \equiv \sigma(X) \equiv B$ and thus $A \equiv B$, which is not the case.

It is easy to see that, for both $\mathcal{L} = \mathcal{EL}$ and $\mathcal{L} = \mathcal{EL}^{-\top}$, an $\mathcal{L}$-unification problem Γ has an $\mathcal{L}$-unifier iff it has a ground $\mathcal{L}$-unifier σ that uses only concept and role names occurring in Γ,[2] i.e., for all variables X, the $\mathcal{L}$-concept term $\sigma(X)$ is a ground term that contains only such concept and role names. In addition, we may without loss of generality restrict our attention to *flat* $\mathcal{L}$*-unification problems*, i.e., unification problems in which the left- and right-hand sides of equations are flat $\mathcal{L}$-concept terms (see, e.g., [7]).

Given a flat $\mathcal{L}$-unification problem Γ, we denote by $\mathrm{At}(\Gamma)$ the set of all atoms of Γ, i.e., the union of all sets of atoms of the concept terms occurring in Γ. By $\mathrm{Var}(\Gamma)$ we denote the variables that occur in Γ, and by $\mathrm{NV}(\Gamma) := \mathrm{At}(\Gamma) \setminus \mathrm{Var}(\Gamma)$ the set of all *non-variable atoms* of Γ.

$\mathcal{EL}$-unification by guessing acyclic assignments

The NP-algorithm for $\mathcal{EL}$-unification introduced in [5] guesses, for every variable X occurring in Γ, a set $S(X)$ of non-variable atoms of Γ. Given such an *assignment* of sets of non-variable atoms to the variables in Γ, we say that the variable X *directly depends on* the variable Y if Y occurs in an atom of $S(X)$. Let *depends on* be the transitive closure of *directly depends on*. If there is no variable that depends on itself, then we call this assignment *acyclic*. In case the guessed assignment is not acyclic, this run of the NP-algorithm returns “fail.” Otherwise, there exists a strict linear order $>$ on the variables occurring in Γ such that $X > Y$ if X depends on Y. One can then define the substitution γ^S induced by the assignment S along this linear order:

[2] Without loss of generality, we assume that Γ contains at least one concept name.

– If X is the least variable w.r.t. $>$, then $\gamma^S(X)$ is the conjunction of the elements of $S(X)$, where the empty conjunction is $\top$.
– Assume $\gamma^S(Y)$ is defined for all variables $Y < X$. If $S(X) = \{D_1, \ldots, D_n\}$, then $\gamma^S(X) := \gamma^S(D_1) \sqcap \ldots \sqcap \gamma^S(D_n)$.

The algorithm then tests whether the substitution γ^S computed this way is a unifier of Γ. If this is the case, then this run returns γ^S; otherwise, it returns "fail." In [5] it is shown that Γ is unifiable iff there is a run of this algorithm on input Γ that returns a substitution (which is then an $\mathcal{EL}$-unifier of Γ).

Why this does not work for $\mathcal{EL}^{-\top}$

The $\mathcal{EL}$-unifiers returned by the $\mathcal{EL}$-unification algorithm sketched above need not be $\mathcal{EL}^{-\top}$-unifiers since some of the sets $S(X)$ in the guessed assignment may be empty, in which case $\gamma^S(X) = \top$. This suggests the following simple modification of the above algorithm: require that the guessed assignment is such that all sets $S(X)$ are nonempty. If such an assignment S is acyclic, then the induced substitution γ^S is actually an $\mathcal{EL}^{-\top}$-substitution, and thus the substitutions returned by the modified algorithm are indeed $\mathcal{EL}^{-\top}$-unifiers. However, this modified algorithm does not always detect $\mathcal{EL}^{-\top}$-unifiability, i.e., it may return no substitution although the input problem is $\mathcal{EL}^{-\top}$-unifiable.

As an example, consider the $\mathcal{EL}^{-\top}$-unification problem

$$\Gamma := \{A \sqcap B \equiv^? Y,\ B \sqcap C \equiv^? Z,\ \exists r.Y \sqsubseteq^? X,\ \exists r.Z \sqsubseteq^? X\},$$

where X, Y, Z are concept variables and A, B, C are distinct concept constants. We claim that, up to equivalence, the substitution that maps X to $\exists r.B$, Y to $A \sqcap B$, and Z to $B \sqcap C$ is the only $\mathcal{EL}^{-\top}$-unifier of Γ. In fact, any $\mathcal{EL}^{-\top}$-unifier γ of Γ must map Y to $A \sqcap B$ and Z to $B \sqcap C$, and thus satisfy $\exists r.(A \sqcap B) \sqsubseteq \gamma(X)$ and $\exists r.(B \sqcap C) \sqsubseteq \gamma(X)$. Lemma 1 then yields that the only possible top-level atom of $\gamma(X)$ is $\exists r.B$. However, there is no non-variable atom $D \in \mathrm{NV}(\Gamma)$ such that $\gamma(D)$ is equivalent to $\exists r.B$. This shows that Γ has an $\mathcal{EL}^{-\top}$-unifier, but this unifier cannot be computed by the modified algorithm sketched above.

