Restricted Unification in the DL $\mathcal{EL}$

Franz Baader[1] and Maryam Rostamigiv[2]

[1] Theoretical Computer Science, TU Dresden, Dresden, Germany
franz.baader@tu-dresden.de
[2] Département d'Informatique, Paul Sabatier University, Toulouse, France
Maryam.Rostamigiv@irit.fr

Abstract. Research on unification in Description Logic (DL) has concentrated on the lightweight DLs $\mathcal{FL}_0$ and $\mathcal{EL}$. For both DLs, the unification type is zero, which is the worst possible type. The complexity of deciding unifiability is ExpTime-complete for $\mathcal{FL}_0$ and NP-complete for $\mathcal{EL}$. In a recent paper, we have shown that, for $\mathcal{FL}_0$, both the unification type and the complexity of the decision problem can be improved by considering restricted versions of the unification problem where either the role depth of concepts is restricted syntactically or the length of role paths in interpretations is restricted semantically. In the present paper, we show that no such improvements can be obtained for $\mathcal{EL}$: both in the syntactically and in the semantically restricted case, the unification type stays zero, and the complexity of the decision problem stays NP-complete.

1 Introduction

Unification theory [13] investigates the unification properties of equational theories (such as associativity and commutativity of a binary function symbol). Given an equational theory E and terms s, t, the decision problem asks whether s and t can be made equal modulo E by replacing variables by terms, i.e., whether there is a substitution σ such that $\sigma(s) =_E \sigma(t)$. In addition to testing unifiability, one is also interested in computing a small representation of all unifiers, i.e., a set of unifiers from which all unifiers can be obtained by instantiation. Such a set is called a complete set of unifiers. From a practical point of view, it is useful to have finite complete sets of unifiers. If the theory has unification type zero, then there is a unification problem w.r.t. this theory such that every complete set of unifiers is redundant in the sense that it contains unifiers that are in an instance relationship, which implies that the unification problem cannot have a finite complete set of unifiers. Unification in modal and description logics [5] can be seen as a special case of unification w.r.t. an equational theory, where the theory axiomatizes equivalence of formulae/concepts (viewed as terms) in the respective logic.

Unification of concept patterns has been proposed as a nonstandard inference service in DL that can, for instance, be used to detect redundancies in ontologies. For example, assume that one ontologist describes the medical concept of myocarditis by the $\mathcal{EL}$ concept $\mathit{HeartDisease} \sqcap \exists \mathit{cause}.\mathit{Virus}$ whereas another one describes it by $\mathit{ViralInfection} \sqcap \exists \mathit{location}.\mathit{Heart}$. These two concepts are not equivalent, but they can be unified (i.e., made equivalent) if we view *HeartDisease* and *ViralInfection* as variables that can be replaced by complex concepts. In fact, replacing *HeartDisease* with $\mathit{Disease} \sqcap \exists \mathit{location}.\mathit{Heart}$ in the first concept yields $\mathit{Disease} \sqcap \exists \mathit{location}.\mathit{Heart} \sqcap \exists \mathit{cause}.\mathit{Virus}$, and replacing *ViralInfection* with $\mathit{Disease} \sqcap \exists \mathit{cause}.\mathit{Virus}$ in the second concept yields $\mathit{Disease} \sqcap \exists \mathit{cause}.\mathit{Virus} \sqcap \exists \mathit{location}.\mathit{Heart}$. These two concepts are clearly equivalent, and they actually yield an appropriate description of myocarditis.

For the DL $\mathcal{FL}_0$, which has the concept constructors conjunction ($\sqcap$), value restriction ($\forall r.C$), and top concept ($\top$), unification was investigated in detail in [12]. It was shown there that unification in $\mathcal{FL}_0$ is ExpTime-complete and of unification type zero. The DL $\mathcal{EL}$, which has the concept constructors conjunction ($\sqcap$), existential restriction ($\exists r.C$), and top concept ($\top$), also has unification type zero, but the complexity of the decision problem is lower than for $\mathcal{FL}_0$: it is "only" NP-complete for $\mathcal{EL}$ [11].

