Dresden University of Technology
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

Unification in the Description Logic $\mathcal{EL}$ Without Top Constructor

Franz Baader Nguyen Thanh Binh Stefan Borgwardt
Barbara Morawska

LTCS-Report 11-01

This is an updated version of the original report that includes Appendix A on locality of unifiers.

Postal Address:
Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
01062 Dresden

http://lat.inf.tu-dresden.de

Visiting Address:
Nöthnitzer Str. 46
Dresden

Unification in the Description Logic $\mathcal{EL}$ Without Top Constructor

Franz Baader   Nguyen Thanh Binh   Stefan Borgwardt
Barbara Morawska

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 that unification in $\mathcal{EL}$ without the top concept is PSPACE-complete.

1 Introduction

Description logics (DLs) [3] 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 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, 2]. 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 [7] 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 [7] 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 [4], 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 [4] was a brutal "guess and then test" NP-algorithm, but we have since then also developed more practical algorithms. On the one hand, in [6] we describe a goal-oriented unification algorithm for $\mathcal{EL}$, in which nondeterministic decisions are only made if they are triggered by "unsolved parts" of the unification problem. On the other

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

hand, in [5], 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 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 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.

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

The syntax of the following DLs is based on a set N_C of *concept names* and a set N_R of *role names*. *$\mathcal{EL}$-concept terms* are built from concept names using the constructors conjunction $C \sqcap D$, existential restriction $\exists r.C$ and top concept $\top$. The syntax of the DL $\mathcal{EL}^{-\top}$ is defined as for $\mathcal{EL}$, with the exception that the concept constructor $\top$ is not allowed.

The semantics of these concept terms is defined as usual, using interpretations $\mathcal{I} = (\mathcal{D}_\mathcal{I}, \cdot^\mathcal{I})$, which consist of a nonempty domain $\mathcal{D}_\mathcal{I}$ and an interpretation function $\cdot^\mathcal{I}$ that assigns subsets of $\mathcal{D}_\mathcal{I}$ to every concept name and binary relations over $\mathcal{D}_\mathcal{I}$ to every role name (see 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 $\mathcal{I}$. C is *equivalent* to D ($C \equiv D$) iff $C^\mathcal{I} = D^\mathcal{I}$ for every interpretation $\mathcal{I}$.

It is useful to know the following characterization of subsumption in $\mathcal{EL}$ [6]. As a special case, this result also holds for $\mathcal{EL}^{-\top}$-concept terms.

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 $C_i \sqsubseteq D_j$.*

Name	Syntax	Semantics
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 $\mathcal{EL}$.

A concept term is called an *atom* iff it is a concept name or an existential restriction. The set $\mathrm{At}(C)$ of all atoms of any concept term C is defined as follows:

- If $C = \top$, then $\mathrm{At}(C) = \emptyset$.
- If C is a concept name, then $\mathrm{At}(C) := \{C\}$.
- If $C = \exists r.D$, then $\mathrm{At}(C) := \{C\} \cup \mathrm{At}(D)$.
- If $C = C_1 \sqcap C_2$, then $\mathrm{At}(C) := \mathrm{At}(C_1) \cup \mathrm{At}(C_2)$.

Every concept term C is a conjunction of atoms $C_1, \ldots, C_n$; these are called the *top-level atoms* of C. As a special case, the concept term $\top$ can be viewed as the empty conjunction ($n = 0$). Using these conventions, the following is an easy consequence of Lemma 1.

Lemma 2. *Let C, D be concept terms. Then $C \sqsubseteq D$ iff for every top-level atom D' of D there is a top-level atom C' of C with $C' \sqsubseteq D'$.*

Concept names and existential restrictions $\exists r.D$, where D is a concept name or $\top$, are called *flat atoms*. A concept term is *flat* iff it is a conjunction of flat atoms.

2.1 Particles

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 $\mathcal{EL}^{-\top}$, however, there are many incomparable maximal concept terms. We will see below that these are exactly the concept terms of the form $\exists r_1.\exists r_2.\ldots\exists r_n.A$ for role names $r_1, \ldots, r_n$ and a concept name A. We call such concept terms *particles* and often abbreviate them by $\exists r_1 \ldots r_n.A$. Thus, particles $\exists w.A$ are characterized by a word $w \in N_R^*$ and a concept name $A \in N_C$. In the case that $w = \varepsilon$, we have $\exists w.A = A$.

The set $\mathrm{Part}(C)$ of all particles of an $\mathcal{EL}^{-\top}$-concept term C is defined as follows.

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

For example, the particles of the concept $A \sqcap \exists r.(A \sqcap \exists r.B)$, where $A, B \in N_C$ and $r \in N_R$, are A, $\exists r.A$ and $\exists rr.B$. The next lemma states that particles are indeed the maximal concept terms w.r.t. subsumption in $\mathcal{EL}^{-\top}$ and characterizes the particles in $\mathrm{Part}(C)$ for an $\mathcal{EL}^{-\top}$-concept term C.

Lemma 3. *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 $C \sqsubseteq B$.*

Proof. We show both claims by induction on the length of B, i.e., the number of existential restrictions it contains.

1. If B is a concept name and $B \sqsubseteq C$, then Lemma 1 yields that B is the only possible top-level atom of C, which implies that $B \equiv C$.

   Otherwise, $B = \exists r.B'$ for a particle B'. Then every top-level atom of C must be of the form $\exists r.C'$ with $B' \sqsubseteq C'$. Since the particle B' is shorter than B, induction yields $B' \equiv C'$ for every top-level atom $\exists r.C'$ of C, which implies $B \equiv C$.

2. If B is a concept name, then $B \in \mathrm{Part}(C)$ is equivalent to the fact that B is a top-level atom of C, which in turn is equivalent to $C \sqsubseteq B$ by Lemma 2.

   Otherwise, $B = \exists r.B'$ for a particle B'. By definition, $B \in \mathrm{Part}(C)$ is equivalent to the existence of a top-level atom $\exists r.C'$ of C with $B' \in \mathrm{Part}(C')$. By induction, this is equivalent to the existence of a top-level atom $\exists r.C'$ of C with $C' \sqsubseteq B'$. By Lemma 1, this is again equivalent to $C \sqsubseteq B$. $\square$

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

In the following, let $\mathcal{L}$ denote one of the DLs $\mathcal{EL}$ or $\mathcal{EL}^{-\top}$.

We partition the set of concept names into a set N_c of *concept constants* and a set N_v of *concept variables*. An *$\mathcal{L}$-substitution* is a mapping σ from the variables to $\mathcal{L}$-concept terms. A substitution can be extended from variables to $\mathcal{L}$-concept terms in the usual way. An $\mathcal{L}$-concept term is called *ground* if it contains no variables and a substitution σ is called *ground* if the concept terms in the image of σ are ground.

Definition 4. 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*.

We will use the subsumption $C \sqsubseteq^? D$ as abbrevation for the equation $C \sqcap D \equiv^? C$. Obviously, σ solves this equation iff $\sigma(C) \sqsubseteq \sigma(D)$.

The problem of *$\mathcal{L}$-unification* is to decide whether a given $\mathcal{L}$-unification problem is $\mathcal{L}$-unifiable. $\mathcal{EL}$-unification was shown to be NP-complete in [6]. We will show that this decision problem is harder in the less expressive DL $\mathcal{EL}^{-\top}$; to be precise, it is PSpace-complete.

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 3 would imply $A \equiv \sigma(X) \equiv B$ and thus $A \equiv B$, which is not the case.

In the following, we will assume that unification problems are *flat*, i.e., they consist of equations between flat concept terms. By introducing new concept variables and eliminating $\top$, every $\mathcal{EL}^{-\top}$-unification problem Γ can be transformed in polynomial time into a flat $\mathcal{EL}^{-\top}$-unification problem Γ' such that Γ is solvable iff Γ' is solvable [6].

Given a flat 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 occurring concept terms. 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.

We now show that when searching for unifiers, we may restrict ourselves to ground unifiers that only contain the role names and constants that occur in the considered unification problem.

Lemma 5. *Let Γ be a flat unification problem with $\mathcal{EL}^{-\top}$-unifier γ. Then there is a ground $\mathcal{EL}^{-\top}$-unifier γ' of Γ such that, for every variable X, the concept term $\gamma'(X)$ contains only constants and role names that occur in Γ.*[2]

[2] To be exact, we have to assume that Γ contains at least one constant and at least one role name, which can always be satisfied by adding a trivial equation like $\exists r.A \equiv^? \exists r.A$ to Γ.

Proof. Let r be a role name and A a constant that both occur in Γ. We define the renaming function f that maps each concept term C to the concept term $f(C)$, where every occurence of a role name that is not in Γ is replaced by r and every occurrence of a constant that is not in Γ and every occurrence of a variable is replaced by A.

It is easy to show that f preserves subsumptions, i.e., $C \sqsubseteq D$ implies $f(C) \sqsubseteq f(D)$ for all concept terms C, D. This can be shown using well-founded induction on the lexicographic order on pairs (C, D), where the components are ordered by $\leq$ as follows: $C_1 \leq C_2$ iff C_1 is a subterm of C_2.

Thus, if $C \equiv^? D$ is an equation in Γ, $\gamma(C) \equiv \gamma(D)$ implies $f(\gamma(C)) \equiv f(\gamma(D))$. The definition $\gamma'(X) := f(\gamma(X))$ for all variables X clearly yields a ground unifier γ' of Γ that has the claimed property. $\square$

In the following, we will assume that N_R is the set of role names occuring in Γ and N_c is the set of constants occuring in Γ. Since we are only interested in the substitution of variables occurring in Γ, we will also restrict the set N_v to $\mathrm{Var}(\Gamma)$.

3.1 $\mathcal{EL}$-unification by guessing acyclic assignments

The NP-algorithm for $\mathcal{EL}$-unification introduced in [4] 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:

- 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 [4] 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).

3.2 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 2 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

We will now present a series of reductions that enable us to show that the unification problem for $\mathcal{EL}^{-\top}$ is in PSpace. For the remainder of this section, let Γ be a flat $\mathcal{EL}^{-\top}$-unification problem. We assume that Γ is a set of flat subsumptions of the form $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? D$. Every equation $C_1 \sqcap \ldots \sqcap C_n \equiv^? D_1 \sqcap \ldots \sqcap D_m$ in Γ can equivalently be expressed by $n + m$ such subsumptions.