The main idea underlying the $\mathcal{EL}^{-\top}$-unification algorithm introduced in the next section is that one starts with an $\mathcal{EL}$-unifier, and then conjoins "appropriate" particles to the images of the variables that are replaced by $\top$ by this unifier. It is, however, not so easy to decide which particles can be added this way without turning the $\mathcal{EL}$-unifier into an $\mathcal{EL}^{-\top}$-substitution that no longer solves the unification problem.

4 An $\mathcal{EL}^{-\top}$-unification algorithm

In the following, let Γ be a flat $\mathcal{EL}^{-\top}$-unification problem. Without loss of generality we assume that Γ consists of subsumptions of the form $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? D$ for atoms $C_1, \ldots, C_n, D$. Our decision procedure for $\mathcal{EL}^{-\top}$-unifiability proceeds in four steps.

Step 1. If S is an acyclic assignment guessed by the $\mathcal{EL}$-unification algorithm sketched above, then $D \in S(X)$ implies that the subsumption $\gamma^S(X) \sqsubseteq \gamma^S(D)$ holds for the substitution γ^S induced by S. Instead of guessing just subsumptions between variables and non-variable atoms, our $\mathcal{EL}^{-\top}$-unification algorithm starts with guessing subsumptions between arbitrary atoms of Γ. To be more precise, it guesses a mapping $\tau : \mathrm{At}(\Gamma)^2 \to \{0, 1\}$, which specifies which subsumptions between atoms of Γ should hold for the $\mathcal{EL}^{-\top}$-unifier that it tries to generate: if $\tau(D_1, D_2) = 1$ for $D_1, D_2 \in \mathrm{At}(\Gamma)$, then this means that the search for a unifier is restricted (in this branch of the search tree) to substitutions γ satisfying $\gamma(D_1) \sqsubseteq \gamma(D_2)$. Obviously, any such mapping τ also yields an assignment

$$S^\tau(X) := \{D \in \mathrm{NV}(\Gamma) \mid \tau(X, D) = 1\},$$

and we require that this assignment is acyclic and induces an $\mathcal{EL}$-unifier of Γ.

Definition 2. *The mapping* $\tau : \mathrm{At}(\Gamma)^2 \to \{0, 1\}$ *is called a* subsumption mapping *for* Γ *if it satisfies the following three conditions:*

1. *It respects the properties of subsumption in* $\mathcal{EL}$*:*
   (a) $\tau(D, D) = 1$ *for each* $D \in \mathrm{At}(\Gamma)$.
   (b) $\tau(A_1, A_2) = 0$ *for distinct concept constants* $A_1, A_2 \in \mathrm{At}(\Gamma)$.
   (c) $\tau(\exists r.C_1, \exists s.C_2) = 0$ *for distinct* $r, s \in N_R$ *with* $\exists r.C_1, \exists s.C_2 \in \mathrm{At}(\Gamma)$.
   (d) $\tau(A, \exists r.C) = \tau(\exists r.C, A) = 0$ *for each constant* $A \in \mathrm{At}(\Gamma)$*, role name* r *and variable or constant* C *with* $\exists r.C \in \mathrm{At}(\Gamma)$.
   (e) *If* $\exists r.C_1, \exists r.C_2 \in \mathrm{At}(\Gamma)$*, then* $\tau(\exists r.C_1, \exists r.C_2) = \tau(C_1, C_2)$.
   (f) *For all atoms* $D_1, D_2, D_3 \in \mathrm{At}(\Gamma)$*, if* $\tau(D_1, D_2) = \tau(D_2, D_3) = 1$*, then* $\tau(D_1, D_3) = 1$.
2. *It induces an* $\mathcal{EL}$*-substitution, i.e., the assignment* S^τ *is acyclic and thus induces a substitution* γ^{S^τ}*, which we will simply denote by* γ^τ.
3. *It respects the subsumptions of* Γ*, i.e., it satisfies the following conditions for each subsumption* $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? D$ *in* Γ*:*
   (a) *If* D *is a non-variable atom, then there is at least one* C_i *such that* $\tau(C_i, D) = 1$.
   (b) *If* D *is a variable and* $\tau(D, C) = 1$ *for a non-variable atom* $C \in \mathrm{NV}(\Gamma)$*, then there is at least one* C_i *with* $\tau(C_i, C) = 1$.

Though this is not really necessary for the proof of correctness of our $\mathcal{EL}^{-\top}$-unification algorithm, it can be shown that the substitution γ^τ induced by a subsumption mapping τ for Γ is indeed an $\mathcal{EL}$-unifier of Γ. It should be noted that γ^τ need not be an $\mathcal{EL}^{-\top}$-unifier of Γ. In addition, γ^τ need not agree with τ on every subsumption between atoms of Γ. The reason for this is that τ specifies subsumptions which should hold in the $\mathcal{EL}^{-\top}$-unifier of Γ to be constructed. To turn γ^τ into such an $\mathcal{EL}^{-\top}$-unifier, we may have to add certain particles, and these additions may invalidate subsumptions that hold for γ^τ. However, we will ensure that no subsumption claimed by τ is invalidated.

