Notre Dame Journal of Formal Logic
Volume 57, Number 4, 2016

Deciding Unifiability and Computing Local Unifiers in the Description Logic $\mathcal{EL}$ without Top Constructor

Franz Baader, Nguyen Thanh Binh,
Stefan Borgwardt, and 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 been shown to be NP-complete and, thus, of considerably lower complexity than unification in other description logics 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. In addition to the decision problem, we also consider the problem of actually computing $\mathcal{EL}^{-\top}$-unifiers.

1 Introduction

Description logics (DLs) are a well-investigated family of logic-based knowledge representation formalisms (see Baader, Calvanese, McGuinness, et al. [8]). 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.

Received January 10, 2012; accepted August 21, 2013
First published online July 19, 2016
2010 Mathematics Subject Classification: Primary 68Q17, 68Q25, 68Q45; Secondary 68T27, 68T30, 03B70, 03B45
Keywords: unification, description logics
10.1215/00294527-3555507

For example, by 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, that is, 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 $\mathcal{EL}$ (see Baader [1], Baader, Brandt, and Lutz [7]). On the other hand, though quite inexpressive, $\mathcal{EL}$ is 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 Baader and Narendran [13] as a novel inference service that can, for instance, 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 represents this notion in a somewhat different way, for example, by using the concept term

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

While these two concept terms are not equivalent, they are nevertheless meant to represent the same concept. They can obviously be made equivalent by substituting the concept name $\mathsf{Professor}$ by the concept term $\mathsf{Teacher} \sqcap \mathsf{Researcher}$ and the concept name Woman by the concept term $\mathsf{Person} \sqcap \mathsf{Female}$.

In general, unification is the problem of making two concept terms equivalent by allowing some of the concept names, which are designated variables, to be replaced by other concept terms. 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 with respect to these definitions.

In [13] 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 Baader and Morawska [10], 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

[10] was a brutal "guess and then test" NP-algorithm, but we have since then also developed more practical algorithms. On the one hand, in Baader and Morawska [12] 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 hand, in Baader and Morawska [11], 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.

One can additionally allow background knowledge of the application domain to be encoded using so-called *general concept inclusions* (GCIs), which restrict one concept to be a subconcept of another concept. For instance, we can use the GCI $\mathsf{Woman} \sqsubseteq \mathsf{Female}$ to express the fact that every woman is female. Since equivalence of concept terms is evaluated with respect to this background knowledge, this changes the unification problem considerably. We were able to show that unification remains in NP if the GCIs satisfy a certain cycle restriction. For example, the cyclic GCI $\exists \mathsf{child}.\mathsf{Human} \sqsubseteq \mathsf{Human}$ satisfies this restriction, whereas the cyclic GCI $\mathsf{Human} \sqsubseteq \exists \mathsf{parent}.\mathsf{Human}$ does not. We again developed three algorithms that generalize the ones for $\mathcal{EL}$ without GCIs (see Baader, Borgwardt, and Morawska [4]–[6]). These algorithms even decide unification in the extension $\mathcal{ELH}_{R^+}$ of $\mathcal{EL}$ that allows one to specify additional domain knowledge in the form of a role hierarchy ($\mathcal{H}$) and transitive roles (R^+).

In contrast to the above works, we here consider a DL that is even less expressive than $\mathcal{EL}$. The motivation for this is that, as mentioned above, SNOMED CT is not formulated in $\mathcal{EL}$, but rather in its sublogic $\mathcal{EL}^{-\top}$, which differs from $\mathcal{EL}$ in that the use of the top concept is disallowed. We also do not consider GCIs since the knowledge in SNOMED CT is expressed by so-called *acyclic concept definitions*, which can be expressed in the unification problem itself (see [12]), thereby eliminating the need to take into account any background knowledge. 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 syntax of the logic influences the unification problem. 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.

This article extends the original conference paper Baader, Binh, Borgwardt, et al. [3] by providing detailed proofs of all results and describing their relevance for the fields of unification modulo equational theories and unification in modal logics. It also incorporates additional results on the complexity of actually computing unifiers in $\mathcal{EL}^{-\top}$ originally published in the workshop paper Baader, Binh, Borgwardt, et al. [2]. To determine unifiability in $\mathcal{EL}$, it is enough to consider local unifiers since every solvable $\mathcal{EL}$-unification problem has a local unifier. Although local unifiers may be of size exponential in the input unification problem, they can be represented by an acyclic TBox (i.e., an acyclic collection of concept definitions) of polynomial size (see [10]). For $\mathcal{EL}^{-\top}$, we have to extend the definition of local unifiers in order to ensure that every solvable unification problem has a local unifier. We will show that, with respect to this new notion of locality, we can effectively compute local unifiers for solvable unification problems, but these unifiers may be of exponential size even if we use acyclic TBoxes to represent them.

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

DLs are logic-based formalisms used to represent the knowledge of an application domain in a structured way (see [8]). Concepts of the domain are described through *concept terms* that are built from atomic concepts (basically, unary predicates) and roles (binary relations) using concept constructors. In this article, we are concerned with the DL $\mathcal{EL}$, which uses the constructors *conjunction* ($\sqcap$), *existential restriction* ($\exists r$ for every role r), and *top concept* ($\top$), and its fragment $\mathcal{EL}^{-\top}$, in which the top concept is disallowed.

More formally, let $\mathsf{N_C}$ and $\mathsf{N_R}$ be two disjoint sets of *concept names* and *role names*, respectively. The set of *$\mathcal{EL}$-concept terms* is the smallest set containing $\mathsf{N_C}$ such that:

- $\top$ is an $\mathcal{EL}$-concept term;
- if C and D are $\mathcal{EL}$-concept terms, then so is $C \sqcap D$; and
- if C is an $\mathcal{EL}$-concept term and $r \in \mathsf{N_R}$, then $\exists r.C$ is an $\mathcal{EL}$-concept term.

The set of *$\mathcal{EL}^{-\top}$-concept terms* is defined in the same way, but using only the latter two rules. Since $\mathcal{EL}^{-\top}$-concept terms are special $\mathcal{EL}$-concept terms, many definitions and results transfer from $\mathcal{EL}$ to $\mathcal{EL}^{-\top}$, and thus we only formulate them for $\mathcal{EL}$. We will explicitly mention it if this is not the case.

The semantics of concept terms is defined 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. This function is extended to $\mathcal{EL}$-concept terms as shown in the semantics column of Table 1. The concept term C is *subsumed* by the concept term D (written $C \sqsubseteq D$) if and only if $C^\mathcal{I} \subseteq D^\mathcal{I}$ holds for all interpretations $\mathcal{I}$; and C is *equivalent* to D ($C \equiv D$) if and only if $C^\mathcal{I} = D^\mathcal{I}$ for every interpretation $\mathcal{I}$.

A *concept definition* is an expression of the form $A \equiv C$, where A is a concept name and C is an arbitrary $\mathcal{EL}^{-\top}$-concept term. An *acyclic TBox* $\mathcal{T}$ is a set of concept definitions such that (i) every concept name occurs at most once on the

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

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

left-hand side of a concept definition in $\mathcal{T}$, and (ii) no concept name is defined in terms of itself, that is, a concept name A does not occur in its own definition (either directly or indirectly through other definitions). The *unfolding* of an $\mathcal{EL}^{-\top}$-concept term C with respect to an acyclic TBox $\mathcal{T}$ (denoted by $C^{\mathcal{T}}$) is the $\mathcal{EL}^{-\top}$-concept term resulting from exhaustively replacing all defined concept names occurring in C by their definitions from $\mathcal{T}$.

2.1 Atoms For unification in $\mathcal{EL}$, it suffices to look at unifiers that substitute variables by conjunctions of so-called *flat atoms* that occur in the unification problem. A concept term is called an *atom* if and only if it is a concept name $A \in \mathsf{N_C}$ or an existential restriction $\exists r.D$. Concept names and existential restrictions $\exists r.D$, where D is a concept name or $\top$, are called *flat atoms*. The set $\mathrm{At}(C)$ of *atoms of a 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 $\mathrm{At}(C) := \mathrm{At}(C_1) \cup \mathrm{At}(C_2)$.

For example, the concept term $C = A \sqcap \exists r.(B \sqcap \exists r.\top)$ has the set of atoms $\mathrm{At}(C) = \{A, \exists r.(B \sqcap \exists r.\top), B, \exists r.\top\}$.

Obviously, any concept term C is a conjunction of atoms $C = C_1 \sqcap \cdots \sqcap C_n$, where the empty conjunction is $\top$ and all conjuncts $\top$ are removed from this conjunction if it is nonempty. We call $C_1, \ldots, C_n$ the *top-level atoms* of C. The concept term C is called *flat* if all its top-level atoms are flat.

The following lemma gives a recursive characterization of subsumption in $\mathcal{EL}$ and $\mathcal{EL}^{-\top}$ and turned out to be very useful for solving unification in these DLs (see [12]).

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

This means that we can check a subsumption $C \sqsubseteq D$ by testing whether for every top-level atom D' of D there is a top-level atom C' of C with $C' \sqsubseteq D'$. A subsumption $C \sqsubseteq D$ between two atoms C, D is then evaluated structurally; that is, either (i) these atoms are the same concept name or (ii) they are of the form $C = \exists r.C'$, $D = \exists r.D'$ for some role name r and $C' \sqsubseteq D'$ holds.

2.2 Particles For unification in $\mathcal{EL}^{-\top}$, building unifiers from the flat atoms of a unification problem is not enough. It will turn out that one may need to add so-called particles to make sure that a variable is not substituted by the empty conjunction of atoms, which is $\top$.

Modulo equivalence, the subsumption relation is a partial order on concept terms. In $\mathcal{EL}$, the top concept $\top$ is the greatest element with respect to this order. If we disallow $\top$, then there are many incomparable maximal concept terms. We will see below that these are exactly the $\mathcal{EL}^{-\top}$-concept terms of the form $\exists r_1.\exists r_2.\cdots\exists r_n.A$ for $n \geq 0$ role names $r_1, \ldots, r_n$ and a concept name A. We call such concept terms *particles*. The set $\mathrm{Part}(C)$ of *particles of an* $\mathcal{EL}^{-\top}$*-concept term C* is defined as follows.

- If C is a concept name, then $\mathrm{Part}(C) := \{C\}$.

- 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 $A \sqcap \exists r.(B \sqcap \exists r.A)$, where $A, B \in \mathsf{N_C}$ and $r \in \mathsf{N_R}$, are A, $\exists r.B$, and $\exists r.\exists r.A$. The next lemma states that particles are indeed the maximal concept terms with respect to subsumption in $\mathcal{EL}^{-\top}$, and that the particles subsuming an $\mathcal{EL}^{-\top}$-concept term C are exactly the particles of C.

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

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

Proof We show both claims by induction on the length of B, that is, 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 $B \equiv C$ by Lemma 1.
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 1.