4.1 Modifying the Subsumptions

The first reduction modifies Γ in such a way that only subsumptions of the form $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? X$ remain, where $C_1, \ldots, C_n$ are atoms of Γ and X is a variable.

We will remove all other subsumptions from Γ, but introduce new subsumptions of the form $C \sqsubseteq^? X$, where C is an atom and X a variable.

To this purpose, we guess a function $\tau : \mathrm{At}(\Gamma)^2 \to \{0,1\}$, which specifies which subsumptions between atoms of Γ should hold for the $\mathcal{EL}^{-\top}$-unifier we are looking for. The assignment $\tau(D_1, D_2) = 1$ for $D_1, D_2 \in \mathrm{At}(\Gamma)$ means that we restrict our search to substitutions σ satisfying $\sigma(D_1) \sqsubseteq \sigma(D_2)$. 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 6. 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 different constants $A_1, A_2 \in \mathrm{At}(\Gamma)$.
   (c) $\tau(\exists r.C_1, \exists s.C_2) = 0$ for different role names r, s 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 represents a unifier 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 $\tau(C_i, C) = 1$.

Note that these conditions express the nearly same restrictions on τ as the propositional clauses that were constructed in [5] to show that $\mathcal{EL}$-unifiability is in NP. There it was shown in Proposition 3.7 that γ^τ is actually an $\mathcal{EL}$-unifier of Γ. From this fact it follows that if Γ has no $\mathcal{EL}$-unifier, then it is impossible to guess τ with the restrictions above.

It is important to note that 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. We will later construct such $\mathcal{EL}^{-\top}$-unifiers of Γ by adding particles to the sets $\mathrm{Part}(\gamma^\tau(X))$. In the process some subsumptions that are satisfied by γ^τ will become unsatisfied. However, if $\tau(C_1, C_2) = 1$ holds, then the subsumption $C_1 \sqsubseteq^? C_2$ will never be violated. Since we have no way of knowing beforehand which of these subsumptions will still hold, we guess them nondeterministically by guessing τ. It is clear that guessing τ and checking the above conditions can be done in NP.

We now specify the new unification problem $\Delta_{\Gamma,\tau}$ that contains only simple subsumptions that have a single variable on the right-hand side. It consists of the two parts Δ_Γ and Δ_τ, which are defined as follows:

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

Finally, we set $\Delta_{\Gamma,\tau} := \Delta_\Gamma \cup \Delta_\tau$.

For an arbitrary $\mathcal{EL}^{-\top}$-substitution σ we will in the following write $S^\tau \leq S^\sigma$ if

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

holds for every variable X. Before we can show a connection between the unification problems Γ and $\Delta_{\Gamma,\tau}$, we need the following auxiliary lemma. We show that under some conditions on an $\mathcal{EL}^{-\top}$-substitution σ (most importantly $S^\tau \leq S^\sigma$), we can infer $\sigma(C) \sqsubseteq \sigma(D)$ from $\tau(C, D) = 1$ for $C \in \mathrm{At}(\Gamma)$ and $D \in \mathrm{NV}(\Gamma)$.

Lemma 7. *Let Γ be a flat $\mathcal{EL}^{-\top}$-unification problem, τ a subsumption mapping for Γ, and σ an $\mathcal{EL}^{-\top}$-substitution with $S^\tau \leq S^\sigma$. For all atoms $C \in \mathrm{At}(\Gamma)$ and $D \in \mathrm{NV}(\Gamma)$, the following holds:*

- *If D is ground, then $\tau(C, D) = 1$ implies $\sigma(C) \sqsubseteq \sigma(D)$.*
- *If $D = \exists r.Y$ and σ satisfies all subsumptions of the form $C' \sqsubseteq^? Y$ in Δ_τ, then $\tau(C, D) = 1$ implies $\sigma(C) \sqsubseteq \sigma(D)$.*

Proof. If C is a variable, then $\tau(C, D) = 1$ implies $D \in S^\tau(C) \subseteq S^\sigma(C)$, and thus $\sigma(C) \sqsubseteq \sigma(D)$ by the definition of S^σ. Otherwise, we consider the structure of D. If D is a constant, then the Conditions 1(b) and 1(d) of Definition 6 yield $C = D$, and the subsumption is clearly satisfied.

If D is not a constant, then it is of the form $\exists r.D'$. By the Conditions 1(c)–(e) of Definition 6, C must be of the form $\exists r.C'$ and $\tau(C', D') = 1$. It remains to show that $\sigma(C') \sqsubseteq \sigma(D')$ holds.

If D' is a constant, then either $C' = D'$, in which case we immediately have $\sigma(C') \sqsubseteq \sigma(D')$, or C' is a variable and $D' \in S^\tau(C') \subseteq S^\sigma(C')$. In the latter case, the claim follows from the definition of S^σ.

It only remains to consider the case that D' is a variable. Then, $C' \sqsubseteq D'$ is a subsumption in Δ_τ and we have $\sigma(C') \sqsubseteq \sigma(D')$ by assumption. $\square$

We can now show the following characterization regarding the two unification problems Γ and $\Delta_{\Gamma,\tau}$.

Lemma 8. *Let Γ be a flat $\mathcal{EL}^{-\top}$-unification problem. Then the following statements are equivalent:*

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

Proof. If Γ has a ground $\mathcal{EL}^{-\top}$-unifier σ, we can define τ as $\tau(D_1, D_2) = 1$ iff $\sigma(D_1) \sqsubseteq \sigma(D_2)$ holds for $D_1, D_2 \in \mathrm{At}(\Gamma)$. It is easy to see that σ satisfies all the subsumptions in $\Delta_{\Gamma,\tau}$, and $S^\tau \leq S^\sigma$. Additionally, τ is a solution mapping:

- Conditions 1(a)–(f) of Definition 6 are satisfied by the subsumption relation.
- Conditions 3(a) and 3(b) of Definition 6 are satisfied, since σ is a unifier of Γ and Lemma 2 holds.
- Assume that there is a sequence $X_1, \ldots, X_n$ $(n > 1)$ of variables such that $X_1 = X_n$ and $\sigma(X_i) \sqsubseteq \sigma(\exists r_i.X_{i+1})$ for each $i \in \{1, \ldots, n-1\}$. By the properties of subsumption, this would imply $\sigma(X_1) \sqsubseteq \sigma(\exists r_1 \ldots r_{n-1}.X_1)$, which is impossible. Thus, Condition 2 of Definition 6 is also satisfied.

Conversely, let $\tau : \mathrm{At}(\Gamma)^2 \to \{0,1\}$ be a subsumption mapping for Γ and σ be an $\mathcal{EL}^{-\top}$-unifier of $\Delta_{\Gamma,\tau}$ with $S^\tau \leq S^\sigma$. We will show that σ also satisfies all discarded subsumptions of the form $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? D \in \Gamma$, where D is a non-variable atom of Γ.

By Condition 3(a) of Definition 6, there is an index $i \in \{1, \ldots, n\}$ with $\tau(C_i, D) = 1$. Since σ satisfies all the subsumptions in Δ_τ, we can apply Lemma 7 and get $\sigma(C_i) \sqsubseteq \sigma(D)$. Thus, σ satisfies all subsumptions of Γ. $\square$

4.2 Linear Language Inclusions

For the next section, we fix a subsumption mapping $\tau : \mathrm{At}(\Gamma)^2 \to \{0,1\}$ for Γ. We will show that unifiability of $\Delta_{\Gamma,\tau}$ (with an $\mathcal{EL}^{-\top}$-unifier σ satisfying $S^\tau \leq S^\sigma$) can be characterized by the existence of a certain solution to a finite system of linear language inclusion over the alphabet N_R and the indeterminates $X_1, \ldots, X_n$. These inclusions are of the form

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

where $X \in \{X_1, \dots, X_n\}$ and each L_i ($i \in \{0, \dots, n\}$) is a subset of $N_R \cup \{\varepsilon\}$.

A *solution* θ of a set of such inclusions assigns sets of words $\theta(X_i) \subseteq N_R^*$ to the indeterminates X_i such that the specified inclusions hold. The operation $\cup$ is the union of sets and $L_i X_i$ stands for the element-wise concatenation of the languages L_i and $\theta(X_i)$. We additionally define $\theta(L_i X_i) := L_i \theta(X_i)$ and $\theta(L_0) := L_0$.

We will now build the finite set $\mathcal{I}_{\Gamma,\tau}$ of inclusions corresponding to $\Delta_{\Gamma,\tau}$. 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 $\mathfrak{s}$ of the form $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? X$ in $\Delta_{\Gamma,\tau}$, we add the following linear inclusion $I_A(\mathfrak{s})$ to $\mathcal{I}_{\Gamma,\tau}$:

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

One can see that all the inclusions $I_A(\mathfrak{s})$ for $\mathfrak{s} \in \Delta_{\Gamma,\tau}$ are of the form (1) since $\Delta_{\Gamma,\tau}$ only contains flat atoms. For example, the subsumption

$$\exists s.A \sqcap B \sqcap \exists r.X \sqcap Y \sqcap A \sqsubseteq^? X$$

for constants A, B, role names r, s and variables X, Y would be translated into the two inclusions

$$X_A \subseteq \{\varepsilon, s\} \cup \{r\} X_A \cup Y_A \ ,$$
$$X_B \subseteq \{\varepsilon\} \cup \{r\} X_B \cup Y_B \ ,$$

if we assume that A, B are the only constants that occur in Γ.

Intuitively, the solutions of these inclusions represent sets of particles that can be added to the corresponding variables. If θ is a solution of $\mathcal{I}_{\Gamma,\tau}$, X is a variable, A a constant and $w \in \theta(X_A)$, then the particle $\exists w.A$ is a candidate for the set $\mathrm{Part}(\sigma(X))$ for some unifier σ of $\Delta_{\Gamma,\tau}$. We will formalize this connection in the proof of Lemma 9.

In the following, we are only interested in solutions to $\mathcal{I}_{\Gamma,\tau}$ that have the following properties. A solution θ of $\mathcal{I}_{\Gamma,\tau}$ is called *admissible* if, for every variable $X \in N_v$, there is a constant $A \in N_c$ such that $\theta(X_A)$ is nonempty. The solution θ is called *finite* if all the sets $\theta(X_A)$ are finite.