Step 2. In this step, we use τ to turn Γ into a unification problem that has only variables on the right-hand sides of subsumptions. More precisely, we define $\Delta_{\Gamma,\tau} := \Delta_\Gamma \cup \Delta_\tau$, where

$$\Delta_\Gamma := \{C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? X \in \Gamma \mid X \text{ is a variable of } \Gamma\},$$
$$\Delta_\tau := \{C \sqsubseteq^? X \mid X \text{ is a variable and } C \text{ an atom of } \Gamma \text{ with } \tau(C,X) = 1\}.$$

For an arbitrary $\mathcal{EL}^{-\top}$-substitution σ, we define

$$S^\sigma(X) := \{D \in \mathrm{NV}(\Gamma) \mid \sigma(X) \sqsubseteq \sigma(D)\},$$

and write $S^\tau \leq S^\sigma$ if $S^\tau(X) \subseteq S^\sigma(X)$ for every variable X. The following lemma states the connection between $\mathcal{EL}^{-\top}$-unifiability of Γ and of $\Delta_{\Gamma,\tau}$, using the notation that we have just introduced.

Lemma 3. *Let Γ be a flat $\mathcal{EL}^{-\top}$-unification problem. Then the following statements are equivalent for any $\mathcal{EL}^{-\top}$-substitution σ:*

1. *σ is an $\mathcal{EL}^{-\top}$-unifier of Γ.*
2. *There is a subsumption mapping $\tau : \mathrm{At}(\Gamma)^2 \to \{0,1\}$ for Γ such that σ is an $\mathcal{EL}^{-\top}$-unifier of $\Delta_{\Gamma,\tau}$ and $S^\tau \leq S^\sigma$.*

Step 3. In this step, we characterize which particles can be added in order to turn γ^τ into an $\mathcal{EL}^{-\top}$-unifier σ of $\Delta_{\Gamma,\tau}$ satisfying $S^\tau \leq S^\sigma$. Recall that particles are of the form $\exists r_1.\cdots\exists r_n.A$ for $n \geq 0$ role names $r_1,\ldots,r_n$ and a concept name A. We write such a particle as $\exists w.A$, where $w = r_1 \cdots r_n$ is viewed as a word over the alphabet N_R of all role names. If $n = 0$, then w is the empty word ε and $\exists\varepsilon.A$ is just A.

Admissible particles are determined by solutions of a system of linear language inclusions. These *linear inclusions* are of the form

$$X_i \subseteq L_0 \cup L_1 X_1 \cup \ldots \cup L_n X_n, \tag{1}$$

where $X_1,\ldots,X_n$ are indeterminates, $i \in \{1,\ldots,n\}$, and each L_i ($i \in \{0,\ldots,n\}$) is a subset of $N_R \cup \{\varepsilon\}$. A *solution* θ of such an inclusion assigns sets of words $\theta(X_i) \subseteq N_R^*$ to the indeterminates X_i such that $\theta(X_i) \subseteq L_0 \cup L_1\theta(X_1) \cup \ldots \cup L_n\theta(X_n)$.

The unification problem $\Delta_{\Gamma,\tau}$ induces a finite system $\mathcal{I}_{\Gamma,\tau}$ of such inclusions. The indeterminates of $\mathcal{I}_{\Gamma,\tau}$ are of the form X_A, where $X \in N_v$ and $A \in N_c$. For each constant $A \in N_c$ and each subsumption of the form $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? X \in \Delta_{\Gamma,\tau}$, we add the following inclusion to $\mathcal{I}_{\Gamma,\tau}$:

$$X_A \subseteq f_A(C_1) \cup \ldots \cup f_A(C_n), \text{ where}$$
$$f_A(C) := \begin{cases} \{r\} f_A(C') & \text{if } C = \exists r.C' \\ Y_A & \text{if } C = Y \text{ is a variable} \\ \{\varepsilon\} & \text{if } C = A \\ \emptyset & \text{if } C \in N_c \setminus \{A\} \end{cases}$$

Since $\Delta_{\Gamma,\tau}$ contains only flat atoms, these inclusion are indeed of the form (1).

We call a solution θ of $\mathcal{I}_{\Gamma,\tau}$ *admissible* if, for every variable $X \in N_v$, there is a constant $A \in N_c$ such that $\theta(X_A)$ is nonempty. This condition will ensure that we can add enough particles to turn γ^τ into an $\mathcal{EL}^{-\top}$-substitution. In order to obtain a substitution at all, only finitely many particles can be added. Thus, we are interested in *finite* solutions of $\mathcal{I}_{\Gamma,\tau}$, i.e., solutions θ such that all the sets $\theta(X_A)$ are finite.

Lemma 4. *Let Γ be a flat $\mathcal{EL}^{-\top}$-unification problem and τ a subsumption mapping for Γ. Then $\Delta_{\Gamma,\tau}$ has an $\mathcal{EL}^{-\top}$-unifier σ with $S^\tau \leq S^\sigma$ iff $\mathcal{I}_{\Gamma,\tau}$ has a finite, admissible solution.*

Proof sketch. Given a ground $\mathcal{EL}^{-\top}$-unifier σ of $\Delta_{\Gamma,\tau}$ with $S^\tau \leq S^\sigma$, we define for each concept variable X and concept constant A occurring in Γ:

$$\theta(X_A) := \{w \in N_R^* \mid \exists w.A \in \mathrm{Part}(\sigma(X))\}.$$

It can then be shown that θ is a solution of $\mathcal{I}_{\Gamma,\tau}$. This solution is finite since any concept term has only finitely many particles, and it is admissible since σ is an $\mathcal{EL}^{-\top}$-substitution.