   Otherwise, $B = \exists r.B'$ for a particle B'. By definition, $B \in \mathrm{Part}(C)$ if and only if there exists 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 equivalent to $C \sqsubseteq B$. □

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

To define unification in $\mathcal{EL}$ and $\mathcal{EL}^{-\top}$ simultaneously, let $\mathcal{L} \in \{\mathcal{EL}, \mathcal{EL}^{-\top}\}$. When defining unification in $\mathcal{L}$, we assume that the set of concept names is partitioned into a set $\mathsf{N_v}$ of concept variables (which may be replaced by substitutions) and a set $\mathsf{N_c}$ of concept constants (which must not be replaced by substitutions). An *$\mathcal{L}$-substitution* σ is a mapping from $\mathsf{N_v}$ into the set of all $\mathcal{L}$-concept terms. This mapping is extended to concept terms in the usual way, that is, by replacing all occurrences of variables in the term by their σ-images. An $\mathcal{L}$-concept term is called *ground* if it contains no variables, and an $\mathcal{L}$-substitution σ is called *ground* if the concept terms $\sigma(X)$ are ground for all $X \in \mathsf{N_v}$.

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

Definition 3 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 if and only if it solves all the equations $C_i \equiv^? D_i$ in Γ, that is, if and only if $\sigma(C_i) \equiv \sigma(D_i)$ for $i = 1, \ldots, n$. In this case, Γ is called *$\mathcal{L}$-unifiable*.

We will often use the subsumption $C \sqsubseteq^? D$ as an abbreviation for the equation $C \sqcap D \equiv^? C$. Obviously, σ solves this equation if and only if $\sigma(C) \sqsubseteq \sigma(D)$.

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

We may without loss of generality restrict our attention to *flat $\mathcal{L}$-unification problems*, that is, $\mathcal{L}$-unification problems in which the right- and left-hand sides of the equations are flat $\mathcal{L}$-concept terms. Nonflat $\mathcal{L}$-concept terms can be flattened by introducing new variables as abbreviations for subterms (see [12]). Given a flat unification problem Γ, we denote by $\mathrm{At}(\Gamma)$ the set of all atoms of Γ, that is, the union of all sets of atoms of the concept terms occurring in Γ. By $\mathrm{Var}(\Gamma)$ we denote the variables that occur in Γ and by $\mathrm{NV}(\Gamma) := \mathrm{At}(\Gamma) \setminus \mathrm{Var}(\Gamma)$ the set of all *nonvariable atoms* of Γ.

Although arbitrary $\mathcal{L}$-substitutions σ are used in the definition of an $\mathcal{L}$-unifier, it is actually sufficient to consider ground $\mathcal{L}$-substitutions σ such that all $\mathcal{L}$-concept descriptions $\sigma(X)$ in the range of σ contain only concept and role names occurring in Γ. It is an easy consequence of well-known results from unification theory (see Baader and Snyder [14]) that an $\mathcal{L}$-unification problem Γ has an $\mathcal{L}$-unifier if and only if it has such a ground $\mathcal{L}$-unifier. Thus, for simplicity we will assume in the following that $\mathsf{N_R}$ is the set of role names occurring in Γ and that $\mathsf{N_c}$ is the set of concept constants occurring in Γ. Since we are only interested in the substitution of variables occurring in Γ, we will also assume that $\mathsf{N_v} = \mathrm{Var}(\Gamma)$.

3.1 Connection to other unification problems Unification was originally not introduced for DLs, but for equational theories (see [14]). In Sofronie-Stokkermans [26] and [12] it was shown that equivalence and unification in $\mathcal{EL}$ are the same as the word problem and unification, respectively, in the equational theory *bSLmO* of bounded (meet-)semilattices with monotone operators. The signature Σ_{bSLmO} of this equational theory consists of a binary function symbol $\wedge$, a constant symbol 1, and finitely many unary function symbols $f_1, \dots, f_n$. Terms can be built by using these symbols and additional variable symbols and free constant symbols. The signature Σ_{SLmO} is obtained from Σ_{bSLmO} by dropping the constant 1.

Definition 4 The equational theory of *bounded semilattices with monotone operators* is defined by the following identities:

$$\begin{aligned} bSLmO := \{&x \wedge (y \wedge z) = (x \wedge y) \wedge z,\ x \wedge y = y \wedge x,\ x \wedge x = x,\ x \wedge 1 = x\} \\ &\cup \{f_i(x \wedge y) \wedge f_i(y) = f_i(x \wedge y) \mid 1 \leq i \leq n\}. \end{aligned}$$

The equational theory *SLmO* of *semilattices with monotone operators* is obtained from the above definition by dropping the identity $x \wedge 1 = x$.

Any $\mathcal{EL}$-concept description C using only the roles $r_1, \dots, r_n$ can be translated into a term t_C over the signature Σ_{bSLmO} by replacing each concept constant A by a free constant a, each concept variable X by a variable x, $\top$ by 1, $\sqcap$ by $\wedge$, and $\exists r_i$ by f_i. For example, the $\mathcal{EL}$-concept description $C = A \sqcap \exists r_1.\top \sqcap \exists r_3.(X \sqcap B)$ is translated into $t_C = a \wedge f_1(1) \wedge f_3(x \wedge b)$. Conversely, any term t over the signature Σ_{bSLmO} can be translated back into an $\mathcal{EL}$-concept description C_t. The same holds for $\mathcal{EL}^{-\top}$-concept descriptions and terms over Σ_{SLmO}. As shown in [26], the word problem in the theory $SLmO$ is the same as the equivalence problem for $\mathcal{EL}$-concept descriptions. Again, a similar result holds for $\mathcal{EL}^{-\top}$.

Lemma 5 *Let C, D be $\mathcal{EL}$-concept descriptions using only the roles $r_1, \ldots, r_n$. Then $C \equiv D$ holds if and only if $t_C =_{bSLmO} t_D$. If C, D are $\mathcal{EL}^{-\top}$-concept descriptions, then this is also equivalent to $t_C =_{SLmO} t_D$.*

As an immediate consequence of this lemma, every $\mathcal{EL}$- or $\mathcal{EL}^{-\top}$-unification problem can be translated into a unification problem modulo the corresponding equational theory that, apart from the translation between concept descriptions and terms, has the same unifiers.

Thus, previous results for unification in $\mathcal{EL}$ imply that unification modulo *bSLmO* is NP-complete, even if a certain restricted form of ground equations is added to the equational theory (see [6], [12] for details). Correspondingly, the results we present in this article show that unification modulo *SLmO* is PSpace-complete.

There is also a strong connection between DLs and modal logics and, therefore, between unification in these formalisms. In the basic multimodal logic $\mathsf{K_m}$, formulae are built from a set of propositional variables using the propositional connectives $\top$, $\bot$, $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\leftrightarrow$, and two unary connectives $\Box_{r_i}$ and $\Diamond_{r_i}$ for each relation symbol r_i from a fixed set $\{r_1, \ldots, r_n\}$ (see Blackburn, van Benthem, and Wolter [16]).

These formulae are interpreted over so-called *Kripke models*, which consist of a set of worlds that are connected by binary relations corresponding to the symbols $r_1, \ldots, r_n$. The validity of a formula φ in a world w of such a model is defined inductively on the structure of φ, where the validity of the atomic propositions in a world is fixed by the given model. The definition of validity is extended to the propositional connectives in the usual way. A formula of the form $\Box_{r_i}\varphi$ is said to be valid in a given world w if φ is valid in all worlds connected to w via the binary relation associated to r_i. Dually, $\Diamond_{r_i}\varphi$ is valid in w if φ is valid in at least one world connected to w by r_i.

It was first observed in Schild [25] that $\mathsf{K_m}$ is a notational variant of the DL $\mathcal{ALC}$. There is a bijective translation of formulae of $\mathsf{K_m}$ into $\mathcal{ALC}$-concept descriptions, and Kripke models can be characterized as DL interpretations. In this setting, the DL $\mathcal{EL}$ corresponds to the syntactic fragment of $\mathsf{K_m}$ that is restricted to the connectives $\top$, $\wedge$, and $\Diamond_{r_i}$. Every $\mathcal{EL}$-concept description C can be translated into a modal formula φ_C by replacing every concept name by a propositional variable, $\sqcap$ by $\wedge$, and $\exists r_i$ by $\Diamond_{r_i}$. On the other hand, every modal formula φ in this fragment of $\mathsf{K_m}$ can be translated back into an $\mathcal{EL}$-concept description C_φ by applying the inverse transformation. In the same way, $\mathcal{EL}^{-\top}$ corresponds to the $\wedge$-$\Diamond_{r_i}$-fragment of $\mathsf{K_m}$.

It is an easy consequence of the results of [25] that two $\mathcal{EL}$-concept descriptions C and D are equivalent if their translations φ_C and φ_D are valid in the same Kripke models. In $\mathsf{K_m}$, this is usually expressed as the validity of $\varphi_C \leftrightarrow \varphi_D$ in every Kripke model. Note, however, that in the sub-Boolean fragments of $\mathsf{K_m}$ corresponding to $\mathcal{EL}$ and $\mathcal{EL}^{-\top}$ this bi-implication cannot be expressed. In particular, in $\mathcal{EL}$ there are no constructors directly corresponding to negation, disjunction, implication, or bi-implication.

Traditionally, unification in modal logics is the problem of finding, for a given modal formula φ, a substitution σ of the propositional variables by modal formulae such that $\sigma(\varphi)$ becomes valid in all Kripke models (see Baader and Ghilardi [9], [16]). A famous open problem is the decidability of unification in K, the unimodal version of $\mathsf{K_m}$ with only one relation symbol. Unification in several extensions of

K has been shown to be undecidable in Wolter and Zakharyaschev [28]. For an overview of known results about unification in modal logics, see [9].

Following the translations between concepts and modal formulae described above, the unifiability of a set $\{C_1 \equiv^? D_1, \ldots, C_m \equiv^? D_m\}$ of equations over concept descriptions in some DL is equivalent to the unifiability of $\varphi_{C_1} \leftrightarrow \varphi_{D_1} \wedge \cdots \wedge \varphi_{C_m} \leftrightarrow \varphi_{D_m}$ in the corresponding modal logic (see [9]). Note that some of the propositional variables in the translated formula have to be viewed as *constants*, which are not allowed to be replaced by substitutions. Again, for $\mathcal{EL}$ and $\mathcal{EL}^{-\top}$ this results in a formula which in general cannot be expressed in the fragments of $\mathsf{K_m}$ mentioned above. To the best of our knowledge, unification in such sub-Boolean modal logics has not been considered in the modal logic literature.

On the other hand, a modal formula φ is unifiable if $\{C_\varphi \equiv^? \top\}$ is unifiable in the corresponding DL. Consider now the $\top$-$\wedge$-$\Diamond_{r_i}$-fragment of $\mathsf{K_m}$, which corresponds to $\mathcal{EL}$. According to Lemma 1, we know that φ is unifiable if and only if it is a conjunction of variables or $\top$: if φ is of this form, then it can be made valid by substituting every variable by $\top$; otherwise, for any substitution σ the concept description $\sigma(C_\varphi)$ must contain at least one atom, which cannot be contained in $\top$. Thus, unification in the $\top$-$\wedge$-$\Diamond_{r_i}$-fragment of $\mathsf{K_m}$ is trivial. In the $\wedge$-$\Diamond_{r_i}$-fragment of $\mathsf{K_m}$, unification is even more absurd since $\sigma(C_\varphi)$ must always contain at least one atom, regardless of the form of φ. This means that this modal logic does not have unifiable formulae.