Lemma 9. *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. Let σ be a ground $\mathcal{EL}^{-\top}$-unifier of $\Delta_{\Gamma,\tau}$ with $S^\tau \leq S^\sigma$. We define a solution θ of $\mathcal{I}_{\Gamma,\tau}$ as follows: for each variable X and constant A, we set

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

To check that θ is a solution of $\mathcal{I}_{\Gamma,\tau}$, consider the inclusion $I_A(\mathfrak{s})$ for some $\mathfrak{s}$ of the form $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? X$ in $\Delta_{\Gamma,\tau}$ and a word $w \in \theta(X_A)$. By Lemma 3, we have $\sigma(X) \sqsubseteq \exists w.A$, and thus, Lemma 2 implies that there is a C_i such that $\sigma(C_i) \sqsubseteq \exists w.A$. Hence, $\exists w.A$ is a particle of $\sigma(C_i)$. We show that this implies that $w \in \theta(f_A(C_i))$ by considering the structure of C_i.

(i) If C_i is a constant, then it must be A, since $\exists w.A$ is one of its particles. Then $w = \varepsilon$ and thus, $w \in f_A(C_i) = \theta(f_A(C_i))$.

(ii) If $C_i = Y$ is a variable, then $w \in \theta(Y_A) = \theta(f_A(C_i))$ by definition.

(iii) If C_i is of the form $\exists r.C'$ for a role name r and a constant or variable C', then w must be of the form rw' for $w' \in N_R^*$ and $\exists w'.A$ must be a particle of $\sigma(C')$. Applying the considerations from cases (i) and (ii) to C' and w' yields $w' \in \theta(f_A(C'))$ and thus, $w = rw' \in \{r\}\theta(f_A(C')) = \theta(f_A(C_i))$.

In all of the above cases, we have $w \in \theta(f_A(C_i))$, which implies that θ satisfies $I_A(\mathfrak{s})$. Furthermore, θ is finite, since $\sigma(X)$ can have only finitely many particles. Additionally, since σ is a ground $\mathcal{EL}^{-\top}$-substitution, for every variable X there is at least one word $w_X \in N_R^*$ and constant $A_X \in N_c$ such that $\exists w_X.A_X \in \mathrm{Part}(\sigma(X))$. This implies that θ is also admissible.

Conversely, let θ be a finite, admissible solution of $\mathcal{I}_{\Gamma,\tau}$. We define the $\mathcal{EL}^{-\top}$-substitution σ by induction on the dependency order $>$ induced by S^τ as follows. Let X be a variable and assume that $\sigma(Y)$ has already been defined for all variables Y with $X > Y$. 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 actually an $\mathcal{EL}^{-\top}$-substitution. The property $S^\tau \leq S^\sigma$ follows from the fact that, for each $D \in S^\tau(X)$, the atom $\sigma(D)$ is a top-level atom of $\sigma(X)$ and thus, $\sigma(X) \sqsubseteq \sigma(D)$ holds. It remains to show that σ is a unifier of $\Delta_{\Gamma,\tau}$.

We will show that σ satisfies all subsumptions in $\Delta_{\Gamma,\tau}$ using induction on the total order $>$ on the variables. Let X be a variable and let σ satisfy all subsumptions $D_1 \sqcap \ldots \sqcap D_m \sqsubseteq^? Y$ in $\Delta_{\Gamma,\tau}$ for all variables Y with $X > Y$. We consider a

subsumption $\mathfrak{s}$ of the form $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? X$ in $\Delta_{\Gamma,\tau}$

and have to show that every top-level atom of $\sigma(X)$ subsumes some $\sigma(C_i)$. There are two kinds of top-level atoms of $\sigma(X)$.

If $D \in S^\tau(X)$, then $\tau(X, D) = 1$ and $\sigma(D)$ is a top-level atom of $\sigma(X)$. If $\mathfrak{s} \in \Delta_\Gamma$, then Condition 3(b) of Definition 6 implies that there is a C_i with $\tau(C_i, D) = 1$.

But also in the case that $\mathfrak{s} \in \Delta_\tau$, we know that $\mathfrak{s}$ is of the form $C_1 \sqsubseteq X$ and $\tau(C_1, X) = 1$ holds. By condition I.6, we deduce that $\tau(C_i, D) = 1$ holds for $i = 1$.

By definition of the order $>$, the non-variable atom D can only contain a variable Y with $X > Y$. By the induction hypothesis, σ satisfies all subsumptions from Δ_τ having variables smaller than X w.r.t. $>$ on the right-hand side. Thus, we can apply Lemma 7 to conclude that $\sigma(C_i) \sqsubseteq \sigma(D)$ holds.

The other top-level atoms of $\sigma(X)$ that we have to consider are of the form $\exists w.A$ for $A \in N_c$ and $w \in \theta(X_A)$. Since θ is a solution of $\mathcal{I}_{\Gamma,\tau}$, it satisfies the inclusion $I_A(\mathfrak{s})$, which implies that there is a C_i such that $w \in \theta(f_A(C_i))$. We consider the following cases:

(i) If $C_i = A$, then $w \in \theta(\{\varepsilon\}) = \{\varepsilon\}$ implies $w = \varepsilon$ and thus, $\sigma(C_i) = A = \exists w.A$. C_i cannot be a constant other than A, since this would imply $w \in \theta(\emptyset) = \emptyset$.

(ii) In the case that $C_i = Y$ is a variable, we have $w \in \theta(Y_A)$. Thus, $\exists w.A$ is a top-level atom of $\sigma(Y) = \sigma(C_i)$, which implies $\sigma(C_i) \sqsubseteq \exists w.A$.

(iii) In the remaining case that $C_i = \exists r.C'$ for a role name r and a variable or constant C', we have $w \in \theta(\{r\} f_A(C'))$. Thus, w is of the form rw' for $w' \in \theta(f_A(C'))$. Applying the considerations from cases (i) and (ii) to C' and w' yields the subsumption $\sigma(C') \sqsubseteq \exists w'.A$, which implies $\sigma(C_i) = \exists r.\sigma(C') \sqsubseteq \exists r.\exists w'.A = \exists w.A$. □

4.3 Maximal Solutions

To obtain a PSpace-decision procedure for the unification problem in $\mathcal{EL}^{-\top}$, we need to check for finite, admissible solutions of finite sets of inclusions of the form (1). Later, we will construct automata that recognize exactly the maximal solutions of such problems. In this section, we bridge the gap between finite and maximal admissible solutions.

In the following, let $\mathcal{I}$ be a finite set of inclusions of the form (1), Σ be the underlying alphabet, and $\mathrm{Ind}(\mathcal{I})$ denote the set of indeterminates occurring in $\mathcal{I}$.

First, we establish the existence of a maximal solution for such problems w.r.t. to the following order on solutions. If θ_1 and θ_2 are two solutions of $\mathcal{I}$, then we write $\theta_1 \subseteq \theta_2$ if $\theta_1(X) \subseteq \theta_2(X)$ holds for all $X \in \mathrm{Ind}(\mathcal{I})$. Furthermore, if $(\theta_i)_{i \in I}$ is a family of solutions, then their *union* is defined as the mapping $\left(\bigcup_{i\in I} \theta_i\right)(X) := \bigcup_{i \in I} \theta_i(X)$ for all $X \in \mathrm{Ind}(\mathcal{I})$.

Lemma 10. *The set of all solutions of $\mathcal{I}$ is closed under arbitrary unions and has a maximal element w.r.t. $\subseteq$.*

Proof. Let $(\theta_i)_{i\in I}$ be a nonempty family of solutions of $\mathcal{I}$ and consider an inclusion

$$X \subseteq L_0 \cup L_1X_1 \cup \ldots \cup L_nX_n$$

of $\mathcal{I}$. We have

$$\begin{aligned}\bigcup_{i\in I}\theta_i(X) &\subseteq \bigcup_{i\in I}\big(L_0 \cup L_1\theta_i(X_1) \cup \ldots \cup L_n\theta_i(X_n)\big)\\ &= L_0 \cup L_1\left(\bigcup_{i\in I}\theta_i(X_1)\right) \cup \ldots \cup L_n\left(\bigcup_{i\in I}\theta_i(X_n)\right)\end{aligned}$$

by monotonicity of $\cup$ w.r.t. $\subseteq$ and idempotency, associativity, and commutativity of $\cup$. Thus, $\bigcup_{i\in I}\theta_i$ is also a solution of $\mathcal{I}$.

The empty union $\theta_\emptyset$, for which $\theta_\emptyset(X) = \emptyset$ holds for all indeterminates X, is always a trivial solution of $\mathcal{I}$. These facts imply the existence of a maximal solution of $\mathcal{I}$ w.r.t. $\subseteq$. □

Now we establish a connection between finite and maximal solutions that map some variables to a nonempty set.

Lemma 11. *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 $u \in L_i\theta(X_i)$. □

Thus, in order to check whether Γ is unifiable in $\mathcal{EL}^{-\top}$, we only have to check for admissibility of the maximal solution of $\mathcal{I}_{\Gamma,\tau}$.

Lemma 12. *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 11) 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. □

4.4 Recognizing the Maximal Solution

We will now construct several finite automata $\mathcal{A}_X$, one for each indeterminate, that compute the sets $\theta^*(X)$ of the maximal solution θ^* of $\mathcal{I}$. The automata model we will use is that of *alternating finite automata*, which can make two kinds of transitions: traditional, nondeterministic transitions that "guess" the next state of the automaton; and "universal" transitions that force the automaton to explore every possible successor state. One can imagine these universal transitions as the splitting of the automaton into several copies, each of which goes into one possible successor state and continues the computation independently.

Definition 13. An *alternating finite automaton with ε-transitions (ε-AFA)* $\mathcal{A} = (Q_\exists, Q_\forall, \Sigma, q_0, \delta, F)$ consists of

- two finite, disjoint sets $Q_\exists, Q_\forall$ of *(existential/universal) states* (we will write Q for $Q_\exists \cup Q_\forall$),
- a finite alphabet Σ of *input symbols*,
- an *initial state* $q_0 \in Q$,
- a *transition function* $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$ and
- a set $F \subseteq Q$ of *final states.*

A *configuration* of $\mathcal{A}$ is a pair (q, w), where $q \in Q$ and $w \in \Sigma^*$. The transition function δ induces the following binary relation $\vdash_\mathcal{A}$ between configurations: $(q, w) \vdash_\mathcal{A} (q', w')$ iff either

- $w = w'$ and $q' \in \delta(q, \varepsilon)$ (ε-*transition*) or
- $w = \alpha w'$ and $q' \in \delta(q, \alpha)$ for some $\alpha \in \Sigma$ (α-*transition*).