Conversely, let θ be a finite, admissible solution of $\mathcal{I}_{\Gamma,\tau}$. We define the substitution σ by induction on the dependency order $>$ induced by S^τ as follows. Let X be a variable of Γ and assume that $\sigma(Y)$ has already been defined for all variables Y with $X > Y$. Then we set

$$\sigma(X) := \bigsqcap_{D \in S^\tau(X)} \sigma(D) \sqcap \bigsqcap_{A \in N_c} \bigsqcap_{w \in \theta(X_A)} \exists w.A.$$

Since θ is finite and admissible, σ is a well-defined $\mathcal{EL}^{-\top}$-substitution. It can be shown that $\sigma(X)$ is indeed an $\mathcal{EL}^{-\top}$-unifier of $\Delta_{\Gamma,\tau}$ with $S^\tau \leq S^\sigma$. □

Step 4. In this step we show how to test whether the system $\mathcal{I}_{\Gamma,\tau}$ of linear language inclusions constructed in the previous step has a finite, admissible solution or not. The main idea is to consider the greatest solution of $\mathcal{I}_{\Gamma,\tau}$.

To be more precise, given a system of linear language inclusions $\mathcal{I}$, we can order the solutions of $\mathcal{I}$ by defining $\theta_1 \subseteq \theta_2$ iff $\theta_1(X) \subseteq \theta_2(X)$ for all indeterminates X of $\mathcal{I}$. Since $\theta_\emptyset$, which assigns the empty set to each indeterminate of $\mathcal{I}$, is a solution of $\mathcal{I}$ and solutions are closed under argument-wise union, the following clearly defines the (unique) greatest solution θ^* of $\mathcal{I}$ w.r.t. this order:

$$\theta^*(X) := \bigcup_{\theta \text{ solution of } \mathcal{I}} \theta(X).$$

Lemma 5. *Let X be an indeterminate in $\mathcal{I}$ and θ^* the maximal solution of $\mathcal{I}$. If $\theta^*(X)$ is nonempty, then there is a finite solution θ of $\mathcal{I}$ such that $\theta(X)$ is nonempty.*

Proof. Let $w \in \theta^*(X)$. We construct the finite solution θ of $\mathcal{I}$ by keeping only the words of length $|w|$: for all indeterminates Y occurring in $\mathcal{I}$ we define

$$\theta(Y) := \{u \in \theta^*(Y) \mid |u| \leq |w|\}.$$

By definition, we have $w \in \theta(X)$. To show that θ is indeed a solution of $\mathcal{I}$, consider an arbitrary inclusion $Y \subseteq L_0 \cup L_1X_1 \cup \ldots \cup L_nX_n$ in $\mathcal{I}$, and assume that $u \in \theta(Y)$. We must show that $u \in L_0 \cup L_1\theta(X_1) \cup \ldots \cup L_n\theta(X_n)$. Since $u \in \theta^*(Y)$ and θ^* is a solution of $\mathcal{I}$, we have (i) $u \in L_0$ or (ii) $u \in L_i\theta^*(X_i)$ for some $i, 1 \leq i \leq n$. In the first case, we are done. In the second case, $u = \alpha u'$ for some $\alpha \in L_i \subseteq N_R \cup \{\varepsilon\}$ and $u' \in \theta^*(X_i)$. Since $|u'| \leq |u| \leq |w|$, we have $u' \in \theta(X_i)$, and thus $u \in L_i\theta(X_i)$. □

Lemma 6. *There is a finite, admissible solution of $\mathcal{I}_{\Gamma,\tau}$ iff the maximal solution θ^* of $\mathcal{I}_{\Gamma,\tau}$ is admissible.*

Proof. If $\mathcal{I}_{\Gamma,\tau}$ has a finite, admissible solution θ, then the maximal solution of $\mathcal{I}_{\Gamma,\tau}$ contains this solution, and is thus also admissible.

Conversely, if θ^* is admissible, then (by Lemma 5) for each $X \in \mathrm{Var}(\Gamma)$ there is a constant $A(X)$ and a finite solution θ_X of $\mathcal{I}_{\Gamma,\tau}$ such that $\theta_X(X_{A(X)}) \neq \emptyset$. The union of these solutions θ_X for $X \in \mathrm{Var}(\Gamma)$ is the desired finite, admissible solution. □

Given this lemma, it remains to show how we can test admissibility of the maximal solution θ^* of $\mathcal{I}_{\Gamma,\tau}$. For this purpose, it is obviously sufficient to be able to test, for each indeterminate X_A in $\mathcal{I}_{\Gamma,\tau}$, whether $\theta^*(X_A)$ is empty or not. This can be achieved by representing the languages $\theta^*(X_A)$ using *alternating finite automata with ε-transitions (ε-AFA)*, which are a special case of two-way alternating finite automata. In fact, as shown in [11], the emptiness problem for two-way alternating finite automata (and thus also for ε-AFA) is in PSpace.