This shows that in the above fragments of $\mathsf{K_m}$ it does not make sense to consider unification in the modal logic sense. If we consider instead the *equational* variant of modal unification that corresponds to Definition 3, then unification in the $\top$-$\wedge$-$\Diamond_{r_i}$-fragment of $\mathsf{K_m}$ is NP-complete (see [12]), and it is PSpace-complete in the $\wedge$-$\Diamond_{r_i}$-fragment, as we show in this article.

3.2 Locality of $\mathcal{EL}$-unification The NP-algorithm for unification in $\mathcal{EL}$ introduced in [10] is based on the fact that every unifiable $\mathcal{EL}$-unification problem Γ has a so-called *local $\mathcal{EL}$-unifier*, which we define in the following.

Given an $\mathcal{EL}$-unification problem Γ, an *assignment* is a function S mapping each variable $X \in \mathrm{Var}(\Gamma)$ to a set $S(X) \subseteq \mathrm{NV}(\Gamma)$. Such an assignment S induces the relation $>_S$, which is the transitive closure of the *depends on* relation

$$\big\{(X,Y) \in \mathrm{Var}(\Gamma) \times \mathrm{Var}(\Gamma) \;\big|\; Y \text{ occurs in an element of } S(X)\big\}.$$

We call the assignment S *acyclic* if $>_S$ is irreflexive (and thus a strict partial order). Any acyclic assignment S induces a unique substitution γ^S, which can be defined by induction along $>_S$.

- If X is a minimal element of $\mathrm{Var}(\Gamma)$ with respect to $>_S$, then $\gamma^S(X)$ is the conjunction of the elements of $S(X)$, where the empty conjunction is $\top$.
- Assume that $\gamma^S(Y)$ is defined for all $Y <_S X$. If $S(X) = \{D_1, \ldots, D_n\}$, then $\gamma^S(X) := \gamma^S(D_1) \sqcap \cdots \sqcap \gamma^S(D_n)$.

A unifier γ of Γ is called a *local $\mathcal{EL}$-unifier* of Γ if it is of the above form, that is, if there is an acyclic assignment S such that $\gamma = \gamma^S$ (see Example 6).

In [10] it was shown that every unifiable $\mathcal{EL}$-unification problem Γ has a local $\mathcal{EL}$-unifier. This gives rise to a simple NP-algorithm for deciding unification in $\mathcal{EL}$: first, guess an assignment S in polynomial time, and then check whether S is acyclic and whether the induced substitution γ^S solves the unification problem. The former property can clearly be tested in polynomial time in the size of S.

However, the substitution γ^S may be exponential in the size of S, which is basically due to the fact that subterms are copied if variables occur several times in a set $S(X)$. In [12], this problem was solved by representing γ^S by the acyclic TBox

$$\mathcal{T}^S := \{X \equiv D_1 \sqcap \cdots \sqcap D_n \mid X \in \mathrm{Var}(\Gamma),\ S(X) = \{D_1, \ldots, D_n\}\},$$

which has the same size as S. It is easy to see that $\gamma^S(X) = X^{\mathcal{T}^S}$; that is, the substitution of a variable $X \in \mathrm{Var}(\Gamma)$ under γ^S is simply the unfolding of X with respect to $\mathcal{T}^S$. Thus, to check whether γ^S solves an equation $C \equiv^? D$ from Γ, we have to decide whether $C^{\mathcal{T}^S}$ is equivalent to $D^{\mathcal{T}^S}$. It has been shown that this problem can be solved in polynomial time for any acyclic TBox in $\mathcal{EL}$ (see [1]). This shows that the second test, that is, whether γ^S solves Γ, can also be done in polynomial time, which yields an NP-algorithm for deciding unification in $\mathcal{EL}$.

3.3 Why this does not work for $\mathcal{EL}^{-\top}$ The $\mathcal{EL}$-unifiers returned by the 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; that is, it may return no substitution although the input problem is $\mathcal{EL}^{-\top}$-unifiable.

Example 6 Consider the flat $\mathcal{EL}$-unification problem Γ that contains the three equations

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

The substitutions

$$\sigma_0 := \{X \mapsto A,\ Y \mapsto \top,\ Z \mapsto \top\},$$
$$\sigma_1 := \{X \mapsto A,\ Y \mapsto \top,\ Z \mapsto \exists r.A\}$$

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 having $\exists r.X$ in $S(X)$ would make the assignment S acyclic). 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 Γ can also be seen as an $\mathcal{EL}^{-\top}$-unification problem, but 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 a local $\mathcal{EL}$-unifier. 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 modified definition of locality for $\mathcal{EL}^{-\top}$.

Definition 7 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 $D \in \mathrm{NV}(\Gamma)$.

Note that this definition becomes equivalent to locality in $\mathcal{EL}$ if the second option is left out. Since there are at most polynomially many assignments S for a given $\mathcal{EL}$-unification problem, there can only be polynomially many local $\mathcal{EL}$-unifiers. In $\mathcal{EL}^{-\top}$, however, it is possible that there exist infinitely many local unifiers, as the next example illustrates.

Example 8 Consider the unification problem Γ from Example 6 and the following $\mathcal{EL}^{-\top}$-substitutions γ_n:

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

where $\exists r^n.A$ is short for the concept term $\exists r. \cdots \exists r.A$ with n nested existential restrictions.

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 nonvariable atoms of Γ. Thus, Γ has infinitely many local $\mathcal{EL}^{-\top}$-unifiers.

These unifiers are even incomparable with respect to the subsumption order on unifiers; that is, for no two $n, m \in \mathbb{N}$ with $n \neq m$ does it hold that $\gamma_n(W) \sqsubseteq \gamma_m(W)$ for all variables W. This is the case since the particles $\gamma_n(Z) = \exists r^{n+1}.A$ are incomparable.

In Section 4, we consider two problems: how to decide unifiability in $\mathcal{EL}^{-\top}$ and how to actually compute an $\mathcal{EL}^{-\top}$-unifier. It will turn out that, similarly to $\mathcal{EL}$, it actually suffices to search for local $\mathcal{EL}^{-\top}$-unifiers. We will present an algorithm that decides $\mathcal{EL}^{-\top}$-unification in PSpace and can be used to compute local $\mathcal{EL}^{-\top}$-unifiers of at most exponential size. The main idea underlying the algorithm 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.

In Section 5, we will then provide corresponding hardness results. First, we show that deciding $\mathcal{EL}^{-\top}$-unification is PSpace-hard and then present a series of $\mathcal{EL}^{-\top}$-unification problems whose local unifiers are always of exponential size.

4 Our $\mathcal{EL}^{-\top}$-Unification Algorithm

For the remainder of this section, let Γ be a flat $\mathcal{EL}^{-\top}$-unification problem. We assume without loss of generality that Γ is a set of flat subsumptions of the form $C_1 \sqcap \cdots \sqcap C_n \sqsubseteq^? D$, where $C_1, \ldots, C_n, D$ are flat atoms. Every equation $C_1 \sqcap \cdots \sqcap C_n \equiv^? D_1 \sqcap \cdots \sqcap D_m$ in Γ can equivalently be expressed by $n + m$ such subsumptions.

4.1 Step 1: Guessing an $\mathcal{EL}$-unifier The first step of the algorithm is to guess an $\mathcal{EL}$-unifier that is the starting point for constructing an $\mathcal{EL}^{-\top}$-unifier. Recall that if S is an acyclic assignment, then $D \in S(X)$ implies that the subsumption $\gamma^S(X) \sqsubseteq \gamma^S(D)$ holds for the substitution γ^S induced by S. Thus, guessing the sets $S(X)$ can be seen as guessing subsumptions between variables and nonvariable atoms of Γ. In addition to guessing these subsumptions, our $\mathcal{EL}^{-\top}$-unification algorithm also guesses subsumptions between all other atoms of Γ. To be more precise, it guesses a mapping $\tau \colon \mathrm{At}(\Gamma)^2 \to \{0, 1\}$, which specifies which subsumptions between atoms of Γ should hold for the $\mathcal{EL}^{-\top}$-unifier that it tries to generate: if $\tau(D_1, D_2) = 1$ for $D_1, D_2 \in \mathrm{At}(\Gamma)$, then this means that the search for a unifier is restricted (in this branch of the search tree) to substitutions γ satisfying $\gamma(D_1) \sqsubseteq \gamma(D_2)$. Any such mapping τ also yields an assignment

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

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

Definition 9 The mapping $\tau \colon \mathrm{At}(\Gamma)^2 \to \{0, 1\}$ is called a *subsumption mapping for* Γ if it satisfies the following three conditions.

1. It respects the properties of subsumption in $\mathcal{EL}$.
   (a) $\tau(D, D) = 1$ for each $D \in \mathrm{At}(\Gamma)$.
   (b) $\tau(A_1, A_2) = 0$ for distinct concept constants $A_1, A_2 \in \mathrm{At}(\Gamma)$.
   (c) $\tau(\exists r.C_1, \exists s.C_2) = 0$ for distinct $r, s \in \mathsf{N_R}$ with $\exists r.C_1, \exists s.C_2 \in \mathrm{At}(\Gamma)$.
   (d) $\tau(A, \exists r.C) = \tau(\exists r.C, A) = 0$ for each constant $A \in \mathrm{At}(\Gamma)$, role name r, and variable or constant C with $\exists r.C \in \mathrm{At}(\Gamma)$.
   (e) If $\exists r.C_1, \exists r.C_2 \in \mathrm{At}(\Gamma)$, then $\tau(\exists r.C_1, \exists r.C_2) = \tau(C_1, C_2)$.
   (f) For all atoms $D_1, D_2, D_3 \in \mathrm{At}(\Gamma)$, if $\tau(D_1, D_2) = \tau(D_2, D_3) = 1$, then $\tau(D_1, D_3) = 1$.
2. It induces an $\mathcal{EL}$-substitution; that is, 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; that is, it satisfies the following conditions for each subsumption $C_1 \sqcap \cdots \sqcap C_n \sqsubseteq^? D$ in Γ.
   (a) If D is a nonvariable 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 nonvariable atom $C \in \mathrm{NV}(\Gamma)$, then there is at least one C_i with $\tau(C_i, C) = 1$.

Though it is not necessary for the proof of the correctness of our $\mathcal{EL}^{-\top}$-unification algorithm, it can be shown that the substitution γ^τ induced by a subsumption mapping τ for Γ is indeed an $\mathcal{EL}$-unifier of Γ.[2] It should be noted that γ^τ need not be an $\mathcal{EL}^{-\top}$-unifier of Γ. In addition, γ^τ need not agree with τ on every subsumption between atoms of Γ. The reason for this is that τ specifies subsumptions which should hold in the $\mathcal{EL}^{-\top}$-unifier of Γ to be constructed. To turn γ^τ into such an $\mathcal{EL}^{-\top}$-unifier, we may have to add certain particles, and these additions may invalidate subsumptions that hold for γ^τ. However, we will ensure that no subsumption claimed by τ is invalidated. It is clear that guessing τ and checking the above conditions can be done in NP.