Note that the second kind of transition is only possible if $w \neq \varepsilon$, i.e., there is still a part of the input word left to read.

A *run* of $\mathcal{A}$ is a finite, nonempty tree labeled by configurations of $\mathcal{A}$ that satisfies the following conditions. If (q, w) is the label of some node and $q \in Q_\exists$, then the node has exactly one successor labeled by a configuration (q', w') with $(q, w) \vdash_{\mathcal{A}} (q', w')$. If (q, w) is the label of some node and $q \in Q_\forall$, then for all configurations (q', w') with $(q, w) \vdash_{\mathcal{A}} (q', w')$ there is exactly one successor of the node labeled by (q', w').

An ε-*path* is a path in this tree that consists only of ε-transitions. A run is called *successful* iff for every leaf one of the following conditions holds. If (q, w) is the label of the leaf, then either $q \in F$ and $w = \varepsilon$ or $q \in Q_\forall$ and there is no configuration (q', w') with $(q, w) \vdash (q', w')$.

An input word $w \in \Sigma^*$ is *accepted* by $\mathcal{A}$ iff there is a successful run of $\mathcal{A}$ the root of which is labeled by (q_0, w). The *language recognized by* $\mathcal{A}$ is $L(\mathcal{A}) := \{w \in \Sigma^* \mid w \text{ is accepted by } \mathcal{A}\}$.

Our goal is to define an ε-AFA $\mathcal{A}_X$ that recognizes exactly $\theta^*(X)$ for one indeterminate X of $\mathcal{I}$. The automaton checks whether the word w can be part of $\theta^*(X)$ using the following ideas. Starting from the indeterminate X, the automaton splits into several copies, each of which checks the restrictions imposed by one inclusion of the form $X \subseteq L_0 \cup L_1X_1 \cup \ldots \cup L_nX_n$. Each of these copies guesses nondeterministically which of the sets $L_0, L_1\theta^*(X_1), \ldots, L_n\theta^*(X_n)$ contains w.

Definition 14. Let $X \in \mathrm{Ind}(\mathcal{I})$. The ε-AFA $\mathcal{A}_X = (Q_\exists, Q_\forall, \Sigma, q_0, \delta, F)$ is defined as follows:

- $Q_\exists := \big(\mathcal{I} \times \{0, \ldots, |\mathrm{Ind}(\mathcal{I})|\}\big) \cup \{f_0\}$,
- $Q_\forall := \big(\mathrm{Ind}(\mathcal{I}) \times \{0, \ldots, |\mathrm{Ind}(\mathcal{I})|\}\big) \cup \{f_1\}$,
- $q_0 := (X, 0)$,
- $F := \{f_0\}$,
- $\delta(f_i, \alpha) := \emptyset$ for every $i \in \{0, 1\}$ and $\alpha \in \Sigma \cup \{\varepsilon\}$,
- $\delta((Y, \lambda), \varepsilon) := \{(\mathfrak{i}, \lambda) \mid \mathfrak{i} : Y \subseteq \ldots \in \mathcal{I}\}$ and $\delta((Y, \lambda), \alpha) := \emptyset$ for all $Y \in \mathrm{Ind}(\mathcal{I})$, $\lambda \in \{0, \ldots, |\mathrm{Ind}(\mathcal{I})|\}$, and $\alpha \in \Sigma$,
- For all inclusions $\mathfrak{i}$ of the form $Y \subseteq L_0 \cup L_1X_1 \cup \ldots \cup L_nX_n$ in $\mathcal{I}$, $\lambda \in \{0, \ldots, |\mathrm{Ind}(\mathcal{I})|\}$ and $\alpha \in \Sigma$,

  $$\begin{aligned}\delta((\mathfrak{i}, \lambda), \varepsilon) &:= \{f_0 \mid \varepsilon \in L_0\} \cup \{g(X_i, \lambda) \mid i \in \{1, \ldots, n\}, \varepsilon \in L_i\}\ , \\ \delta((\mathfrak{i}, \lambda), \alpha) &:= \{f_0 \mid \alpha \in L_0\} \cup \{(X_i, 0) \mid i \in \{1, \ldots, n\}, \alpha \in L_i\}\ .\end{aligned}$$

  The auxiliary function g is defined as $g(X_i, \lambda) := (X_i, \lambda + 1)$ if $\lambda < |\mathrm{Ind}(\mathcal{I})|$ and $g(X_i, \lambda) := f_1$, otherwise.

In the case where there is one inclusion $\mathfrak{i}$ of the form $X \subseteq L_0 \cup L_1 X_1 \cup \ldots \cup L_n X_n$ in $\mathcal{I}$ for which there is a symbol $\alpha \in \Sigma$ with $\varepsilon \notin L_i$ and $\alpha \notin L_i$ for all $i \in \{0, \ldots, n\}$, there is no valid α- or ε-transition from the (existential) state $(\mathfrak{i}, \lambda)$. Thus, $\mathcal{A}_X$ will accept no word starting with α. This is consistent with the restriction imposed by $\mathfrak{i}$ on $\theta^*(X)$, since $\theta^*(X)$ can never contain a word starting with α.

The second component of the states is used to detect ε-cycles. Every time the automaton makes an ε-transition it increases the counter λ in the second component of its state. This counts the number of consecutive states of the form (X, λ) connected only by ε-transitions. If λ grows larger than $|\mathrm{Ind}(\mathcal{I})|$, some indeterminate must have occured twice, i.e., there must have been an ε-cycle. The automaton then goes to f_1, i.e., it accepts everything that follows. The use of this cycle detection mechanism is illustrated in the following example.

Example 15. Let $\mathcal{I}$ consist of the three inclusions

$$\mathfrak{i}_1 : X \subseteq \{r\} \cup \{\varepsilon\}Y, \ \mathfrak{i}_2 : Y \subseteq \{\varepsilon\}X, \text{ and } \mathfrak{i}_3 : Y \subseteq \{s\}.$$

Consider Figure 1, which shows the only successful run of $\mathcal{A}_Y$ accepting $s \in \theta^*(Y)$.

Intuitively, the automaton starts by asking whether s can be an element of $\theta^*(Y)$. From $\mathfrak{i}_3$ it can derive no contradiction, while from $\mathfrak{i}_2$ it derives the information that this is possible only if s is also an element of $\theta^*(X)$. It then proceeds to the inclusion $\mathfrak{i}_1$, which again redirects it to Y. In essence, at this point it has the following information: s can be an element of $\theta^*(Y)$ only if s is an element of $\theta^*(Y)$. Thus, the automaton can affirm the question, since θ^* is the maximal solution and will certainly contain a word if there is no reason against it.

One can see that in order to accept anything at all, the restriction on the length of ε-paths is necessary. Otherwise, there would be no successful run starting in the configuration $((Y, 0), s)$.

Intuitively, if the automaton has already checked the restrictions imposed on a particular indeterminate, then it does not need to check them again. Thus, in a successful run everything that lies below the second occurrence of an indeterminate on the same ε-path can be ignored.

We want to construct an automaton that is of polynomial size in the size of $\mathcal{I}$. Thus, in order to detect these cycles, the automaton cannot simply remember every indeterminate that has already been visited on the current ε-path. Instead, we use the indirect approach to detect cycles via the length of ε-paths.

Lemma 16. *Let $X \in \mathrm{Ind}(\mathcal{I})$ and θ^* be the maximal solution of $\mathcal{I}$. Then $L(\mathcal{A}_X) = \theta^*(X)$.*

Proof. If $w \in L(\mathcal{A}_X)$, then there is a successful run R of $\mathcal{A}_X$ on w. Let V denote the set of nodes of R and $l(v)$ the label of the node $v \in V$. We restrict the set of

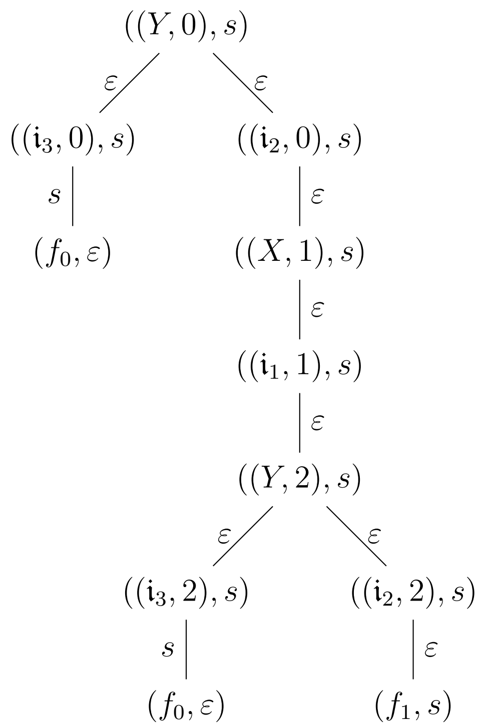

Figure 1: A successful run of the automaton $\mathcal{A}_Y$.

nodes to a subset $V' \subseteq V$ as follows. Intuitively, since we used the restriction on the length of ε-paths only to detect if one indeterminate has occurred twice, we now remove the unnecessary parts from R, i.e., the parts of R below the second occurrence of an indeterminate on an ε-path.

Formally, for every leaf of R labeled by (f_1, u) for some word $u \in \Sigma^*$, there must be an ε-path with nodes labeled by

$$((X_1, 0), u), ((X_2, 1), u), \ldots, ((X_{|\mathrm{Ind}(\mathcal{I})|+1}, |\mathrm{Ind}(\mathcal{I})|), u), (f_1, u)$$

that ends in this leaf. We consider the smallest $j \in \{1, \ldots, |\mathrm{Ind}(\mathcal{I})| + 1\}$ that marks the second occurrence of an indeterminate on this path. We remove the node labeled by $((X_j, j-1), u)$ and all nodes below it from V. After we have done this for every leaf labeled by f_1, V' no longer contains a node with an outgoing edge to f_1.

We now define the solution θ_R by

$$\theta_R(Y) := \{u \in \Sigma^* \mid \exists v \in V' : l(v) = ((Y, \ldots), u)\}$$

for all $Y \in \mathrm{Ind}(\mathcal{I})$.