Lemma 7. *For each indeterminate X_A in $\mathcal{I}_{\Gamma,\tau}$, we can construct in polynomial time in the size of $\mathcal{I}_{\Gamma,\tau}$ an ε-AFA $\mathcal{A}(X, A)$ such that the language $L(\mathcal{A}(X, A))$ accepted by $\mathcal{A}(X, A)$ is equal to $\theta^*(X_A)$, where θ^* denotes the maximal solution of $\mathcal{I}_{\Gamma,\tau}$.*

This finishes the description of our $\mathcal{EL}^{-\top}$-unification algorithm. It remains to argue why it is a PSpace decision procedure for $\mathcal{EL}^{-\top}$-unifiability.

Theorem 1. *The problem of deciding unifiability in $\mathcal{EL}^{-\top}$ is in* PSpace.

Proof. We show that the problem is in NPSpace, which is equal to PSpace by Savitch's theorem [14].

Let Γ be a flat $\mathcal{EL}^{-\top}$-unification problem. By Lemma 3, Lemma 4, and Lemma 6, we know that Γ is $\mathcal{EL}^{-\top}$-unifiable iff there is a subsumption mapping τ for Γ such that the maximal solution θ^* of $\mathcal{I}_{\Gamma,\tau}$ is admissible.

Thus, we first guess a mapping $\tau : \mathrm{At}(\Gamma)^2 \to \{0, 1\}$ and test whether τ is a subsumption mapping for Γ. Guessing τ can clearly be done in NPSpace. For a given mapping τ, the test whether it is a subsumption mapping for Γ can be done in polynomial time.

From τ we can first construct $\Delta_{\Gamma,\tau}$ and then $\mathcal{I}_{\Gamma,\tau}$ in polynomial time. Given $\mathcal{I}_{\Gamma,\tau}$, we then construct the (polynomially many) ε-AFA $\mathcal{A}(X, A)$, and test them for emptiness. Since the emptiness problem for ε-AFA is in PSPACE, this can be achieved within PSPACE. Given the results of these emptiness tests, we can then check in polynomial time whether, for each concept variable X of Γ there is a concept constant A of Γ such that $\theta^*(X_A) = L(\mathcal{A}(X, A)) \neq \emptyset$. If this is the case, then θ^* is admissible, and thus Γ is $\mathcal{EL}^{-\top}$-unifiable. □

5 PSpace-hardness of $\mathcal{EL}^{-\top}$-unification

We show PSPACE-hardness of $\mathcal{EL}^{-\top}$-unification by reducing the PSPACE-hard intersection emptiness problem for deterministic finite automata (DFA) [12, 9] to the problem of deciding whether a given $\mathcal{EL}^{-\top}$-unification problem has an $\mathcal{EL}^{-\top}$-unifier or not.

First, we define a translation from a given DFA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ to a set of subsumptions $\Gamma_\mathcal{A}$. In the following, we only consider automata that accept a nonempty language. For such DFAs we can assume without loss of generality that there is no state $q \in Q$ that cannot be reached from q_0 or from which F cannot be reached. In fact, such states can be removed from $\mathcal{A}$ without changing the accepted language.

For every state $q \in Q$, we introduce a concept variable X_q. We use only one concept constant, A, and define $N_R := \Sigma$. The set $\Gamma_\mathcal{A}$ is defined as follows:

$$\Gamma_\mathcal{A} := \{L_q \sqsubseteq^? X_q \mid q \in Q \setminus F\} \cup \{A \sqcap L_q \sqsubseteq^? X_q \mid q \in F\}, \text{ where}$$
$$L_q := \bigsqcap_{\substack{\alpha \in \Sigma \\ \delta(q,\alpha) \text{ is defined}}} \exists \alpha . X_{\delta(q,\alpha)}.$$

Note that the left-hand sides of the subsumptions in $\Gamma_\mathcal{A}$ are indeed $\mathcal{EL}^{-\top}$-concept terms, i.e., the conjunctions on the left-hand sides are nonempty. In fact, every state $q \in Q$ is either a final state or a final state is reachable by a nonempty path from q. In the first case, A occurs in the conjunction, and in the second, there must be an $\alpha \in \Sigma$ such that $\delta(q, \alpha)$ is defined, in which case $\exists \alpha . X_{\delta(q,\alpha)}$ occurs in the conjunction.

The following lemma, which can easily be proved by induction on $|w|$, connects particles occurring in $\mathcal{EL}^{-\top}$-unifiers of $\Gamma_\mathcal{A}$ to words accepted by states of the DFA $\mathcal{A}$.

Lemma 8. *Let $q \in Q$, $w \in \Sigma^*$, and γ be a ground $\mathcal{EL}^{-\top}$-unifier of $\Gamma_\mathcal{A}$ with $\gamma(X_q) \sqsubseteq \exists w.A$. Then $w \in L(\mathcal{A}_q)$, where $\mathcal{A}_q := (Q, \Sigma, q, \delta, F)$ is obtained from $\mathcal{A}$ by making q the initial state.*

Together with Lemma 2, this lemma implies that, for every ground $\mathcal{EL}^{-\top}$-unifier γ of $\Gamma_\mathcal{A}$, the language $\{w \in \Sigma^* \mid \exists w.A \in \mathrm{Part}(\gamma(X_{q_0}))\}$ is contained in

$L(\mathcal{A})$. Conversely, we will show that for every word w accepted by $\mathcal{A}$ we can construct a unifier γ_w such that $\exists w.A \in \mathrm{Part}(\gamma_w(X_{q_0}))$.