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

$$\Delta_\Gamma := \{C_1 \sqcap \cdots \sqcap C_n \sqsubseteq^? X \in \Gamma \mid X \text{ is a variable of } \Gamma\},$$

$$\Delta_\tau := \left\{C \sqsubseteq^? X \;\middle|\; X \text{ is a variable and } C \text{ is an atom of } \Gamma \text{ with } \tau(C, X) = 1\right\}.$$

Before we can formulate the connection between the $\mathcal{EL}^{-\top}$-unifiability of Γ and of $\Delta_{\Gamma,\tau}$, we need to introduce some notation and show an auxiliary result. For an arbitrary $\mathcal{EL}^{-\top}$-substitution σ, we will in the following write $S^\tau \leq S^\sigma$ if

$$S^\tau(X) \subseteq S^\sigma(X) := \left\{D \in \mathrm{NV}(\Gamma) \;\middle|\; \sigma(X) \sqsubseteq \sigma(D)\right\}$$

holds for every variable X. We now show that under some conditions on the $\mathcal{EL}^{-\top}$-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 10 *Let τ be a subsumption mapping for Γ, and let σ be 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 statements hold.*

1. *If D is ground, then $\tau(C, D) = 1$ implies $\sigma(C) \sqsubseteq \sigma(D)$.*
2. *If $D = \exists r.Y$ for a variable 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, regardless of the form of D. Otherwise, we consider the structure of D.

1. If D is a constant, then conditions 1(b) and 1(d) of Definition 9 yield $C = D$, and the subsumption is clearly satisfied.

   If D is of the form $\exists r.D'$ for a constant D', then by conditions 1(c)–1(e) of Definition 9, 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. Since D' is a constant, we know that 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.
2. If $D = \exists r.Y$ for a variable Y, then again C must be of the form $\exists r.C'$ and $\tau(C', Y) = 1$. But then $C' \sqsubseteq Y$ is a subsumption in Δ_τ, and we have $\sigma(C') \sqsubseteq \sigma(Y)$. Thus, $\sigma(C) \sqsubseteq \sigma(D)$, by assumption. □

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

Lemma 11 *The following statements are equivalent.*

- *Γ is $\mathcal{EL}^{-\top}$-unifiable.*
- *There is a subsumption mapping τ 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, then we can define τ as $\tau(D_1, D_2) = 1$ if and only if $\sigma(D_1) \sqsubseteq \sigma(D_2)$ 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 subsumption mapping.

- Conditions 1(a)–1(f) of Definition 9 are obviously satisfied by the subsumption relation.

- Conditions 3(a) and 3(b) of Definition 9 are satisfied, since σ is a unifier of Γ and Lemma 1 holds.
- To show that condition 2 of Definition 9 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 $\sigma(X_1) \sqsubseteq \sigma(\exists r_1. \cdots \exists r_{n-1}.X_1) = \exists r_1. \cdots \exists r_{n-1}.\sigma(X_1)$, which is impossible. Thus, condition 2 of Definition 9 is also satisfied.

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

By condition 3(a) of Definition 9, 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 10 and get $\sigma(C_i) \sqsubseteq \sigma(D)$. Thus, σ satisfies all subsumptions of Γ. □

For the problem of actually computing local $\mathcal{EL}^{-\top}$-unifiers of Γ, we also need to consider the locality of the unifiers of $\Delta_{\Gamma,\tau}$. Fortunately, it can easily be seen from the second part of the proof of Lemma 11 that any local $\mathcal{EL}^{-\top}$-unifier σ of $\Delta_{\Gamma,\tau}$ with $S^\tau \le S^\sigma$ is also a local $\mathcal{EL}^{-\top}$-unifier of Γ since every nonvariable atom in $\Delta_{\Gamma,\tau}$ must also occur in Γ.

Lemma 12 *Let τ be a subsumption mapping for Γ, and let σ be a local $\mathcal{EL}^{-\top}$-unifier of $\Delta_{\Gamma,\tau}$ with $S^\tau \le S^\sigma$. Then σ is also a local $\mathcal{EL}^{-\top}$-unifier of Γ.*

The converse of this lemma does not hold. However, our aim is not to construct local $\mathcal{EL}^{-\top}$-unifiers of $\Delta_{\Gamma,\tau}$ from local $\mathcal{EL}^{-\top}$-unifiers of Γ, but only to do the other way around. Thus, in the following we need to solve the problem of computing local $\mathcal{EL}^{-\top}$-unifiers σ of $\Delta_{\Gamma,\tau}$ that satisfy the additional condition $S^\tau \le S^\sigma$. For the following steps, we fix a subsumption mapping τ for Γ.

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

Admissible words (particles) are determined by solutions of a system of linear language inclusions.

Definition 13 Let $\{X_1, \ldots, X_n\}$ be a finite set of *indeterminates*. A *linear language inclusion* over these indeterminates is an expression of the form

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

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

We will often use the abbreviations $\theta(L_i X_i) := L_i\theta(X_i)$ and $\theta(L_0) := L_0$.

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

$$X_A \subseteq f_A(C_1) \cup \cdots \cup f_A(C_n), \quad \text{where}$$

$$f_A(C) := \begin{cases} \{r\} f_A(C') & \text{if } C = \exists r.C', \\ Y_A & \text{if } C = Y \in \mathsf{N_v}, \\ \{\varepsilon\} & \text{if } C = A, \\ \emptyset & \text{if } C \in \mathsf{N_c} \setminus \{A\}. \end{cases}$$

All the inclusions $I_A(\mathfrak{s})$ for $\mathfrak{s} \in \Delta_{\Gamma,\tau}$ are linear language inclusions 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 is translated into the two inclusions

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

if A and B are the only constants that occur in Γ.

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

Theorem 14 *We have that $\Delta_{\Gamma,\tau}$ has an $\mathcal{EL}^{-\top}$-unifier σ with $S^\tau \leq S^\sigma$ if and only if $\mathcal{I}_{\Gamma,\tau}$ has a finite, admissible solution.*

We prove the two directions of this equivalence separately.

Lemma 15 *If $\Delta_{\Gamma,\tau}$ has an $\mathcal{EL}^{-\top}$-unifier σ with $S^\tau \leq S^\sigma$, then $\mathcal{I}_{\Gamma,\tau}$ 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

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

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 $C_1 \sqcap \cdots \sqcap C_n \sqsubseteq^? X$ in $\Delta_{\Gamma,\tau}$ and a word $w \in \theta(X_A)$. By Lemma 2, we have $\sigma(X) \sqsubseteq \exists w.A$, and thus, Lemma 1 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.

- 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, $w \in f_A(C_i) = \{\varepsilon\} = \theta(f_A(C_i))$.
- If $C_i = Y$ is a variable, then $w \in \theta(Y_A) = \theta(f_A(C_i))$ by definition.
- 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 $w' \in \mathsf{N_R^*}$ 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 $I_A(\mathfrak{s})$ since w was an arbitrary element of $\theta(X_A)$. 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 particle $\exists w.A \in \mathrm{Part}(\sigma(X))$ for some constant A and word w, and thus, $\theta(X_A)$ is nonempty. This shows that θ is also admissible. □

It remains to show the other direction of Theorem 14, that is, how to construct an $\mathcal{EL}^{-\top}$-unifier of $\Delta_{\Gamma,\tau}$ from a solution of $\mathcal{I}_{\Gamma,\tau}$. Recall that we want to compute *local* $\mathcal{EL}^{-\top}$-unifiers of $\Delta_{\Gamma,\tau}$. For this reason, we will prove a stronger result, which uses a corresponding notion of locality for solutions of $\mathcal{I}_{\Gamma,\tau}$.

Definition 16 Let $\mathcal{I}$ be a finite set of linear language inclusions over the indeterminates $X_1, \ldots, X_n$. A solution θ of $\mathcal{I}$ is called *local* if for all $i \in \{1, \ldots, n\}$ and all words $w \in \theta(X_i) \setminus \{\varepsilon\}$ there is an inclusion $Y \subseteq L_0 \cup L_1 X_1 \cup \cdots \cup L_n X_n$ in $\mathcal{I}$ such that $w \in L_0$ or $w \in (L_j \setminus \{\varepsilon\})\theta(X_j)$ for some $j \in \{1, \ldots, n\}$.

Note that in a solution θ of $\mathcal{I}$ any word w that violates this condition can safely be removed from θ. Thus, whenever $\mathcal{I}$ has a finite, admissible solution, it also has a local one.

Lemma 17 *If $\mathcal{I}_{\Gamma,\tau}$ has a finite, local, admissible solution, then $\Delta_{\Gamma,\tau}$ has a local $\mathcal{EL}^{-\top}$-unifier σ with $S^\tau \leq S^\sigma$.*

Proof Let θ be a finite, local, admissible solution of $\mathcal{I}_{\Gamma,\tau}$. We now define an $\mathcal{EL}^{-\top}$-substitution σ by induction on the dependency order $> \, := \, >_{S^\tau}$ induced by S^τ (see Sections 3.2 and 4.1). Let X be a variable, and assume that $\sigma(Y)$ has already been defined for all variables Y with $X > Y$. We set

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

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 local $\mathcal{EL}^{-\top}$-unifier of $\Delta_{\Gamma,\tau}$.

We first show that σ satisfies all subsumptions in $\Delta_{\Gamma,\tau}$ by using induction on the strict partial order $>$ on the variables. Let X be a variable, and let σ satisfy all subsumptions $D_1 \sqcap \cdots \sqcap D_m \sqsubseteq^? Y$ in $\Delta_{\Gamma,\tau}$ for all variables Y with $X > Y$. We consider an arbitrary subsumption $\mathfrak{s}$ of the form $C_1 \sqcap \cdots \sqcap C_n \sqsubseteq^? X$ in $\Delta_{\Gamma,\tau}$. This of course includes the subsumptions from Δ_τ, but in that case we always have $n = 1$. We have to show that every top-level atom of $\sigma(X)$ subsumes some $\sigma(C_i)$. Recall that 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 9 implies that there is a C_i with $\tau(C_i, D) = 1$. But also in the case in which $\mathfrak{s} \in \Delta_\tau$, we know that $\mathfrak{s}$ is of the form $C_1 \sqsubseteq X$ and $\tau(C_1, X) = 1$ holds. By condition 1(f), we deduce that $\tau(C_i, D) = 1$ holds for $i = 1$. By the definition of the order $>$, the nonvariable 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 with respect to $>$ on the right-hand side. Thus, we can apply Lemma 10 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 $A \in \mathsf{N_c}$ 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.

1. If C_i is a concept constant, then it must be A since otherwise we would have $w \in \theta(\emptyset) = \emptyset$. Thus, we have $w \in \theta(\{\varepsilon\}) = \{\varepsilon\}$, that is, $w = \varepsilon$, which implies $\sigma(C_i) = A = \exists w.A$.
2. In the case in which $C_i = Y$ 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 $\sigma(C_i) \sqsubseteq \exists w.A$.
3. In the remaining case in which $C_i = \exists r.C'$ 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 1 and 2 to C' and w' yields the subsumption $\sigma(C') \sqsubseteq \exists w'.A$, which implies $\sigma(C_i) = \exists r.\sigma(C') \sqsubseteq \exists r.\exists w'.A = \exists w.A$.