In [3], we investigate two kinds of restrictions on unification in $\mathcal{FL}_0$. On the one hand, we *syntactically restrict the role depth* (i.e., the maximal nesting of value restrictions) in the concepts obtained by applying a unifier to be bounded by a natural number $k \geq 1$. This restriction was motivated by a similar restriction used in research on least common subsumers (lcs) [19], where imposing a bound on the role depth guarantees the existence of the lcs also in the presence of a (possibly cyclic) terminology. Also note that such a restriction was used in [16] for the equational theory ACh, for which unification is known to be undecidable [18]. It is shown in [16] that the problem becomes decidable if a bound on the maximal nesting of applications of homomorphisms is imposed. On the other hand, we consider in [3] a *semantic restriction* where, when defining the semantics of concepts, only interpretations for which the length of role paths is bounded by a given number k are considered. A similar restriction (for $k = 1$) was employed in [14] to improve the unification type from type zero for the modal logic $\mathbf{K}$ [15] to finitary for $\mathbf{K} + \Box\Box\bot$, which means that unification problems always have a finite complete set of unifiers in this extension of $\mathbf{K}$. We show in [3] that both the syntactic and the semantic restriction ensures that the unification type of $\mathcal{FL}_0$ improves from type zero to finitary. Regarding the decision problem, we show that the complexity depends on whether the bound k is assumed to be encoded in unary or binary. For binary encoding of k, the complexity stays ExpTime, whereas for unary coding it drops from ExpTime to PSpace. This is again the case both for the syntactic and the semantic restriction.

In the present paper, we investigate whether similar improvements of the unification type and the complexity of the decision problem can be obtained for the DL $\mathcal{EL}$. Surprisingly, the answer to this question is "no." Both in the syntactically and in the semantically restricted case, the unification type stays zero, and the complexity of the decision problem stays NP-complete for $\mathcal{EL}$. This demonstrates that the reason for type zero and the complexity lower bound to hold are different for $\mathcal{FL}_0$ and $\mathcal{EL}$. Whereas in $\mathcal{FL}_0$ they depend on the possibility of arbitrarily deep nesting of role restrictions, this is not the case for $\mathcal{EL}$. In fact, we will see that for $\mathcal{EL}$ already a nesting depth of $k = 1$ suffices to obtain type zero and NP-hardness.

2 The DL $\mathcal{EL}$ and Restrictions

Starting with mutually disjoint countably infinite sets N_C and N_R of concept and role names, the set of $\mathcal{EL}$ *concepts* is inductively defined as follows:

- $\top$ (top concept) and every concept name $A \in N_C$ is an $\mathcal{EL}$ concept,
- if C, D are $\mathcal{EL}$ concepts and $r \in N_R$ is a role name, then $C \sqcap D$ (conjunction) and $\exists r.C$ (existential restriction) are $\mathcal{EL}$ concepts.

The *semantics* of $\mathcal{EL}$ concepts is defined using interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a non-empty domain $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ that assigns a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ to each concept name A, and a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ to each role name r. This function is extended to $\mathcal{EL}$ concepts as follows:

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}} \text{ and } (C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}},$$
$$(\exists r.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}}\colon (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}.$$

Given two $\mathcal{EL}$ concepts C and D, we say that C is *subsumed* by D (written $C \sqsubseteq D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all interpretations $\mathcal{I}$, and that C is *equivalent* to D (written $C \equiv D$) if $C \sqsubseteq D$ and $D \sqsubseteq C$.

It is well known that subsumption (and thus also equivalence) of $\mathcal{EL}$ concepts can be decided in polynomial time. This was first shown in [7] based on the notion of a homomorphism between trees representing concepts, but it is also an easy consequence of the following recursive characterization of subsumption.