For the construction of γ_w, we first consider every $q \in Q$ and try to find a word u_q of minimal length that is accepted by $\mathcal{A}_q$. Such a word always exists since we have assumed that we can reach F from every state. Taking arbitrary such words is not sufficient, however. They need to be related in the following sense.

Lemma 9. *There exists a mapping from the states $q \in Q$ to words $u_q \in L(\mathcal{A}_q)$ such that that either $q \in F$ and $u_q = \varepsilon$ or there is a symbol $\alpha \in \Sigma$ such that $\delta(q, \alpha)$ is defined and $u_q = \alpha u_{\delta(q,\alpha)}$.*

Proof. We construct the words u_q by induction on the length n of a shortest word accepted by $\mathcal{A}_q$.

If $n = 0$, then q must be a final state. In this case, we set $u_q := \varepsilon$.

Now, let q be a state such that a shortest word w_q accepted by $\mathcal{A}_q$ has length $n > 0$. Then $w_q = \alpha w'$ for $\alpha \in \Sigma$ and $w' \in \Sigma^*$ and the transition $\delta(q, \alpha) = q'$ is defined. The length of a shortest word accepted by $\mathcal{A}_{q'}$ must be smaller than n, since w' is accepted by $\mathcal{A}_{q'}$. By induction, $u_{q'} \in L(\mathcal{A}_{q'})$ has already been defined and we have $\alpha u_{q'} \in L(\mathcal{A}_q)$. Since $\alpha u_{q'}$ cannot be shorter than $w_q = \alpha w'$, it must also be of length n. We now define $u_q := \alpha u_{q'}$. ☐

We can now proceed with the definition of γ_w for a word $w \in L(\mathcal{A})$. The (unique) accepting run of $\mathcal{A}$ on $w = w_1 \dots w_n$ yields a sequence of states $q_0, q_1, \dots, q_n$ with $q_n \in F$ and $\delta(q_i, w_{i+1}) = q_{i+1}$ for every $i \in \{0, \dots, n-1\}$. We define the substitution γ_w as follows:

$$\gamma_w(X_q) := \exists u_q.A \sqcap \bigsqcap_{i \in I_q} \exists w_{i+1} \dots w_n.A,$$

where $I_q := \{i \in \{0, \dots, n-1\} \mid q_i = q\}$. For every $q \in Q$, we include at least the conjunct $\exists u_q.A$ in $\gamma_w(X_q)$, and thus γ_w is in fact an $\mathcal{EL}^{-\top}$-substitution.

Lemma 10. *If $w \in L(\mathcal{A})$, then γ_w is an $\mathcal{EL}^{-\top}$-unifier of $\Gamma_\mathcal{A}$ and $\gamma_w(X_{q_0}) \sqsubseteq \exists w.A$.*

Proof. Let the unique accepting run of $\mathcal{A}$ on $w = w_1 \dots w_n$ be given by the sequence $q_0 q_1 \dots q_n$ of states with $q_n \in F$ and $\delta(q_i, w_{i+1}) = q_{i+1}$ for every $i \in \{0, \dots, n-1\}$, and let γ_w be defined as above.

We must show that γ_w satisfies the subsumption constraints introduced in $\Gamma_\mathcal{A}$ for every state $q \in Q$: $L_q \sqsubseteq^? X_q$ if $q \in Q \setminus F$ and $A \sqcap L_q \sqsubseteq^? X_q$ if $q \in F$, where

$$L_q := \bigsqcap_{\substack{\alpha \in \Sigma \\ \delta(q,\alpha) \text{ is defined}}} \exists \alpha.X_{\delta(q,\alpha)}.$$

To do this, we consider every top-level atom of $\gamma_w(X_q)$ and show that it subsumes the left-hand side of the above subsumption.

– Consider the conjunct $\exists u_q.A$. If $u_q = \varepsilon$, then $q \in F$ and the left-hand side contains the conjunct A. In this case, the subsumption is satisfied. Otherwise, there is a symbol $\alpha \in \Sigma$ such that $q' := \delta(q, \alpha)$ is defined and $u_q = \alpha u_{q'}$. Since $\exists u_{q'}.A$ is a top-level atom of $\gamma_w(X_{q'})$, we have $\gamma(X_{q'}) \sqsubseteq \exists u_{q'}.A$, and thus $\gamma_w(L_q) \sqsubseteq \exists \alpha.\gamma_w(X_{q'}) \sqsubseteq \exists u_q.A$.
– Let $i \in I_q$, i.e., $q_i = q$, and consider the conjunct $\exists w_{i+1} \dots w_n.A$. Since we have $\delta(q_i, w_{i+1}) = q_{i+1}$ and $\exists w_{i+2} \dots w_n.A$ is a conjunct of $\gamma_w(X_{q_{i+1}})$,[3] we obtain $\gamma_w(L_q) \sqsubseteq \exists w_{i+1}.\gamma_w(X_{q_{i+1}}) \sqsubseteq \exists w_{i+1} \exists w_{i+2} \dots w_n.A = \exists w_{i+1} \dots w_n.A$.