Finally, to show that σ is a *local* $\mathcal{EL}^{-\top}$-unifier, we again consider all top-level atoms of $\sigma(X)$ for each variable X. For the top-level atoms of the form $\sigma(D)$ for $D \in S^\tau(X)$, we immediately have $D \in \mathrm{NV}(\Gamma)$ since $S^\tau(X) \subseteq \mathrm{NV}(\Gamma)$. Now consider a top-level particle $\exists w.A$ of $\sigma(X)$. If $w = \varepsilon$, then A is a nonvariable atom of Γ since we assumed that all elements of $\mathsf{N_c}$ occur in Γ. Otherwise, $w \in \theta(X_A) \setminus \{\varepsilon\}$, and by the locality of θ, there are subsumption of the form $C_1 \sqcap \cdots \sqcap C_n \sqsubseteq^? X$ in $\Delta_{\Gamma,\tau}$ and an index $i \in \{1, \ldots, n\}$ such that $w \in \theta(f_A(C_i))$ and C_i is neither a variable nor a constant.

Thus, C_i is of the form $\exists r.C'$, where C' is either the constant A or a variable. 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, we have $w = rw'$ for some $w' \in \theta(C'_A)$. By the definition of σ, this implies $\sigma(C') \sqsubseteq \exists w'.A$, which yields $\sigma(C_i) \sqsubseteq \exists w.A$. By Lemma 2, $\exists w.A$ is a particle of $\sigma(C_i)$. Since $C_i \in \mathrm{NV}(\Gamma)$, the particle $\exists w.A$ fulfills the condition for the locality of σ. □

This concludes the proof of Theorem 14, which shows that the solvability of $\Delta_{\Gamma,\tau}$ with a unifier σ that satisfies $S^\tau \leq S^\sigma$ can be reduced to the solvability of $\mathcal{I}_{\Gamma,\tau}$ with a finite, admissible solution θ. However, we are also interested in the size of the computed unifier σ in terms of the size of the solution θ. We will denote the *size* of something by $\|\cdot\|$, which is basically the number of symbols it takes to write it down, where we assume that every role name $r \in \mathsf{N_R}$ is of size 1 and auxiliary symbols like (, }, and $\exists$ are of size 0.

For a solution θ of $\mathcal{I}_{\Gamma,\tau}$, we define

$$\|\theta\| := \sum_{A \in \mathsf{N_c}} \sum_{X \in \mathsf{N_v}} \sum_{w \in \theta(X_A)} \big(|w| + 1\big),$$

where $|w|$ is the length of a word $w \in \mathsf{N_R^*}$. Similarly, we measure the size $\|C\|$ of an $\mathcal{EL}^{-\top}$-concept term C by the number of distinct occurrences of concept and role names in C, and the size of a set of concept terms is the sum of the sizes of its elements. The size of $\Gamma = \{C_1 \sqsubseteq^? D_1, \ldots, C_n \sqsubseteq^? D_n\}$ is the sum of the sizes of $C_1, D_1, \ldots, C_n, D_n$. Finally, the size $\|\sigma\|$ of a substitution σ is the sum of the sizes of $\sigma(X)$ for all $X \in \mathsf{N_v}$.

To analyze the size of the unifier constructed in Lemma 17, we need another auxiliary definition. For a variable $X \in \mathrm{Var}(\Gamma)$, we consider all sequences

$X = X_n > \cdots > X_1$ where X_1 is minimal with respect to $> := >_{S^\tau}$. The length of such a sequence is the number of variables it contains, that is, n. The *height* of X is defined as the maximal length of all these sequences. This means that the height of a minimal variable is 1, and the height is bounded by $|\operatorname{Var}(\Gamma)|$ since S^τ is acyclic.

Lemma 18 *Let θ be a finite, local, admissible solution of $\mathcal{I}_{\Gamma,\tau}$, and let σ be the local $\mathcal{EL}^{-\top}$-unifier constructed from θ as described in Lemma 17. Then for every $X \in \mathsf{N_v}$ of height n, we have*

$$\|\sigma(X)\| \leq n\|\Gamma\|^n + \|\theta\|\Big(\sum_{i=0}^{n-1}\|\Gamma\|^i\Big).$$

Proof We prove the claim by induction on $>$. Let X be a minimal variable with respect to $>$. Since all nonvariable atoms in $S^\tau(X)$ are ground and occur in Γ, the size of $\sigma(X)$ is bounded by $\|S^\tau(X)\| + \|\theta\| \leq \|\Gamma\| + \|\theta\|$.

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

$$n\|\Gamma\|^{n-1} + \|\theta\|\Big(\sum_{i=0}^{n-2}\|\Gamma\|^i\Big),$$

where the additional $\|\Gamma\|^{n-1}$ is a very loose bound for the additional role name in the case of $D = \exists r.Y$, assuming without loss of generality that Γ is nonempty. Since there are at most $\|\Gamma\|$ elements in $S^\tau(X)$, the size of $\sigma(X)$ is thus bounded by

$$\|\Gamma\|\Big(n\|\Gamma\|^{n-1} + \|\theta\|\Big(\sum_{i=0}^{n-2}\|\Gamma\|^i\Big)\Big) + \|\theta\| = n\|\Gamma\|^n + \|\theta\|\Big(\sum_{i=0}^{n-1}\|\Gamma\|^i\Big),$$

which concludes the proof. □

Since the height of any variable is bounded by $|\Gamma|$, this lemma bounds the overall size of σ by

$$\|\Gamma\|^{\|\Gamma\|+1} + \|\theta\|\Big(\sum_{i=0}^{\|\Gamma\|-1}\|\Gamma\|^i\Big) \leq \big(\|\theta\| + 1\big)\|\Gamma\|^{\|\Gamma\|+1}.$$

The goal of the next step is to construct a solution θ of size exponential in the size of Γ, which then yields an exponential upper bound on the size of the constructed unifier.

4.4 Step 4: Constructing local solutions In this step, we show how to test whether the system $\mathcal{I}_{\Gamma,\tau}$ of linear language inclusions has a finite, admissible solution or not, and construct a local one if it does. The main idea is to consider the greatest solution of $\mathcal{I}_{\Gamma,\tau}$.

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

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

Lemma 19 *Let X be an indeterminate in $\mathcal{I}$, and let θ^* be 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 at most $|w|$: for all indeterminates Y occurring in $\mathcal{I}$ we define

$$\theta(Y) := \big\{u \in \theta^*(Y) \;\big|\; |u| \le |w|\big\}.$$

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

Lemma 20 *There is a finite, admissible solution of $\mathcal{I}_{\Gamma,\tau}$ if and only if 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 19) for each $X \in \mathrm{Var}(\Gamma)$ there is a constant $A(X)$ and a finite solution θ_X of $\mathcal{I}_{\Gamma,\tau}$ such that $\theta_X(X_{A(X)}) \neq \emptyset$. The union of these solutions θ_X for $X \in \mathrm{Var}(\Gamma)$ is the desired finite, admissible solution. □

Given this lemma, it remains to show how we can test the admissibility of the maximal solution θ^* of $\mathcal{I}_{\Gamma,\tau}$. For this purpose, it is obviously sufficient to be able to test, for each indeterminate X_A in $\mathcal{I}_{\Gamma,\tau}$, whether $\theta^*(X_A)$ is empty or not. We will do this by representing the languages $\theta^*(X_A)$ using *alternating finite automata with ε-transitions (ε-AFAs)*, which are a special case of two-way alternating finite automata, and testing these automata for emptiness. As shown in Jiang and Ravikumar [21], the emptiness problem for two-way alternating finite automata (and thus also for ε-AFAs) is in PSpace.

Alternating finite automata can make two kinds of transitions: 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 21 An *alternating finite automaton with ε-transitions (ε-AFA)* is a tuple $\mathcal{A} = (Q_\exists, Q_\forall, \Sigma, q_0, \delta, F)$ consisting 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* $\delta\colon Q \times (\Sigma \cup \{\varepsilon\}) \to \mathcal{P}(Q)$, 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 *transition relation* $\vdash_{\mathcal{A}}$ between configurations: $(q, w) \vdash_{\mathcal{A}} (q', w')$ if and only if 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 $w \neq \varepsilon$, that is, there is still a part of the input word left to read.

A *run* of $\mathcal{A}$ is a finite, directed, 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* if and only if 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 $\mathcal{A}$ if and only if there is a successful run of $\mathcal{A}$ with root label (q_0, w). The *language recognized by* $\mathcal{A}$ is denoted by $L(\mathcal{A})$ and contains exactly the words 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 a set $\mathcal{J}$ of linear language inclusions. The automaton checks whether the word w is an element of $\theta^*(X)$ by using the following ideas. Starting from the indeterminate X, the automaton splits into several copies that check the restrictions imposed by all the inclusions of the form $X \subseteq L_0 \cup L_1 X_1 \cup \cdots \cup L_n X_n$. Each of these copies nondeterministically guesses which of the sets $L_0, L_1\theta^*(X_1), \ldots, L_n\theta^*(X_n)$ contains w. When a copy guesses w to be in $\theta^*(X_i)$, this corresponds to an ε-transition and is only possible if L_i contains ε. We will describe below how we use counters to detect if a sequence of such ε-transitions visits the same variable twice and why this should not prevent the automaton from accepting a word.

In the following, let $\mathcal{J}$ be a finite set of linear language inclusions. We denote by $\mathrm{Ind}(\mathcal{J})$ the set of all indeterminates occurring in $\mathcal{J}$.

Definition 22 Let $X \in \mathrm{Ind}(\mathcal{J})$. The ε-AFA $\mathcal{A}_X = (Q_\exists, Q_\forall, \Sigma, q_0, \delta, F)$ is defined as follows:

- $Q_\exists := (\mathcal{J} \times \{0, \ldots, |\mathrm{Ind}(\mathcal{J})|\}) \cup \{f_0\}$,
- $Q_\forall := (\mathrm{Ind}(\mathcal{J}) \times \{0, \ldots, |\mathrm{Ind}(\mathcal{J})|\}) \cup \{f_1\}$,
- $q_0 := (X, 0)$,
- $F := \{f_0\}$,
- $\delta(f_i, \alpha) := \emptyset$ for every $i \in \{0, 1\}$ and $\alpha \in \Sigma \cup \{\varepsilon\}$,
- $\delta((Y, \lambda), \varepsilon) := \{(\mathrm{i}, \lambda) \mid \mathrm{i}\colon Y \subseteq \cdots \in \mathcal{J}\}$ and $\delta((Y, \lambda), \alpha) := \emptyset$ for all $Y \in \mathrm{Ind}(\mathcal{J})$, $\lambda \in \{0, \ldots, |\mathrm{Ind}(\mathcal{J})|\}$, and $\alpha \in \Sigma$,

- for all inclusions i of the form $Y \subseteq L_0 \cup L_1X_1 \cup \cdots \cup L_nX_n$ in $\mathcal{I}$, $\lambda \in \{0, \ldots, |\operatorname{Ind}(\mathcal{I})|\}$, and $\alpha \in \Sigma$,

$$\delta\big((\mathrm{i}, \lambda), \varepsilon\big) := \big\{g(X_i, \lambda) \mid i \in \{1, \ldots, n\}, \varepsilon \in L_i\big\} \cup \begin{cases} \{f_0\} & \text{if } \varepsilon \in L_0, \\ \emptyset & \text{otherwise,} \end{cases}$$

$$\delta\big((\mathrm{i}, \lambda), \alpha\big) := \big\{(X_i, 0) \mid i \in \{1, \ldots, n\}, \alpha \in L_i\big\} \cup \begin{cases} \{f_0\} & \text{if } \alpha \in L_0, \\ \emptyset & \text{otherwise.} \end{cases}$$

The auxiliary function g is defined as follows:

$$g(X_i, \lambda) := \begin{cases} (X_i, \lambda + 1) & \text{if } \lambda < |\operatorname{Ind}(\mathcal{I})|, \\ f_1 & \text{otherwise.} \end{cases}$$

In the case where there is one inclusion i of the form $X \subseteq L_0 \cup L_1X_1 \cup \cdots \cup L_nX_n$ 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 (i, λ). Thus, $\mathcal{A}_X$ will accept no word starting with α. This is consistent with the restriction imposed by 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 $|\operatorname{Ind}(\mathcal{I})|$, then some indeterminate must have occurred twice, that is, there must have been an ε-cycle. The automaton then goes to f_1, that is, it accepts everything that follows. 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. The use of this cycle detection mechanism is illustrated in the following example.