To show that this actually defines a solution of $\mathcal{I}$, we consider an inclusion

$$\mathfrak{i} : Y \subseteq L_0 \cup L_1 X_1 \cup \ldots \cup L_n X_n$$

from $\mathcal{I}$ and a word $u \in \theta_R(Y)$. Then there is a node $v \in V'$ labeled by $((Y,\lambda),u)$ for some $\lambda \in \{0,\ldots,|\mathrm{Ind}(\mathcal{I})|\}$. This node must have a successor $v' \in V'$ labeled by $((\mathfrak{i},\lambda),u)$, which in turn has a single successor $v'' \in V$. We make a case distinction on the label $l(v'') = (q,u')$.

- If $u = u'$, then $q \in \delta((\mathfrak{i},\lambda),\varepsilon)$. Then either $q = f_0$, which implies $u = \varepsilon \in L_0$, since R is successful, or $q = g(X_i,\lambda)$ for some $i \in \{1,\ldots,n\}$ with $\varepsilon \in L_i$.

  In the second case, λ must be smaller than $|\mathrm{Ind}(\mathcal{I})|$ by construction of V'. If $v'' \in V'$, then $q = (X_i,\lambda+1)$ implies that $u \in \theta_R(X_i) = \{\varepsilon\}\theta_R(X_i) \subseteq L_i\theta_R(X_i)$. If $v'' \notin V'$, there is an ancestor $\widetilde{v} \in V'$ of v'' with $l(\widetilde{v}) = ((X_i,\lambda'),u)$ and $\lambda' \leq \lambda$, since v'' marks the second occurrence of an indeterminate on an ε-path. In this case, we also have $u \in \theta_R(X_i) \subseteq L_i\theta_R(X_i)$.

- If $u = \alpha u'$ for $\alpha \in \Sigma$, then $q \in \delta((\mathfrak{i},\lambda),\alpha)$. Either $q = f_0$ and $\alpha \in L_0$, which implies $u' = \varepsilon$, since R is successful. In this case, we have $u = \alpha \in L_0$. The other possibility is that $q = (X_i,0)$ for some $i \in \{1,\ldots,n\}$ with $\alpha \in L_i$. In this case, v'' must be an element of V' and thus, $u' \in \theta_R(X_i)$, which implies $u = \alpha u' \in \{\alpha\}\theta_R(X_i) \subseteq L_i\theta_R(X_i)$.

In every case, u is also contained in the substitution of the right-hand side of $\mathfrak{i}$ under θ_R. Thus, θ_R is a solution of $\mathcal{I}$. Furthermore, $w \in \theta_R(X)$, since the root of R is contained in V' and labeled by $((X,0),w)$. Concluding, we have $w \in \theta_R(X) \subseteq \theta^*(X)$, since θ^* is the maximal solution of $\mathcal{I}$.

For the other direction, let $w \in \theta^*(X)$. We construct a run of $\mathcal{A}_X$ on w as follows. For every node v, we maintain the invariant $P(v)$ that $u \in \theta^*(Y)$ holds if the node is labeled by $((Y,\ldots),u)$ or $((\mathfrak{i},\ldots),u)$ for some inclusion $\mathfrak{i} \in \mathcal{I}$ with Y on the left-hand side.

The root v_0 is labeled by $((X,0),w)$ and satisfies $P(v_0)$ by assumption. Let now v be a node of the run that already satisfies $P(v)$.

- If $l(v) = ((Y,\lambda),u)$, then $P(v)$ implies $u \in \theta^*(Y)$. For every $\mathfrak{i} \in \mathcal{I}$ having Y on the right hand side, we introduce a successor $v_{\mathfrak{i}}$ of v that is labeled by $((\mathfrak{i},\lambda),u)$. $P(v_{\mathfrak{i}})$ follows directly from $P(v)$.

- If $l(v) = ((\mathfrak{i},\lambda),u)$ for an inclusion

  $$\mathfrak{i} : Y \subseteq L_0 \cup L_1X_1 \cup \ldots \cup L_nX_n$$

  in $\mathcal{I}$, then $u \in \theta^*(Y)$. Since θ^* is a solution of $\mathcal{I}$, either $u \in L_0$ or $u \in L_i\theta^*(X_i)$ for some $i \in \{1,\ldots,n\}$. In the first case, we introduce a successor v' of v that is labeled by (f_0,ε). Otherwise, there is $\alpha \in L_i$ such that $u = \alpha u'$ with $u' \in \theta^*(X_i)$. We introduce a single successor v' of v that is labeled as follows.

– If $\alpha = \varepsilon$ and $\lambda < |\mathrm{Ind}(\mathcal{I})|$, then we label v' by $((X_i, \lambda + 1), u')$. $P(v')$ is satisfied, since $u' \in \theta^*(X_i)$.

– If $\alpha = \varepsilon$ and $\lambda = |\mathrm{Ind}(\mathcal{I})|$, then we label v' by (f_1, u').

– If $\alpha \in \Sigma$, then we label v' by $((X_i, 0), u)$. $P(v')$ is again satisfied by the same reason as above.

It is easily verified that all introduced transitions are valid w.r.t. $\vdash_{\mathcal{A}_X}$. Furthermore, the label of any leaf is either (f_0, ε) or contains an universal state without possible successors w.r.t. $\vdash_{\mathcal{A}_X}$, i.e., either f_1 or a state containing an indeterminate Y that does not occur on the left-hand side of any inclusion from $\mathcal{I}$.

The constructed tree is finite, since every ε-path is terminated by f_1 after finitely many steps. Thus, we have constructed a successful run of $\mathcal{A}_X$, which implies $w \in L(\mathcal{A}_X)$. □

By applying this lemma to the problem $\mathcal{I}_{\Gamma,\tau}$, we see that we can construct polynomially many ε-AFA of polynomial size that we can use to test admissibility of the maximal solution θ^* of $\mathcal{I}_{\Gamma,\tau}$.

Lemma 17. *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 $\mathcal{I}_{\Gamma,\tau}$.*

Proof. If we define $\mathcal{A}(X, A) := \mathcal{A}_{X_A}$, then it is easy to see that this automaton can be constructed in polynomial time in the size of $\mathcal{I}_{\Gamma,\tau}$. Lemma 16 shows that it recognizes exactly the language $\theta^*(X_A)$. □

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 18. *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 [16].

Let Γ be a flat $\mathcal{EL}^{-\top}$-unification problem. By Lemma 8, Lemma 9, and Lemma 12, 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 emptiness of two-way alternating finite automata (where in addition to normal and ε-transitions also backwards transitions are allowed) can be tested in PSpace [13], 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

In this section, we reduce the intersection emptiness problem for deterministic finite automata (DFA) to a unification problem in $\mathcal{EL}^{-\top}$. These DFA are a special case of nondeterministic finite automata, which in turn are special AFA.

An *alternating finite automaton (AFA)* $\mathcal{A} = (Q_\exists, Q_\forall, \Sigma, q_0, \delta, F)$ is an ε-AFA with a restricted transition function $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ that does not allow ε-transitions. The semantics of these automata is the same as for ε-AFA, except that the relation $\vdash_\mathcal{A}$ is restricted to non-ε-transitions. The automaton is called *nondeterministic finite automaton (NFA)* if $Q_\forall = \emptyset$ and is then written as $(Q, \Sigma, q_0, \delta, F)$. It is called *deterministic finite automaton (DFA)* if it is an NFA and for each $q \in Q$ and $\alpha \in \Sigma$, the set $\delta(q, \alpha)$ has the cardinality 0 or 1. The transition function is then equivalently expressed as the partial function $\delta' : Q \times \Sigma \to Q$ where $\delta'(q, \alpha) = q'$ iff $\delta(q, \alpha) = \{q'\}$. This definition implies that any DFA has at most one run on any given word.

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 variable X_q. There is only one constant, A, and we 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.

Lemma 19. *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.*

Proof. We prove this by induction on the length of w. If $|w| = 0$, then $\gamma(X_q) \sqsubseteq A$. Thus, A must be a top-level conjunct of $\gamma(X_q)$. Since γ is a unifier of $\Gamma_{\mathcal{A}}$, this can only be the case if $q \in F$. Thus, $w = \varepsilon$ is accepted by $\mathcal{A}_q$.

Let now $w = \alpha' w'$ with $\alpha' \in \Sigma$, $w' \in \Sigma^*$. Since γ is a unifier of $\Gamma_{\mathcal{A}}$,

$$\bigsqcap_{\substack{\alpha\in\Sigma \\ \delta(q,\alpha) \text{ is defined}}} \exists\alpha.\gamma(X_{\delta(q,\alpha)}) \sqsubseteq \exists\alpha' w'.A \ .$$

Thus, we must have $\gamma(X_{\delta(q,\alpha')}) \sqsubseteq \exists w'.A$ by Lemma 1. By induction, we know that w' is accepted by $\mathcal{A}_{\delta(q,\alpha')}$. Thus, $w = \alpha' w'$ is accepted by $\mathcal{A}_q$. $\square$

Together with Lemma 3, 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 with $\exists w.A \in \mathrm{Part}(\gamma_w(X_{q_0}))$.

For the construction of γ_w, we 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 20. *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 using 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 $u_q := \alpha u_{q'}$. $\square$

We can now proceed with the definition of γ_w for a word $w \in \Sigma^*$ that is accepted by $\mathcal{A}$. The unique successful 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 21. *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 successful 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 have to show that γ_w satisfies the subsumption constraint introduced for every state $q \in Q$, i.e.,

$$F_q \sqcap \bigsqcap_{\substack{\alpha \in \Sigma \\ \delta(q,\alpha) \text{ is defined}}} \exists \alpha.\gamma_w(X_{\delta(q,\alpha)}) \sqsubseteq \gamma_w(X_q) \ .$$

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 $F_q = A$. In this case, the subsumption is satisfied. Otherwise, by construction there is a transition $\delta(q, \alpha) = q'$ with $u_q = \alpha u_{q'}$. Since $\exists u'_q.A$ is a top-level conjunct of $\gamma_w(X_{q'})$, we have $\gamma(X_{q'}) \sqsubseteq \exists u_{q'}.A$ and thus, $\exists \alpha.\gamma_w(X_{q'}) \sqsubseteq \exists u_q.A$.
- 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 know $\exists w_{i+1}.\gamma_w(X_{q_{i+1}}) \sqsubseteq \exists w_{i+1} \dots w_n.A$.