This shows that γ_w is a ground $\mathcal{EL}^{-\top}$-unifier of $\Gamma_{\mathcal{A}}$. Furthermore, since $0 \in I_{q_0}$, the particle $\exists w_1 \dots w_n.A = \exists w.A$ is a top-level atom of $\gamma_w(X_{q_0})$, and thus $\gamma_w(X_{q_0}) \sqsubseteq \exists w.A$. □

For the *intersection emptiness problem* one considers finitely many DFAs $\mathcal{A}_1, \dots, \mathcal{A}_k$, and asks whether $L(\mathcal{A}_1) \cap \dots \cap L(\mathcal{A}_k) \neq \emptyset$. Since this problem is trivially solvable in polynomial time in case $L(\mathcal{A}_i) = \emptyset$ for some $i, 1 \leq i \leq k$, we can assume that the languages $L(\mathcal{A}_i)$ are all nonempty. Thus, we can also assume without loss of generality that the automata $\mathcal{A}_i = (Q_i, \Sigma, q_{0,i}, \delta_i, F_i)$ have pairwise disjoint sets of states Q_i and are reduced in the sense introduced above, i.e., there is no state that cannot be reached from the initial state or from which no final state can be reached. The flat $\mathcal{EL}^{-\top}$-unification problem Γ is now defined as follows:

$$\Gamma := \bigcup_{i \in \{1,\dots,k\}} \left(\Gamma_{\mathcal{A}_i} \cup \{X_{q_{0,i}} \sqsubseteq^? Y\} \right) \quad ,$$

where Y is a new variable not contained in $\Gamma_{\mathcal{A}_i}$ for $i = 1, \dots, k$.

Lemma 11. *Γ is unifiable in $\mathcal{EL}^{-\top}$ iff $L(\mathcal{A}_1) \cap \dots \cap L(\mathcal{A}_k) \neq \emptyset$.*

Proof. If Γ is unifiable in $\mathcal{EL}^{-\top}$, then it has a ground $\mathcal{EL}^{-\top}$-unifier γ and there must be a particle $\exists w.A$ with $w \in \Sigma^*$ and $\gamma(Y) \sqsubseteq \exists w.A$. Since $\gamma(X_{q_{0,i}}) \sqsubseteq \gamma(Y) \sqsubseteq \exists w.A$, Lemma 8 yields $w \in L(\mathcal{A}_{i,q_{0,i}}) = L(\mathcal{A}_i)$ for each $i \in \{1, \dots, k\}$. Thus, the intersection of the languages $L(\mathcal{A}_i)$ is nonempty.

Conversely, let $w \in \Sigma^*$ be a word with $w \in L(\mathcal{A}_1) \cap \dots \cap L(\mathcal{A}_k)$. By Lemma 10, we have for each of the unification problems $\Gamma_{\mathcal{A}_i}$ an $\mathcal{EL}^{-\top}$-unifier $\gamma_{w,i}$ such that $\gamma_{w,i}(X_{q_{0,i}}) \sqsubseteq \exists w.A$. Since the automata have disjoint state sets, the unification problems $\Gamma_{\mathcal{A}_i}$ do not share variables. Thus, we can combine the unifiers $\gamma_{w,i}$ into an $\mathcal{EL}^{-\top}$-substitution γ by defining $\gamma(Y) := \exists w.A$ and $\gamma(X_q) := \gamma_{w,i}(X_q)$ for each $i \in \{1, \dots, k\}$ and $q \in Q_i$. Obviously, this is an $\mathcal{EL}^{-\top}$-unifier of Γ since it satisfies the additional subsumptions $X_{q_{0,i}} \sqsubseteq^? Y$. □

Since the intersection emptiness problem for DFAs is PSpace-hard [12, 9], this lemma immediately yields our final theorem:

Theorem 2. *The problem of deciding unifiability in $\mathcal{EL}^{-\top}$ is* PSpace*-hard.*

[3] If $i = n - 1$, then $\exists w_{i+2} \dots w_n.A = A$.

6 Conclusion

Unification in $\mathcal{EL}$ was introduced in [5] as an inference service that can support the detection of redundancies in large biomedical ontologies, which are frequently written in this DL. Motivated by the fact that the large medical ontology SNOMED CT actually does not use the top concept available in $\mathcal{EL}$, we have in this paper investigated unification in $\mathcal{EL}^{-\top}$, which is obtained from $\mathcal{EL}$ by removing the top concept. More precisely, SNOMED CT is a so-called acyclic $\mathcal{EL}^{-\top}$-TBox,[4] rather than a collection of $\mathcal{EL}^{-\top}$-concept terms. However, as shown in [7], acyclic TBoxes can be easily handled by a unification algorithm for concept terms.