Example 23 Let $\mathcal{I}$ consist of the three inclusions

$$\mathrm{i}_1\colon X \subseteq \{r\} \cup \{\varepsilon\}Y, \qquad \mathrm{i}_2\colon Y \subseteq \{\varepsilon\}X, \qquad \text{and} \qquad \mathrm{i}_3\colon Y \subseteq \{s\}.$$

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 i_3 it can derive no contradiction, while from 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 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.

Since ε-AFAs only accept if there is a *finite* successful run, the restriction on the length of ε-paths is necessary. Otherwise, all runs starting in the configuration $((Y, 0), s)$ would have to be infinite.

In the following, we show that the automata introduced in Definition 22 actually accept the maximal solution of $\mathcal{I}$.

Theorem 24 *Let $X \in \operatorname{Ind}(\mathcal{I})$, and let θ^* be the maximal solution of $\mathcal{I}$. Then $L(\mathcal{A}_X) = \theta^*(X)$.*

Again, we prove the two directions of this equality separately.

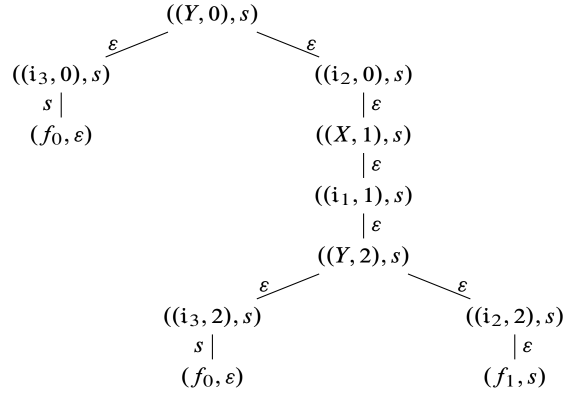

Figure 1 A successful run of the automaton $\mathcal{A}_Y$.

Lemma 25 *Let $X \in \mathrm{Ind}(\mathcal{I})$, and let θ^* be the maximal solution of $\mathcal{I}$. Then $\theta^*(X) \subseteq L(\mathcal{A}_X)$.*

Proof 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 $((\mathrm{i}, \ldots), u)$ for some inclusion $\mathrm{i} \in \mathcal{I}$ with Y on the left-hand side.

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

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

$$\mathrm{i}\colon Y \subseteq L_0 \cup L_1 X_1 \cup \cdots \cup L_n X_n$$

  in $\mathcal{I}$, then $P(v)$ yields $u \in \theta^*(Y)$. 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 $((X_i, \lambda + 1), u')$. Since $u' \in \theta^*(X_i)$, $P(v')$ is satisfied.
  - 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 $((X_i, 0), u')$. Again $P(v')$ is satisfied by the same reason as above.

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

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 $w \in L(\mathcal{A}_X)$. □

As we want to compute finite, local, admissible solutions of $\mathcal{I}_{\Gamma,\tau}$, we do not simply prove the other direction of Theorem 24, but a stronger result which also considers an adequate notion of locality for runs of our ε-AFA.

Definition 26 Let $\mathcal{A}$ be an ε-AFA. A successful run of $\mathcal{A}$ is called *local* if it has at least one leaf labeled by (q, ε) for some state q of $\mathcal{A}$. We denote by $L^{\text{local}}(\mathcal{A})$ the set of all words accepted by $\mathcal{A}$ via local, successful runs.

By definition, in any successful run with a leaf labeled by (q, ε), we know that q is either a final state or a universal state without ε-successors.

Lemma 27 *Let $\mathcal{A}$ be an ε-AFA. Then $L(\mathcal{A})$ is nonempty if and only if $L^{\text{local}}(\mathcal{A})$ is nonempty.*

Proof The if-direction is trivial. For the other direction, consider a successful run R of $\mathcal{A}$ that is not local. All leaves of this run are labeled by configurations (q, w) with $w \neq \varepsilon$. Thus, the states q have to be universal states 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 occurring in it and all other words in R must have the suffix w. Thus, we can 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, that is, words for which no prefix is accepted by $\mathcal{A}$, are always local. This is an important property of locality in ε-AFAs, which will turn out to be useful. We can now proceed to show (a modified version of) the remaining direction of Theorem 24.

Lemma 28 *If $w \in L^{\text{local}}(\mathcal{A}_X)$, 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 $Y \in \text{Ind}(\mathcal{I})$ is a suffix of w.*

Proof If $w \in L^{\text{local}}(\mathcal{A}_X)$, then there is a successful local run R of $\mathcal{A}_X$ on w. Let V denote the set of nodes of R. We restrict the set of 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, that is, 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

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

that ends in this leaf. We consider the smallest $j \in \{1, \ldots, |\text{Ind}(\mathcal{I})| + 1\}$ 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, the set V' no longer contains a node with an outgoing edge to f_1.

We now define the solution θ_R by

$$\theta_R(Y) := \big\{u \in \Sigma^* \;\big|\; \exists v \in V' : v \text{ is labeled by } \big((Y, \ldots), u\big)\big\}$$

for all $Y \in \text{Ind}(\mathcal{I})$. To show that this actually defines a solution of $\mathcal{I}$, we consider an inclusion

$$\text{i}\colon Y \subseteq L_0 \cup L_1 X_1 \cup \cdots \cup L_n X_n$$

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

- If $u = u'$, then $q \in \delta((\mathrm{i}, \lambda), \varepsilon)$. 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, $\lambda < |\operatorname{Ind}(\mathcal{I})|$ by the 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 $v'' \notin V'$, then there is an ancestor $\widetilde{v} \in V'$ of v'' with label $((X_i, \lambda'), u)$ 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 $q \in \delta((\mathrm{i}, \lambda), \alpha)$. One possibility is that $q = f_0$ 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 i under θ_R. Thus, θ_R is a solution of $\mathcal{I}$.

Since V' is a subset of the finite set of nodes of R, θ_R is finite. By the 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 occurring along p satisfies the conditions for the locality of θ_R. 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 vacuously fulfills the conditions for the 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 the locality of θ_R. If v' is the root node, then 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 the locality of θ_R, since u' does.
- Otherwise, $u = \alpha u'$ for some $\alpha \in \Sigma$ and q must be of the form (i, λ) for some inclusion $\mathrm{i}\colon Y \subseteq L_0 \cup L_1 X_1 \cup \cdots \cup L_n X_n$ 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 the definition of θ_R, and thus

$$u = \alpha u' \in \{\alpha\}\theta_R(X_i) \subseteq \big(L_i \setminus \{\varepsilon\}\big)\theta_R(X_i).$$

  Thus, the word u fulfills the condition of locality since it is contained in the right-hand side of i under θ_R. □

The proof of this lemma works for any word w accepted by a (possibly nonlocal) run of $\mathcal{A}_X$. The only difference is that the constructed finite solution θ of $\mathcal{I}$ that contains w need not be local. Since every finite solution of $\mathcal{I}$ is contained in the maximal solution θ^*, this, together with Lemma 27, concludes the proof of Theorem 24. We can now show the main results of this section.

Theorem 29 *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 [24, Theorem 1].

Let Γ be a flat $\mathcal{EL}^{-\top}$-unification problem. By Lemma 11, Theorem 14, and Lemma 20, we know that Γ is $\mathcal{EL}^{-\top}$-unifiable if and only if 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 $\tau\colon \mathrm{At}(\Gamma)^2 \to \{0,1\}$ 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 of whether it is a subsumption mapping for Γ can be done in polynomial time.

From τ we can first construct $\Delta_{\Gamma,\tau}$ and then $\mathcal{I}_{\Gamma,\tau}$ in polynomial time. Given $\mathcal{I}_{\Gamma,\tau}$, we then construct the (polynomially many) ε-AFA $\mathcal{A}_{X_A}$, and test them for emptiness. Since the emptiness of two-way alternating finite automata (where, in addition to normal and ε-transitions, backwards transitions are also allowed) can be tested in PSpace (see [21]), 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. □

We now modify this decision procedure such that it outputs a local $\mathcal{EL}^{-\top}$-unifier for any solvable $\mathcal{EL}^{-\top}$-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; more precisely, we need exponential time in the size of the input unification problem.

The algorithm uses the well-known reduction from any alternating automaton to an equivalent nondeterministic automaton of exponential size (see Birget [15], Chandra, Kozen, and Stockmeyer [17]). Additionally, it employs a polynomial-time algorithm to find shortest paths in a directed graph, for example, Dijkstra's algorithm [18]. This will be used to find a successful run of the nondeterministic automaton.

Theorem 30 *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 exponential time in the size of* Γ.

Proof 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 $\mathcal{A}(X) := \mathcal{A}_{X_{A(X)}}$ accepts a nonempty language. As detailed in the proof of Theorem 29, we can find $A(X)$ and construct $\mathcal{A}(X)$ in polynomial space—and therefore in exponential time—in the size of Γ.

For each X, we now construct a nondeterministic automaton $\mathcal{B}(X)$ that is equivalent to $\mathcal{A}(X)$ (see [15]). This automaton has as its 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 final states of $\mathcal{B}(X)$ are those sets that contain only final states of $\mathcal{A}(X)$ or universal states without successors. The size of $\mathcal{B}(X)$ is at most exponential in the size of $\mathcal{A}(X)$ and, thus, exponential in the size of Γ.

We now search for a successful run r of $\mathcal{B}(X)$ of minimal length, that is, a shortest path in the transition graph of $\mathcal{B}(X)$ that starts in the initial state $\{(X_{A(X)}, 0)\}$ and leads to a final state. Such a path can be found in exponential time using, for example, Dijkstra's algorithm [18]. It is clear that r is of size at most exponential in Γ and it accepts a word w_X that is of length exponential in Γ.

From the state sets occurring in r, a corresponding tree-shaped run R of $\mathcal{A}(X)$ can be reconstructed by the following procedure. We start with a single root node that is labeled by $((X_{A(X)}, 0), w_X)$ 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 occurring in the current tree, we can find a valid transition of $\mathcal{A}(X)$ 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 $\mathcal{A}(X)$ since 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 (see the proof of Lemma 27).

Thus, for every variable X of Γ, we can find a word $w_X \in L^{\text{local}}(\mathcal{A}(X))$ which is of length at most exponential in the size of Γ. By Lemma 28, we can construct a finite, local solution θ_X of $\mathcal{I}_{\Gamma,\tau}$ with $w_X \in \theta_X(X_{A(X)})$ 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.