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 conjunct of $\gamma_w(X_{q_0})$, i.e., $\gamma_w(X_{q_0}) \sqsubseteq \exists w.A$. $\square$

The *intersection emptiness problem* 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

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

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:

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

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

Lemma 22. *Γ is unifiable in $\mathcal{EL}^{-\top}$ iff $L(\mathcal{A}_1) \cap \ldots \cap L(\mathcal{A}_k) \neq \emptyset$.*

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 19 yields $w \in L(\mathcal{A}_{i,q_{0,i}}) = L(\mathcal{A}_i)$ for each $i \in \{1, \ldots, k\}$. Thus, the intersection of the languages $L(\mathcal{A}_i)$ is nonempty.

Conversely, let $w \in \Sigma^*$ be a word with $w \in L(\mathcal{A}_1) \cap \ldots \cap L(\mathcal{A}_k)$. By Lemma 21, 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 $i \in \{1, \ldots, k\}$ 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 [14, 11], this lemma immediately yields our final theorem:

Theorem 23. *The problem of deciding unifiability in $\mathcal{EL}^{-\top}$ is* PSpace*-hard.*

6 Conclusion

Unification in $\mathcal{EL}$ was introduced in [4] 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

[4]Note that the right-identity rules in SNOMED CT [18] are actually not expressed using complex role inclusion axioms, but through the SEP-triplet encoding [19]. Thus, complex role inclusion axioms are not relevant here.

[6], acyclic TBoxes can be easily handled by a unification algorithm for concept terms.

Surprisingly, it 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 [17]. 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 [20]. These undecidability results also imply undecidability of unification in some expressive DLs (e.g., in $\mathcal{SHIQ}$ [12]).

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 [5], 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 [15]. 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.

Appendices

A Locality

In $\mathcal{EL}$, we have the interesting property that for every solvable unification problem there exists a *local* unifier γ, where $\gamma(X)$ is a conjunction of atoms of the form $\gamma(D)$ for $D \in \mathrm{NV}(\Gamma)$. However, simply extending this notion to $\mathcal{EL}^{-\top}$-unifiers does not give a similar result for $\mathcal{EL}^{-\top}$.

Example 24. Consider the flat $\mathcal{EL}$-unification problem Γ that contains the three equations

$$X \equiv^? Y \sqcap A, \quad Y \sqcap \exists r.X \equiv^? \exists r.X, \quad Z \sqcap \exists r.X \equiv^? \exists r.X.$$

Then the substitutions $\sigma_0 := \{X \mapsto A, Y \mapsto \top, Z \mapsto \top\}$ and $\sigma_1 := \{X \mapsto A, Y \mapsto \top, Z \mapsto \exists r.A\}$ are the only local $\mathcal{EL}$-unifiers of Γ. In fact, we have $\mathrm{NV}(\Gamma) = \{A, \exists r.X\}$, and thus the only possible image for X in a local unifier σ is A (since $\sigma(\exists r.X) = \exists r.\sigma(X)$ obviously cannot be a conjunct of $\sigma(X)$). Since the first equation implies that $A = \sigma(X) \sqsubseteq \sigma(Y)$, we know that $\sigma(Y)$ can only be $\top$ or A. However, the second equation prevents the second possibility. Finally, the third equation ensures that $\sigma(Z)$ is $\top$ or $\exists r.A$.

Note that σ_0 and σ_1 both contain $\top$, and thus are not $\mathcal{EL}^{-\top}$-unifiers. This shows that Γ does not have an $\mathcal{EL}^{-\top}$-unifier that is local in the sense defined above. Nevertheless, Γ has $\mathcal{EL}^{-\top}$-unifiers. For example, the substitution $\gamma_1 := \{X \mapsto A \sqcap \exists r.A,\ Y \mapsto \exists r.A, Z \mapsto \exists r.\exists r.A\}$ is such a unifier.

In this example, the top-level atoms of $\gamma_1(X), \gamma_1(Y), \gamma_1(Z)$ that are not of the form $\gamma(D)$ for some $D \in \mathrm{NV}(\Gamma)$ are all particles of $\gamma(D)$ for some $D \in \mathrm{NV}(\Gamma)$. This motivates the following definition.

Definition 25. The $\mathcal{EL}^{-\top}$-unifier γ of Γ is a *local $\mathcal{EL}^{-\top}$-unifier* of Γ if, for every variable X, each top-level atom of $\gamma(X)$ is

- of the form $\gamma(D)$ for some $D \in \mathrm{NV}(\Gamma)$ or
- a particle of $\gamma(D)$ for some $D \in \mathrm{NV}(\Gamma)$.

There are always only finitely many local $\mathcal{EL}$-unifiers for a given unification problem [4]. In $\mathcal{EL}^{-\top}$, however, it is possible that there exist infinitely many local unifiers, as the next example demonstrates.

Example 26. Consider the unification problem Γ from Example 24 and the following $\mathcal{EL}^{-\top}$-substitutions γ_n:

$$\begin{aligned}\gamma_n(X) &:= A \sqcap \exists r.A \sqcap \cdots \sqcap \exists r^n.A\\ \gamma_n(Y) &:= \exists r.A \sqcap \cdots \sqcap \exists r^n.A\\ \gamma_n(Z) &:= \exists r^{n+1}.A\end{aligned}$$

It is easy to verify that each γ_n is an $\mathcal{EL}^{-\top}$-unifier of Γ. Furthermore, every top-level atom of $\gamma_n(X)$, $\gamma_n(Y)$, and $\gamma_n(Z)$ is either A or a particle of $\gamma_n(\exists r.X)$. Note that both A and $\exists r.X$ are non-variable atoms of Γ. Thus, Γ has infinitely many local $\mathcal{EL}^{-\top}$-unifiers.

Additionally, these unifiers are even incomparable w.r.t. the subsumption order on unifiers, i.e., for no two $n, m \in \mathbb{N}$ with $n \neq m$ it holds that $\gamma_n(X) \sqsubseteq \gamma_m(X)$ for all variables X. This is the case since the concept terms $\gamma_n(Z) = \exists r^{n+1}.A$ are incomparable in this sense.

We will show that checking for local unifiers suffices to decide unifiability in $\mathcal{EL}^{-\top}$ by demonstrating that the decision procedure described in Section 4 can be used to construct local $\mathcal{EL}^{-\top}$-unifiers. To be able to use the reductions to the problems of solvability of sets of language inclusions and emptiness of ε-AFA, we first define appropriate notions of locality for these formalisms.

Definition 27. Let $\mathcal{I}$ be a finite set of inclusions of the form

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

as described in Section 4.2. A solution θ of $\mathcal{I}$ is called *local* if all words $w \in \theta(X) \setminus \{\varepsilon\}$ for $X \in \mathrm{Var}(\mathcal{I})$ occur on the right-hand side of some inclusion $Y \subseteq L_0 \cup L_1 X_1 \cup \ldots \cup L_n X_n$ of $\mathcal{I}$ under θ, i.e., either $w \in L_0$ or $w \in (L_i \setminus \{\varepsilon\})\theta(X_i)$ for some $i \in \{1, \ldots, n\}$.

The final definition is concerned with locality in alternating automata.

Definition 28. Let $\mathcal{A}$ be an ε-AFA. A successful run of $\mathcal{A}$ is called *local* if there is at least one leaf labeled by (q, ε) for some state q of $\mathcal{A}$. Since the run is successful, q is then either a final state or a universal state without possible successors. We denote by $L^l(\mathcal{A})$ the set of all words accepted by $\mathcal{A}$ via local, successful runs.

In a successful run R of $\mathcal{A}$ that is not local, all leafs are labeled by configurations (q, w) with $w \neq \varepsilon$. In this case, q has to be a universal state without successors. However, since such states accept any word, it is easy to change R into a local run. We simply identify the shortest word w that occurs in the label of a leaf. Since R is a run, w is the shortest word occuring in it and all other words in R must have the suffix w. Thus, we can simply remove the suffix w from all configurations in R and obtain a successful run that accepts a shorter word. This new run is local since it must contain at least one leaf labeled by (q, ε) for some state q.

This construction also shows that runs accepting minimal words, i.e., words for which no prefix is accepted by $\mathcal{A}$, are always local. This is an important property of locality in ε-AFA which will prove to be useful.

The following lemma proves a connection between local runs and local solutions by analyzing one direction of Lemma 16 in more detail.

Lemma 29. *Let $\mathcal{I}$ be a finite set of inclusions of the form* (1) *and let the ε-AFA $\mathcal{A}_X$ for a variable $X \in \mathrm{Var}(\mathcal{I})$ be constructed as in Definition 14. If $w \in L^l(\mathcal{A}_X)$, then there is a finite, local solution θ of $\mathcal{I}$ such that $w \in \theta(X)$ and every $w' \in \theta(Y)$ for some $Y \in \mathrm{Var}(\mathcal{I})$ is a suffix of w.*

Proof. Let R be a local, successful run of $\mathcal{A}_X$ starting in $((X,0),w)$ and consider the solution θ_R that was constructed in the proof of Lemma 16:

$$\theta_R(Y) := \{u \in \Sigma^* \mid \exists v \in V' : l(v) = ((Y,\ldots),u)\}$$

for all variables $Y \in \mathrm{Var}(\mathcal{I})$. Since V' is a subset of the finite set of nodes of R, θ_R is finite. By definition of the transition relation of $\mathcal{A}_X$, the run R, and thus also θ_R, contains only suffixes of w. Furthermore, $w \in \theta_R(X)$ since the root node of R is labeled by $((X,0),w)$ and contained in V'. It remains to show that θ_R is local.

Since R is local, there is a leaf of R that is labeled by (q,ε) for some state q of $\mathcal{A}_X$. We now consider the path p leading from the root of R to this leaf. Its root is labeled by $((X,0),w)$, while its leaf is labeled by (q,ε). Thus, every suffix of w must occur along this path. To show locality, it thus suffices to show that every word occuring along p satisfies the conditions on locality. We will show this by backwards induction along p.