Surprisingly, it has turned out that the complexity of unification in $\mathcal{EL}^{-\top}$ (PSpace) is considerably higher than of unification in $\mathcal{EL}$ (NP). From a theoretical point of view, this result is interesting since it provides us with a natural example where reducing the expressiveness of a given DL (in a rather minor way) results in a drastic increase of the complexity of the unifiability problem. Regarding the complexity of unification in more expressive DLs, not much is known. If we add negation to $\mathcal{EL}$, then we obtain the well-known DL $\mathcal{ALC}$, which corresponds to the basic (multi-)modal logic K [15]. Decidability of unification in K is a long-standing open problem. Recently, undecidability of unification in some extensions of K (for example, by the universal modality) was shown in [18]. These undecidability results also imply undecidability of unification in some expressive DLs (e.g., in $\mathcal{SHIQ}$ [10]).

Apart from its theoretical interest, the result of this paper also has practical implications. Whereas practically rather efficient unification algorithm for $\mathcal{EL}$ can readily be obtained by a translation into SAT [6], it is not so clear how to turn the PSpace algorithm for $\mathcal{EL}^{-\top}$-unification introduced in this paper into a practically useful algorithm. One possibility could be to use a SAT modulo theories (SMT) approach [13]. The idea is that the SAT solver is used to generate all possible subsumption mappings for Γ, and that the theory solver tests the system $\mathcal{I}_{\Gamma,\tau}$ induced by τ for the existence of a finite, admissible solution. How well this works will mainly depend on whether we can develop such a theory solver that satisfies well all the requirements imposed by the SMT approach.

Another topic for future research is how to actually compute $\mathcal{EL}^{-\top}$-unifiers for a unifiable $\mathcal{EL}^{-\top}$-unification problem. In principle, our decision procedure is constructive in the sense that, from appropriate successful runs of the ε-AFA $\mathcal{A}(X, A)$, one can construct a finite, admissible solution of $\mathcal{I}_{\Gamma,\tau}$, and from this an $\mathcal{EL}^{-\top}$-unifier of Γ. However, this needs to be made more explicit, and we need to investigate what kind of $\mathcal{EL}^{-\top}$-unifiers can be computed this way.

[4] Note that the right-identity rules in SNOMED CT [16] are actually not expressed using complex role inclusion axioms, but through the SEP-triplet encoding [17]. Thus, complex role inclusion axioms are not relevant here.

References

1. Franz Baader. Terminological cycles in a description logic with existential restrictions. In *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 325–330, 2003. Morgan Kaufmann, Los Altos.
2. Franz Baader, Nguyen Thanh Binh, Stefan Borgwardt, and Barbara Morawska. Unification in the description logic $\mathcal{EL}$ without the top concept. LTCS-Report 11-01, TU Dresden, Dresden, Germany, 2011.
   See http://lat.inf.tu-dresden.de/research/reports.html.
3. Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, pages 364–369, 2005. Morgan Kaufmann, Los Altos.
4. Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
5. Franz Baader and Barbara Morawska. Unification in the description logic $\mathcal{EL}$. In *Proc. of the 20th Int. Conf. on Rewriting Techniques and Applications (RTA 2009)*, Springer LNCS 5595, pages 350–364, 2009.
6. Franz Baader and Barbara Morawska. SAT encoding of unification in $\mathcal{EL}$. In *Proc. of the 17th Int. Conf. on Logic for Programming, Artifical Intelligence, and Reasoning (LPAR-17)*, Springer LNCS 6397, pages 97–111, 2010.
7. Franz Baader and Barbara Morawska. Unification in the description logic $\mathcal{EL}$. *Logical Methods in Computer Science*, 6(3), 2010.
8. Franz Baader and Paliath Narendran. Unification of concept terms in description logics. *J. of Symbolic Computation*, 31(3):277–305, 2001.
9. Michael R. Garey and David S. Johnson. *Computers and Intractability — A guide to NP-completeness*. W. H. Freeman and Company, San Francisco (USA), 1979.
10. Ian Horrocks, Ulrike Sattler, and Stefan Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–264, 2000.
11. Tao Jiang and B. Ravikumar. A note on the space complexity of some decision problems for finite automata. *Information Processing Letters*, 40:25–31, 1991.
12. Dexter Kozen. Lower bounds for natural proof systems. *Annual IEEE Symposium on Foundations of Computer Science*, pages 254–266, 1977.
13. Robert Nieuwenhuis, Albert Oliveras, and Cesare Tinelli. Solving SAT and SAT modulo theories: From an abstract Davis-Putnam-Logemann-Loveland procedure to DPLL(T). *J. ACM*, 53(6):937–977, 2006.
14. Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.
15. Klaus Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. of the 12th Int. Joint Conf. on Artificial Intelligence (IJCAI'91)*, pages 466–471, 1991.
16. Kent A. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. *Journal of the American Medical Informatics Association*, 2000. Fall Symposium Special Issue.
17. Boontawee Suntisrivaraporn, Franz Baader, Stefan Schulz, and Kent Spackman. Replacing SEP-triplets in SNOMED CT using tractable description logic operators. In *Proceedings of the 11th Conference on Artificial Intelligence in Medicine (AIME'07)*, Springer LNCS 4594, pages 287–291, 2007.
18. Frank Wolter and Michael Zakharyaschev. Undecidability of the unification and admissibility problems for modal and description logics. *ACM Trans. Comput. Log.*, 9(4), 2008.