The union θ of all the solutions 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 nonempty, and it is local since all contained words satisfy the conditions on locality by the locality of the component solutions θ_X. Finally, θ is of size exponential in the size of Γ since it is the union of the polynomially many solutions θ_X. By Lemmas 12, 17, and 18, we can now construct a local $\mathcal{EL}^{-\top}$-unifier of Γ of size exponential in the size of Γ. □

5 Hardness

In this section, we provide the corresponding hardness results to Theorems 29 and 30. We first reduce the intersection emptiness problem for deterministic finite automata (DFAs) to a unification problem in $\mathcal{EL}^{-\top}$. The intersection emptiness problem for DFAs is PSPACE-complete (see Garey and Johnson [19], Kozen [22]). Afterwards we will use this reduction to construct a series of solvable $\mathcal{EL}^{-\top}$-unification problems that only have local $\mathcal{EL}^{-\top}$-unifiers of exponential size.

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 $\delta\colon Q \times \Sigma \to \mathcal{P}(Q)$ that does not allow ε-transitions. The semantics of these automata is the same as for ε-AFAs, except that the relation $\vdash_{\mathcal{A}}$ is restricted to non-ε-transitions. The automaton is called a *nondeterministic finite automaton* (NFA) if $Q_\forall = \emptyset$ and is then written as $(Q, \Sigma, q_0, \delta, F)$. It is called a *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 $\delta' \colon Q \times \Sigma \to Q$, where $\delta'(q, \alpha) = q'$ if and only if $\delta(q, \alpha) = \{q'\}$. 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 $\mathsf{N_R} := \Sigma$. The set $\Gamma_{\mathcal{A}}$ is defined as follows:

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

Note that the left-hand sides of the subsumptions in $\Gamma_{\mathcal{A}}$ are indeed $\mathcal{EL}^{-\top}$-concept terms, that is, 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 31 *Let $q \in Q$, let $w \in \Sigma^*$, and let 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 $\mathcal{A}_q$.

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

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

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

Together with Lemma 2, this lemma implies that, for every ground $\mathcal{EL}^{-\top}$-unifier γ of $\Gamma_{\mathcal{A}}$, the language $\{w \in \Sigma^* \mid \exists w.A \in \mathrm{Part}(\gamma(X_{q_0}))\}$ is contained in $L(\mathcal{A})$. Conversely, we will show that for every word w accepted by $\mathcal{A}$ we can construct a unifier γ_w 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 32 *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 $u_q := \alpha u_{q'}$. □

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 $w = \alpha_1 \cdots \alpha_n$ yields a sequence of states $q_0, q_1, \ldots, q_n$ with $q_n \in F$ and $\delta(q_i, \alpha_{i+1}) = q_{i+1}$ for every $i \in \{0, \ldots, n-1\}$. We define the substitution γ_w as follows:

$$\gamma_w(X_q) := \exists u_q.A \sqcap \bigsqcap_{i \in I_q} \exists \alpha_{i+1} \cdots \alpha_n.A,$$

where $I_q := \{i \in \{0, \ldots, n-1\} \mid q_i = q\}$. 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 33 *If $w \in L(\mathcal{A})$, then γ_w is an $\mathcal{EL}^{-\top}$-unifier of $\Gamma_{\mathcal{A}}$ and we have $\gamma_w(X_{q_0}) \sqsubseteq \exists w.A$.*

Proof Let the unique successful run of $\mathcal{A}$ on $w = \alpha_1 \cdots \alpha_n$ be given by the sequence $q_0 q_1 \cdots q_n$ of states with $q_n \in F$ and $\delta(q_i, \alpha_{i+1}) = q_{i+1}$ for every $i \in \{0, \ldots, n-1\}$, and let γ_w be defined as above.

We have to show that γ_w satisfies the subsumption constraint introduced for every state $q \in Q$, that is,

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

if $q \notin F$ and

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

if $q \in F$. 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 A occurs on the left-hand side of the subsumption, which is thus 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 $i \in I_q$, that is, $q_i = q$, and consider the conjunct $\exists \alpha_{i+1} \cdots \alpha_n.A$. Since we have $\delta(q_i, \alpha_{i+1}) = q_{i+1}$ and $\exists \alpha_{i+2} \cdots \alpha_n.A$ is a conjunct of $\gamma_w(X_{q_{i+1}})$, we know that $\exists \alpha_{i+1}.\gamma_w(X_{q_{i+1}}) \sqsubseteq \exists \alpha_{i+1} \cdots \alpha_n.A$.

This shows that γ_w is a ground $\mathcal{EL}^{-\top}$-unifier of $\Gamma_{\mathcal{A}}$. Furthermore, $0 \in I_{q_0}$ implies that the particle $\exists \alpha_1 \cdots \alpha_n.A = \exists w.A$ is a top-level conjunct of $\gamma_w(X_{q_0})$, that is, $\gamma_w(X_{q_0}) \sqsubseteq \exists w.A$. □

The *intersection emptiness problem* considers finitely many DFAs $\mathcal{A}_1, \ldots, \mathcal{A}_k$, and asks whether $L(\mathcal{A}_1) \cap \cdots \cap L(\mathcal{A}_k) \neq \emptyset$. Since this problem is trivially solvable in polynomial time in the case in which $L(\mathcal{A}_i) = \emptyset$ for some i, $1 \leq i \leq k$, we can assume that the languages $L(\mathcal{A}_i)$ are all nonempty. Thus, we can also assume without loss of generality that the automata $\mathcal{A}_i = (Q_i, \Sigma, q_{0,i}, \delta_i, F_i)$ have pairwise disjoint sets of states Q_i and are reduced in the sense introduced above, that is, 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 34 *We have that Γ is unifiable in $\mathcal{EL}^{-\top}$ if and only if $L(\mathcal{A}_1) \cap \cdots \cap L(\mathcal{A}_k) \neq \emptyset$.*

Proof If Γ is unifiable in $\mathcal{EL}^{-\top}$, then it has a ground $\mathcal{EL}^{-\top}$-unifier γ. Thus, there must be a particle $\exists w.A$ with $w \in \Sigma^*$ and $\exists w.A \in \mathrm{Part}(\gamma(Y))$. This implies that $\gamma(X_{q_{0,i}}) \sqsubseteq \gamma(Y) \sqsubseteq \exists w.A$, and thus, Lemma 31 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 $w \in L(\mathcal{A}_1) \cap \cdots \cap L(\mathcal{A}_k)$. By Lemma 33, 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 (see [19], [22]), this lemma immediately yields our final theorem.

Theorem 35 *The problem of deciding unifiability in $\mathcal{EL}^{-\top}$ is* PSpace*-hard.*

This concludes our complexity analysis of the unification problem in $\mathcal{EL}^{-\top}$.

Corollary 36 *The problem of deciding unifiability in $\mathcal{EL}^{-\top}$ is* PSpace*-complete.*

The above construction also allows us to derive a lower bound on the size of local $\mathcal{EL}^{-\top}$-unifiers corresponding to the upper bound shown in Theorem 30. We can construct a series of solvable $\mathcal{EL}^{-\top}$-unification problems such that the size of any $\mathcal{EL}^{-\top}$-unifier is at least exponential in the size of the problem. The reason is that these unifiers must contain particles of exponential size. Since a particle is just a nesting of existential restrictions, exponential size also implies exponential depth.

Here, the *depth of a concept description* is the maximal nesting of existential restrictions; that is, given an $\mathcal{EL}^{-\top}$-concept description C, its depth $d(C)$ is defined as (i) 0 if C is a concept name, (ii) the maximum of $d(D)$ and $d(D')$ if $C = D \sqcap D'$, and (iii) $d(D) + 1$ if $C = \exists r.D$.

Example 37 We consider the proof of the PSPACE-completeness of the intersection emptiness problem for DFAs (see [22]). For any deterministic Turing machine $\mathcal{M}$ with polynomial space bound, the proof constructs several DFAs $\mathcal{A}_i$ ($i = 1, \ldots, k$) of size polynomial in the size of $\mathcal{M}$. The number k of these automata is also polynomial in the size of $\mathcal{M}$. These automata have the property that an input word u is accepted by $\mathcal{M}$ if and only if there is a successful run r of $\mathcal{M}$ on u such that $w_r \in \bigcap_{i=1}^{k} L(\mathcal{A}_i)$. 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, that is, it may be exponentially long. This means that the intersection $\bigcap_{i=1}^{k} L(\mathcal{A}_i)$ contains exactly the representations of all successful runs of $\mathcal{M}$.

For each $n \in \mathbb{N}$, consider the following $(n + 2)$-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, then $\mathcal{M}_n$ 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. The Turing machine $\mathcal{M}_n$ can be defined in such a way that it works only on the input tape section (and the two adjoining tape cells) 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 binary numbers.

For this deterministic Turing machine $\mathcal{M}_n$, we can now construct k DFAs $\mathcal{A}_i$ ($i = 1, \ldots, k$) 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 (see [22]). 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 34, we now construct a flat unification problem Γ_n of size polynomial in n that is unifiable in $\mathcal{EL}^{-\top}$ if and only if the intersection $\bigcap_{i=1}^{k} L(\mathcal{A}_i)$ is nonempty. 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 Lemmas 31 and 33, 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 at least exponential in the size of Γ_n and also of depth at least exponential in the size of Γ_n.

We thus have the following tight complexity bounds for the problem of computing a local $\mathcal{EL}^{-\top}$-unifier for a flat $\mathcal{EL}^{-\top}$-unification problem.

Corollary 38 *The depth of the concept terms in the image 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 of at most exponential size and depth in exponential time.*

The reason that we stress the exponential *depth* of the constructed $\mathcal{EL}^{-\top}$-unifiers rather than just the exponential size is that, already in $\mathcal{EL}$, local unifiers may have exponential *size*. However, as detailed in Section 3.2, local $\mathcal{EL}$-unifiers can be encoded by acyclic TBoxes of polynomial size such that the substitution can be computed as the unfolding with respect to this acyclic TBox. The depth of such an unfolding is always linear in the number of defined concept names, and thus, the depth of a concept term in a local $\mathcal{EL}$-unifier is always linear in the number of variables of the unification problem. However, in $\mathcal{EL}^{-\top}$ we cannot compress local unifiers into acyclic TBoxes in the same way since any acyclic TBox encoding of the local $\mathcal{EL}^{-\top}$-unifiers from Example 37 would still have to be of exponential size, as they contain particles of exponential depth.

6 Conclusion

Unification in $\mathcal{EL}$ was introduced in [10] 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 article investigated unification in $\mathcal{EL}^{-\top}$, which is obtained from $\mathcal{EL}$ by removing the top concept. More precisely, SNOMED CT is an acyclic $\mathcal{EL}^{-\top}$-TBox,[3] rather than a collection of $\mathcal{EL}^{-\top}$-concept terms. However, as shown in [12], acyclic TBoxes can be easily handled by a unification algorithm for concept terms. Furthermore, unification in $\mathcal{EL}^{-\top}$ is equivalent to unification modulo the equational theory *SLmO* and can be used to decide unification in the fragment of the modal logic K_m that is restricted to the connectives $\wedge$ and $\Diamond_{r_i}$.