We begin the induction at the leaf of p, which is labeled by (q,ε). The word ε trivially fulfills the conditions for locality of θ_R. Let now v' be a node of p labeled by (q',u') for a state q' and a suffix u' of w that fulfills the conditions for locality of θ_R. If v' is the root node, we are done. Otherwise, we show the same for the predecessor v of v', which also lies on the path p. Let (q,u) be the label of v and consider the following cases:

- If $u = u'$, then u fulfills the condition for locality of θ_R, since u' does.
- Otherwise, $u = \alpha u'$ for some $\alpha \in \Sigma$ and q must be of the form $(\mathfrak{i},\lambda)$ for some inequation $\mathfrak{i} : Y \subseteq L_0 \cup L_1X_1 \cup \ldots L_nX_n$ in $\mathcal{I}$. Then the label (q',u') of v' can only have one of the following forms:
  - If $q' = f_0$, then $\alpha \in L_0$. Since R is successful, we then have $u' = \varepsilon$ and $u = \alpha \in L_0$.
  - Otherwise, $q' = (X_i,0)$ for some $i \in \{1,\ldots,n\}$ and $\alpha \in L_i$. But then $u' \in \theta_R(X_i)$ by definition of θ_R and thus, $u = \alpha u' \in \{\alpha\}\theta_R(X_i) \subseteq (L_i \setminus \{\varepsilon\})\theta_R(X_i)$.

  Thus, the word u fulfills the condition of locality since it is contained in the right-hand side of $\mathfrak{i}$ under θ_R. $\square$

In the following, let Γ be a flat $\mathcal{EL}^{-\top}$-unification problem, τ a subsumption mapping for Γ, and γ^τ, $\Delta_{\Gamma,\tau}$, $\mathcal{I}_{\Gamma,\tau}$, and $\mathcal{A}(X,A)$ be defined as in Section 4. Using the previous lemma, under some conditions we can construct a finite, local, admissible solution of $\mathcal{I}_{\Gamma,\tau}$.

Lemma 30. *If for every $X \in \mathrm{Var}(\mathcal{I})$ there is a constant $A(X)$ such that the automaton $\mathcal{A}(X,A(X))$ accepts a word w_X, then there is a finite, local, admissible solution of $\mathcal{I}_{\Gamma,\tau}$ that contains only suffixes of the words w_X.*

Proof. By Lemma 29, we find for every X a finite, local solution θ_X of $\mathcal{I}_{\Gamma,\tau}$ that contains only suffixes of w_X and satisfies $w_X \in \theta_X(X_{A(X)})$. By Lemma 10, the union θ of all θ_X is still a solution of $\mathcal{I}_{\Gamma,\tau}$. It is finite since it is a finite union of finite solutions. It is also admissible since for every X the set $\theta(X_{A(X)})$ is non-empty. Finally, it is local since all contained words satisfy the conditions on locality by locality of the component solutions θ_X. □

The following lemma proves a connection between finite, local, admissible solutions of $\mathcal{I}_{\Gamma,\tau}$ and local unifiers of Γ by analyzing one direction of Lemma 9 in more detail.

Lemma 31. *Let θ be a finite, local, admissible solution of $\mathcal{I}_{\Gamma,\tau}$. Then there is a local $\mathcal{EL}^{-\top}$-unifier σ of Γ.*

Proof. Consider the $\mathcal{EL}^{-\top}$-unifier σ of $\Delta_{\Gamma,\tau}$ constructed in the proof of Lemma 9 which has the property that $S^\tau \leq S^\sigma$. It was defined by induction on the order $>$ on the variables as follows:

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

for every variable X, where $\sigma(Y)$ has already been defined for each variable Y with $X > Y$. In the proof of Lemma 8, it was shown that σ is also a unifier of Γ.

To show that σ is local, we consider all top-level atoms of $\sigma(X)$ for each $X \in \mathrm{Var}(\Gamma)$. For those top-level atoms of the form $\sigma(D)$ for $D \in S^\tau(X)$, this follows immediately from the fact that $S^\tau(X) \subseteq \mathrm{NV}(\Gamma)$. Now consider a top-level particle $\exists w.A$ of $\sigma(X)$. If $w = \varepsilon$, then A is a non-variable atom of Γ since we assumed that all elements of N_C occur in Γ. Otherwise, $w \in \theta(X_A) \setminus \{\varepsilon\}$ and, by locality of θ, there is an inclusion in $\mathcal{I}_{\Gamma,\tau}$ that contains w in the substitution of its right-hand side under θ.

This inclusion must be of the form $I_A(\mathfrak{s})$, i.e., $X_A \subseteq f_A(C_1) \cup \ldots \cup f_A(C_n)$, for some subsumption $\mathfrak{s}$ of the form $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq^? X$ in $\Delta_{\Gamma,\tau}$. Locality of θ yields an index $i \in \{1, \ldots, n\}$ with $w \in \theta(f_A(C_i))$, where C_i is neither a variable nor a constant.[5]

Thus, C_i is of the form $\exists r.C'$, where C' is either a variable or the constant A. Consequently, either $w \in \{r\}$ or $w \in \{r\}\theta(C'_A)$. In the former case, $\exists w.A = \exists r.A = C_i$ is a ground atom of Γ. In the latter case, $w = rw'$ for some $w' \in \theta(C'_A)$. This implies $\sigma(C') \sqsubseteq \exists w'.A$, which yields $\sigma(C_i) \sqsubseteq \exists w.A$. By Lemma 3, $\exists w.A$ is a particle of $\sigma(C_i)$. Since $C_i \in \mathrm{NV}(\Gamma)$, the particle $\exists w.A$ fulfills the condition for locality of σ. □

[5] Recall the definition of $f_A(C)$ from Section 4.2.

Since we want to obtain a complexity result, we also have to consider the size of σ. In the following, *size* always means the number of symbols it takes to write something down and is denoted by $|\cdot|$. For example, for a solution θ of $\mathcal{I}_{\Gamma,\tau}$, $|\theta|$ denotes the number of symbols it takes to write down all the sets $\theta(X_A)$ for $X \in \mathrm{Var}(\Gamma)$ and $A \in N_C$.

Lemma 32. *If θ is a finite, local, admissible solution of $\mathcal{I}_{\Gamma,\tau}$, then the size of the local $\mathcal{EL}^{-\top}$-unifier σ constructed in Lemma 31 is at most exponential in the size of Γ and polynomial in the size of θ.*

Proof. For a variable $X \in \mathrm{Var}(\Gamma)$, we consider all sequences $X_1 < \cdots < X_n = X$ where X_1 is a minimal variable w.r.t. $<$. The length of such a sequence is the number of variables it contains, i.e., n. The *height* of X is defined as the maximal length of all such sequences. This means that the height of a minimal variable is 1 and the height is bounded by $|\mathrm{Var}(\Gamma)|$ since $<$ is acyclic.

We prove the following claim by induction on the height n of the variables $X \in \mathrm{Var}(\Gamma)$: For every $X \in \mathrm{Var}(\Gamma)$,

$$|\sigma(X)| \leq 5^n \left(|\Gamma|^n + |\theta| \left(\sum_{i=0}^{n-1} |\Gamma|^i \right) \right).[6]$$

Let $n = 1$, i.e., X be a minimal variable w.r.t. $<$. Then all non-variable atoms in $S^\tau(X)$ are ground and the size of $\sigma(X)$ is bounded by $5(|S^\tau(X)| + |\theta|) \leq 5(|\Gamma| + |\theta|)$.

If $n > 1$, then we know that the height of all variables $Y < X$ must be smaller than n. Since all the non-variable atoms $D \in S^\tau(X)$ contain only variables smaller than X, by induction we can bound the size of each $\sigma(D)$ for $D \in S^\tau(X)$ by

$$5^{n-1} \left(|\Gamma|^{n-1} + |\theta| \left(\sum_{i=0}^{n-2} |\Gamma|^i \right) \right).$$

Since $S^\tau(X) \subseteq \mathrm{NV}(\Gamma)$, there are at most $|\Gamma|$ elements in $S^\tau(X)$ and the size of $\sigma(X)$ is thus bounded by

$$5 \left(|\Gamma| 5^{n-1} \left(|\Gamma|^{n-1} + |\theta| \left(\sum_{i=0}^{n-2} |\Gamma|^i \right) \right) + |\theta| \right) \leq 5^n \left(|\Gamma|^n + |\theta| \left(\sum_{i=0}^{n-1} |\Gamma|^i \right) \right).$$

Since the height of any variable is bounded by the number of variables, and thus by $|\Gamma|$, this means that the overall size of σ is bounded by

$$|\Gamma| 5^{|\Gamma|} \left(|\Gamma|^{|\Gamma|} + |\theta| \left(\sum_{i=0}^{|\Gamma|-1} |\Gamma|^i \right) \right),$$

[6] The constant 5 accounts for additional symbols like $\sqcap$ or $\exists$ that are added in the definition of σ.

i.e., an expression that is exponential in $|\Gamma|$ and polynomial in $|\theta|$. $\square$

We can now modify the decision procedure from Theorem 18 such that it outputs a local $\mathcal{EL}^{-\top}$-unifier for any solvable unification problem. However, since we actually have to output the unifier, the complexity of the algorithm is higher than for just deciding the existence of a unifier, i.e., ExpTime instead of PSpace.

The algorithm uses the well-known reduction from any alternating automaton to an equivalent nondeterministic automaton of exponential size [9, 8]. Additionally, it employs a polynomial-time algorithm to find shortest paths in a directed graph, e.g., Dijkstra's algorithm [10]. This will be used to find a successful run of the nondeterministic automaton.

Theorem 33. *Given a solvable $\mathcal{EL}^{-\top}$-unification problem Γ, we can construct a local $\mathcal{EL}^{-\top}$-unifier of Γ of at most exponential size in time exponential in the size of Γ.*

Proof. Recall that in Theorem 18 we have guessed the subsumption mapping τ and constructed $\Delta_{\Gamma,\tau}$, $\mathcal{I}_{\Gamma,\tau}$ and the automata $\mathcal{A}(X, A)$ for every variable X and constant A. Then we tested these automata for emptiness and checked whether for every variable there was a constant such that the corresponding automaton accepted a non-empty language. If this was the case, then Γ was unifiable in $\mathcal{EL}^{-\top}$.

To show that we can actually construct a local $\mathcal{EL}^{-\top}$-unifier in exponential time, we start by enumerating all possible subsumption mappings τ. This can be done in exponential time since the size of τ is polynomial in the size of Γ. Since Γ is unifiable in $\mathcal{EL}^{-\top}$, we will find one τ such that for each variable X there is a constant A_X for which the automaton $\mathcal{A}(X, A_X)$ accepts a non-empty language.

For each X, we now construct a nondeterministic automaton $\mathcal{B}_X$ that is equivalent to $\mathcal{A}(X, A_X)$ [8]. This automaton has as state set the powerset of the original state set. A set can be reached from another if these sets are compatible with the reachability in the alternating automaton. This means that for every universal state, all successor states must be in the successor state set; for an existential state, there must be one successor in the set. The size of $\mathcal{B}_X$ is exponential in the size of $\mathcal{A}(X, A_X)$, and thus exponential in the size of Γ.

We now search for a successful run r of $\mathcal{B}_X$ of minimal length, i.e., a shortest path in the transition graph of $\mathcal{B}_X$ that starts in the initial state $\{(X, 0)\}$ and leads to a final state. Such a path can be found in exponential time using, e.g., Dijkstra's algorithm [10]. It is clear that r is of size exponential in Γ and it accepts a word w_X that is of length exponential in Γ.

From the state sets occurring in r the tree shape of the underlying run R of $\mathcal{A}(X, A_X)$ can be extracted by the following procedure. We start with a single root node that is labeled by $((X, 0), w_X)$ and iteratively construct the layers of R of increasing depth. For each existential state in a state set of r, there must be a successor in the next state set. Similarly, for every universal state all its successors can be found in the next state set. Thus, for each configuration occuring in the current tree, we can find a valid transition of $\mathcal{A}(X, A_X)$ and can add the corresponding child nodes to the tree. Since r is finite, this construction terminates. The result is a successful run R of $\mathcal{A}(X, A_X)$ because r ends in a state set containing only final states or universal states without successors. Since the accepted word is of minimal length, R is local.

Thus, for every variable X we can find a word $w_X \in L^l(\mathcal{A}(X, A_X))$ which is of length at most exponential in the size of Γ. By Lemma 29, we can construct a finite, local solution θ_X of $\mathcal{I}_{\Gamma,\tau}$ with $w_X \in \theta_X(X)$ that contains only suffixes of w_X. Thus, θ_X is of size exponential in the size of Γ since it contains at most exponentially many words of size at most exponential in the size of Γ. Lemma 30 yields a finite, local, admissible solution of $\mathcal{I}_{\Gamma,\tau}$ of exponential size. Finally, using Lemmata 31 and 32 we find a local $\mathcal{EL}^{-\top}$-unifier of Γ of size exponential in the size of Γ. $\square$

This provides an *upper bound* on the size of the smallest $\mathcal{EL}^{-\top}$-unifier of a given flat $\mathcal{EL}^{-\top}$-unification problem. We will now present a flat $\mathcal{EL}^{-\top}$-unification problem such that size of any $\mathcal{EL}^{-\top}$-unifier is *at least* exponential in the size of the problem.

Example 34. We will construct solvable $\mathcal{EL}^{-\top}$-unification problems Γ_n ($n \in \mathbb{N}$) such that the size of any $\mathcal{EL}^{-\top}$-unifier of Γ_n is exponential in the size of Γ_n. For this, we consider the proof of PSPACE-completeness of the intersection emptiness problem for DFA [14].

For any deterministic Turing machine $\mathcal{M}$ with polynomial space bound, the proof constructs several DFA $\mathcal{A}_i$ ($i = 1, \ldots, n$) of size polynomial in the size of $\mathcal{M}$. The number n of these automata is also polynomial in the size of $\mathcal{M}$. These automata have the property that any input word u is accepted by $\mathcal{M}$ iff there is a successful run r of $\mathcal{M}$ on u such that $w_r \in \bigcap_{i=1}^{n} L(\mathcal{A}_i)$. Here, the word w_r is a representation of the run r that is constructed by concatenating the content of the tape of $\mathcal{M}$ for each step of the run, i.e., it may be exponentially long. This means that the intersection $\bigcap_{i=1}^{n} L(\mathcal{A}_i)$ contains exactly the representations of all successful runs of $\mathcal{M}$.

For each $n \in \mathbb{N}$, consider the following n-space bounded deterministic Turing machine $\mathcal{M}_n$ with input alphabet $\{0, 1\}$. First, the machine $\mathcal{M}_n$ checks whether the input is equal to 0^n. If it is not, $\mathcal{M}_n$ rejects the word. Otherwise, it views 0^n as the binary representation of the number 0 and then iteratively increases this number by 1 until it reaches 1^n. $\mathcal{M}_n$ can be defined in such a way that it never leaves the tape section defined by the input and is of size polynomial in n. It accepts only the word $u = 0^n$ and has only one successful run r_n on this word. The length of the representation w_{r_n} of r_n is exponential in n since $\mathcal{M}_n$

enumerates exponentially many numbers.

For this deterministic Turing machine $\mathcal{M}_n$, we can now construct k DFA $\mathcal{A}_i$ ($i = 1, \ldots, k$) with $\{w_{r_n}\} = \bigcap_{i=1}^{k} L(\mathcal{A}_i)$, where k and the size of the automata are bounded by a polynomial in n [14]. The equality $\{w_{r_n}\} = \bigcap_{i=1}^{k} L(\mathcal{A}_i)$ holds since by construction $\bigcap_{i=1}^{k} L(\mathcal{A}_i)$ contains exactly the representations of all successful runs of $\mathcal{M}_n$.

Following the proof of Lemma 22, we can construct a flat unification problem Γ_n of size polynomial in n that is unifiable in $\mathcal{EL}^{-\top}$ iff the intersection $\bigcap_{i=1}^{k} L(\mathcal{A}_i)$ is non-empty. We now consider any local $\mathcal{EL}^{-\top}$-unifier γ of Γ_n, which must exist since this intersection contains the word w_{r_n}. By Lemmata 19 and 21, w_{r_n} is the only word such that $\gamma(X_{q_{0,i}}) \sqsubseteq \exists w_{r_n}.A$ holds for all $i = 1, \ldots, k$. Since γ must satisfy $X_{q_{0,i}} \sqsubseteq^? Y$ for each $i = 1, \ldots, n$ and $\gamma(Y)$ must contain at least one particle, this particle can only be $\exists w_{r_n}.A$. This particle is of size exponential in n, which shows that every local $\mathcal{EL}^{-\top}$-unifier of Γ_n is of size exponential in the size of Γ_n.

We thus have the following completeness result for the problem of constructing a local $\mathcal{EL}^{-\top}$-unifier for a flat $\mathcal{EL}^{-\top}$-unification problem.

Corollary 35. *The size of the local $\mathcal{EL}^{-\top}$-unifiers of an $\mathcal{EL}^{-\top}$-unification problem may grow exponentially in the size of the problem. On the other hand, given a solvable $\mathcal{EL}^{-\top}$-unification problem, we can always compute a local $\mathcal{EL}^{-\top}$-unifier in exponential time.* □

References

[1] Franz Baader. Terminological cycles in a description logic with existential restrictions. In Georg Gottlob and Toby Walsh, editors, *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003)*, pages 325–330, Acapulco, Mexico, 2003. Morgan Kaufmann, Los Altos.

[2] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, pages 364–369, Edinburgh (UK), 2005. Morgan Kaufmann, Los Altos.

[3] 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.

[4] Franz Baader and Barbara Morawska. Unification in the description logic $\mathcal{EL}$. In Ralf Treinen, editor, *Proc. of the 20th Int. Conf. on Rewriting Techniques

and Applications (RTA 2009), volume 5595 of *Lecture Notes in Computer Science*, pages 350–364. Springer-Verlag, 2009.

[5] Franz Baader and Barbara Morawska. SAT encoding of unification in $\mathcal{EL}$. In Christian G. Fermüller and Andrei Voronkov, editors, *Proc. of the 17th Int. Conf. on Logic for Programming, Artifical Intelligence, and Reasoning (LPAR-17)*, volume 6397 of *Lecture Notes in Computer Science*, pages 97–111, Yogyakarta, Indonesia, 2010. Springer-Verlag.

[6] Franz Baader and Barbara Morawska. Unification in the description logic $\mathcal{EL}$. *Logical Methods in Computer Science*, 6(3), 2010.

[7] Franz Baader and Paliath Narendran. Unification of concepts terms in description logics. *J. of Symbolic Computation*, 31(3):277–305, 2001.

[8] Jean-Camille Birget. State-complexity of finite-state devices, state compressibility and incompressibility. *Mathematical Systems Theory*, 26(3):237–269, 1993.

[9] Ashok K. Chandra, Dexter C. Kozen, and Larry J. Stockmeyer. Alternation. *J. of the ACM*, 28(1):114–133, 1981.

[10] Edsger W. Dijkstra. A note on two problems in connexion with graphs. 1:269–271, 1959.

[11] Michael R. Garey and David S. Johnson. *Computers and Intractability — A guide to NP-completeness*. W. H. Freeman and Company, San Francisco (CA, USA), 1979.

[12] Ian Horrocks, Ulrike Sattler, and Stefan Tobies. Practical reasoning for very expressive description logics. *J. of the Interest Group in Pure and Applied Logic*, 8(3):239–264, 2000.

[13] Tao Jiang and B. Ravikumar. A note on the space complexity of some decision problems for finite automata. *Information Processing Letters*, 40:25–31, October 1991.

[14] Dexter Kozen. Lower bounds for natural proof systems. *Annual IEEE Symposium on Foundations of Computer Science*, 0:254–266, 1977.

[15] 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.

[16] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970.

[17] 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.

[18] Kent A. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. *J. of the American Medical Informatics Association*, 2000. Fall Symposium Special Issue.

[19] Boontawee Suntisrivaraporn, Franz Baader, Stefan Schulz, and Kent Spackman. Replacing SEP-triplets in SNOMED CT using tractable description logic operators. In Riccardo Bellazzi, Ameen Abu-Hanna, and Jim Hunter, editors, *Proceedings of the 11th Conference on Artificial Intelligence in Medicine (AIME'07)*, volume 4594 of *Lecture Notes in Computer Science*, pages 287–291. Springer-Verlag, 2007.

[20] Frank Wolter and Michael Zakharyaschev. Undecidability of the unification and admissibility problems for modal and description logics. *ACM Trans. Comput. Log.*, 9(4), 2008.