Surprisingly, it turned out that the complexity of unification in $\mathcal{EL}^{-\top}$ (PSpace) is considerably higher than that 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 results in an increase of the complexity of the unifiability problem. A corresponding complexity increase also occurs when we consider the problem of computing local unifiers. For $\mathcal{EL}$, local unifiers are of polynomial size if represented by acyclic TBoxes. We have shown that a local $\mathcal{EL}^{-\top}$-unifier can be constructed by adding particles to an $\mathcal{EL}$-unifier that has a polynomial representation as an acyclic TBox. These particles may, however, be of exponential length, which cannot be compressed using acyclic TBoxes.

Apart from its theoretical interest, the results of this article also have practical implications. Whereas a practically rather efficient unification algorithm for $\mathcal{EL}$ can readily be obtained by a translation into SAT (see [11]), it is not so clear how to turn the PSpace algorithm for $\mathcal{EL}^{-\top}$-unification introduced in this article into a practically useful algorithm. One possibility could be to use a SAT modulo theory (SMT) approach (see Nieuwenhuis, Oliveras, and Tinelli [23]). 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. The theory solver basically has to solve a series of reachability tests between sets of states of our ε-AFA. How well this works will mainly depend on whether we can develop such a theory solver that efficiently solves these reachability problems and satisfies all the requirements imposed by the SMT approach.

Regarding the complexity of unification in more expressive DLs, not much is known. Allowing for a cycle-restricted form of general TBoxes in $\mathcal{EL}$ does not affect the complexity (see [6]). 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_m (see [25]). The decidability of unification in K is a long-standing open problem. The undecidability of unification in some extensions of K (for example, by the universal modality) was shown in [28]. These undecidability results also imply the undecidability of unification in some expressive DLs (e.g., in $\mathcal{SHIQ}$; see Horrocks, Sattler, and Tobies [20]).

Notes

1. See http://www.ihtsdo.org/snomed-ct/ for more information about SNOMED CT.

2. In [11], nearly the same conditions as in Definition 9 were expressed as propositional clauses to show that $\mathcal{EL}$-unifiability is in NP. It was shown in [11, Proposition 3.7] that γ^τ is actually an $\mathcal{EL}$-unifier of Γ.

3. Note that the right-identity rules in SNOMED CT are actually not expressed using complex role inclusion axioms, but through the SEP-triplet encoding (see Suntisrivaraporn, Baader, Schulz, et al. [27]). Thus, complex role inclusion axioms are not relevant here.

References

[1] Baader, F., "Terminological cycles in a description logic with existential restrictions," pp. 325–30 in *Proceedings of the 18th International Joint Conference on Artificial Intelligence* (*IJCAI'03*), edited by G. Gottlob and T. Walsh, Morgan Kaufmann, San Francisco, Calif., 2003. 444, 452

[2] Baader, F., N. T. Binh, S. Borgwardt, and B. Morawska, "Computing local unifiers in the description logic $\mathcal{EL}$ without the top concept," pp. 2–8 in *Proceedings of the 25th International Workshop on Unification* (*UNIF'11*), edited by F. Baader, B. Morawska, and J. Otop, 2011. 445

[3] Baader, F., N. T. Binh, S. Borgwardt, and B. Morawska, "Unification in the description logic $\mathcal{EL}$ without the top concept," pp. 70–84 in *Automated Deduction—CADE-23*, edited by N. Bjørner and V. Sofronie-Stokkermans, vol. 6803 of *Lecture Notes in Computer Science*, Springer, Heidelberg, 2011. MR 2862860. DOI 10.1007/978-3-642-22438-6_9. 445

[4] Baader, F., S. Borgwardt, and B. Morawska, "A goal-oriented algorithm for unification in $\mathcal{ELH}_{R^+}$ w.r.t. cycle-restricted ontologies," pp. 493–504 in *AI 2012: Advances in Artificial Intelligence*, edited by M. Thielscher and D. Zhang, vol. 7691 of *Lecture Notes in Computer Science*, Springer, Heidelberg, 2012. MR 3067175. DOI 10.1007/978-3-642-35101-3_42. 445

[5] Baader, F., S. Borgwardt, and B. Morawska, "SAT- encoding of unification in $\mathcal{ELH}_{R^+}$ w.r.t. cycle-restricted ontologies," pp. 30–44 in *Automated Reasoning*, edited by B. Gramlich, D. Miller, and U. Sattler, vol. 7364 of *Lecture Notes in Computer Science*, Springer, Heidelberg, 2012. MR 2992325. DOI 10.1007/978-3-642-31365-3_5.

[6] Baader, F., S. Borgwardt, and B. Morawska, "Extending unification in $\mathcal{EL}$ towards general TBoxes," pp. 568–72 in *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR'12)*, edited by G. Brewka, T. Eiter, and S. A. McIlraith, AAAI Press, Palo Alto, Calif., 2012. 445, 450, 474

[7] Baader, F., S. Brandt, and C. Lutz, "Pushing the $\mathcal{EL}$ envelope," pp. 364–69 in *Proceedings of the 19th International Joint Conference on Artificial Intelligence* (*IJCAI'05*), edited by L. P. Kaelbling and A. Saffiotti, Morgan Kaufmann, San Francisco, Calif., 2005. 444

[8] Baader, F., D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, eds., *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, Cambridge, 2003. MR 1991592. 443, 446

[9] Baader, F., and S. Ghilardi, "Unification in modal and description logics," *Logic Journal of the IGPL*, vol. 19 (2011), pp. 705–30. Zbl 1258.03018. MR 2844520. DOI 10.1093/jigpal/jzq008. 450, 451

[10] Baader, F., and B. Morawska, "Unification in the description logic $\mathcal{EL}$," pp. 350–64 in *Rewriting Techniques and Applications*, edited by R. Treinen, vol. 5595 of *Lecture Notes in Computer Science*, Springer, Berlin, 2009. MR 2545489. DOI 10.1007/978-3-642-02348-4_25. 444, 445, 446, 451, 473

[11] Baader, F., and B. Morawska, "SAT encoding of unification in $\mathcal{EL}$," pp. 97–111 in *Logic for Programming, Artificial Intelligence, and Reasoning*, edited by C. G. Fermüller and A. Voronkov, vol. 6397 of *Lecture Notes in Computer Science*, Springer, Berlin, 2010. MR 2926341. DOI 10.1007/978-3-642-16242-8_8. 445, 473, 474

[12] Baader, F., and B. Morawska, "Unification in the description logic $\mathcal{EL}$," *Logical Methods in Computer Science*, vol. 6 (2010), no. 17. MR 2718113. DOI 10.2168/LMCS-6(3:17)2010. 445, 447, 449, 450, 451, 452, 473

[13] Baader, F., and P. Narendran, "Unification of concept terms in description logics," *Journal of Symbolic Computation*, vol. 31 (2001), pp. 277–305. MR 1814334. DOI 10.1006/jsco.2000.0426. 444

[14] Baader, F., and W. Snyder, "Unification theory," pp. 445–532 in *Handbook of Automated Reasoning*, edited by J. A. Robinson and A. Voronkov, MIT Press, Cambridge, Mass., 2001. 449

[15] Birget, J.-C., "State-complexity of finite-state devices, state compressibility and incompressibility," *Mathematical Systems Theory*, vol. 26 (1993), pp. 237–69. Zbl 0779.68061. MR 1209997. DOI 10.1007/BF01371727. 467

[16] Blackburn, P., J. van Benthem, and F. Wolter, *Handbook of Modal Logic*, Elsevier, 2006. 450

[17] Chandra, A. K., D. C. Kozen, and L. J. Stockmeyer, "Alternation," *Journal of the ACM*, vol. 28 (1981), pp. 114–33. Zbl 0473.68043. MR 0603186. DOI 10.1145/322234.322243. 467

[18] Dijkstra, E. W., "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1 (1959), pp. 269–71. MR 0107609. 467, 468

[19] Garey, M. R., and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, Calif., 1979. MR 0519066. 468, 471

[20] Horrocks, I., U. Sattler, and S. Tobies, "Practical reasoning for very expressive description logics," *Logic Journal of the IGPL*, vol. 8 (2000), pp. 239–64. Zbl 0967.03026. MR 1768945. DOI 10.1093/jigpal/8.3.239. 474

[21] Jiang, T., and B. Ravikumar, "A note on the space complexity of some decision problems for finite automata," *Information Processing Letters*, vol. 40 (1991), pp. 25–31. Zbl 0741.68078. MR 1134005. DOI 10.1016/S0020-0190(05)80006-7. 461, 467

[22] Kozen, D., "Lower bounds for natural proof systems," pp. 254–66 in *18th Annual Symposium on Foundations of Computer Science (Providence, R.I., 1977)*, IEEE Computer Science, Long Beach, Calif., 1977. MR 0495200. 468, 471, 472

[23] Nieuwenhuis, R., A. Oliveras, and C. Tinelli, "Solving SAT and SAT modulo theories: From an abstract Davis-Putnam-Logemann-Loveland procedure to DPLL(T)," *Journal of the ACM*, vol. 53 (2006), pp. 937–77. MR 2282098. DOI 10.1145/1217856.1217859. 473

[24] Savitch, W. J., “Relationships between nondeterministic and deterministic tape complexities,” *Journal of Computer and System Sciences*, vol. 4 (1970), pp. 177–92. Zbl 0188.33502. MR 0266702. 467

[25] Schild, K., “A correspondence theory for terminological logics: Preliminary report,” pp. 466–71 in *Proceedings of the 12th International Joint Conference on Artificial Intelligence* (*IJCAI'91*), Kaufmann, San Francisco, Calif., 1991. 450, 474

[26] Sofronie-Stokkermans, V., “Locality and subsumption testing in $\mathcal{EL}$ and some of its extensions,” pp. 315–39 in *Advances in Modal Logic, vol. 7*, edited by C. Areces and R. Goldblatt, College Publications, London, 2008. MR 2642652. 449

[27] Suntisrivaraporn, B., F. Baader, S. Schulz, and K. Spackman, “Replacing SEP-triplets in SNOMED CT using tractable description logic operators,” pp. 287–91 in *Proceedings of the 11th Conference on Artificial Intelligence in Medicine* (*AIME'07*), edited by R. Bellazzi, A. Abu-Hanna, and J. Hunter, vol. 4594 of *Lecture Notes in Computer Science*, Springer, Berlin, 2007. 474

[28] Wolter, F., and M. Zakharyaschev, “Undecidability of the unification and admissibility problems for modal and description logics,” *ACM Transactions on Computational Logic*, vol. 9 (2008), no. 25. MR 2444463. DOI 10.1145/1380572.1380574. 451, 474

Acknowledgment

This work was supported by the Deutsche Forschungsgemeinschaft under grant BA 1122/14-1.

Baader
Faculty of Computer Science
Technische Universität Dresden
01062 Dresden
Germany
baader@tcs.inf.tu-dresden.de

Binh
Department of Computer Science
Eidgenössische Technische Hochschule (ETH) Zürich
8092 Zürich
Switzerland
thannguy@inf.ethz.ch

Borgwardt
Faculty of Computer Science
Technische Universität Dresden
01062 Dresden
Germany
stefborg@tcs.inf.tu-dresden.de

Morawska
Faculty of Computer Science
Technische Universität Dresden
01062 Dresden
Germany
morawska@tcs.inf.tu-dresden.de