Lemma 1 ([11]). *Let $C = A_1 \sqcap \ldots \sqcap A_k \sqcap \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_m.C_m$ and $D = B_1 \sqcap \ldots \sqcap B_\ell \sqcap \exists s_1.D_1 \sqcap \ldots \sqcap \exists s_n.D_n$, where $A_1, \ldots, A_k, B_1, \ldots, B_\ell$ are concept names. Then $C \sqsubseteq D$ iff $\{B_1, \ldots, B_\ell\} \subseteq \{A_1, \ldots, A_k\}$ and for every $j, 1 \le j \le n$, there exists an $i, 1 \le i \le m$, such that $r_i = s_j$ and $C_i \sqsubseteq D_j$.*

2.1 Syntactically Restricting the Role Depth

The *role depth* of an $\mathcal{EL}$ concept is the maximal nesting of existential restrictions in this concept. To be more precise, the role depth $rd(C)$ of the $\mathcal{EL}$ concept C is defined by induction:

- $rd(\top) = rd(A) = 0$ for all $A \in N_C$,
- $rd(C \sqcap D) = \max(rd(C), rd(D))$ and $rd(\exists r.C) = 1 + rd(C)$.

Using Lemma 1, it is easy to see that $C \sqsubseteq D$ implies $rd(C) \geq rd(D)$. Thus, the role depth of $\mathcal{EL}$ concepts is preserved under equivalence.

We are now ready to define our first restricted version of subsumption and equivalence in $\mathcal{EL}$. For an integer $k \geq 1$ and $\mathcal{EL}$ concepts C and D, we define subsumption and equivalence restricted to concepts of role depth $\leq k$ as follows:

- $C \sqsubseteq^k_{syn} D$ if $C \sqsubseteq D$ and $rd(C) \leq k$,
- $C \equiv^k_{syn} D$ if $C \sqsubseteq^k_{syn} D$ and $D \sqsubseteq^k_{syn} C$.

The effect of this definition is that subsumption and equivalence can only hold for concepts that satisfy the restriction of the role depth by k. For concepts satisfying this syntactic restriction, the relations $\sqsubseteq^k_{syn}$ and $\equiv^k_{syn}$ coincide with the classical subsumption and equivalence relations on $\mathcal{EL}$ concepts.

2.2 Semantically Restricting the Length of Role Paths

For an integer $n \geq 1$ and a given interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, a role path of length n is a sequence $d_0, r_1, d_1, \ldots, d_{n-1}, r_n, d_n$, where $d_0, \ldots, d_n$ are elements of $\Delta^{\mathcal{I}}$, $r_1, \ldots, r_n$ are role names, and $(d_{i-1}, d_i) \in r_i^{\mathcal{I}}$ holds for all $i = 1, \ldots, n$. The interpretation $\mathcal{I}$ is called *k-restricted* if it does not admit any role paths of length $> k$.

For an integer $k \geq 1$ and $\mathcal{EL}$ concepts C and D, we define subsumption and equivalence restricted to interpretations with role paths of length $\leq k$ as follows:

- $C \sqsubseteq^k_{sem} D$ if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all k-restricted interpretations $\mathcal{I}$,
- $C \equiv^k_{sem} D$ if $C \sqsubseteq^k_{sem} D$ and $D \sqsubseteq^k_{sem} C$.

The effect of this semantic restriction is that concepts of role depth $> k$ are always interpreted as empty sets, i.e., if $\mathcal{I}$ is a k-restricted interpretation and C an $\mathcal{EL}$ concept such that $rd(C) > k$, then $C^{\mathcal{I}} = \emptyset$. The following lemma is an easy consequence of this fact, where $\bot$ denotes the *bottom concept*, which is interpreted as $\bot^{\mathcal{I}} = \emptyset$ in every interpretation $\mathcal{I}$.

Lemma 2. *Let $k \geq 1$ and C, D be $\mathcal{EL}$ concepts. Then $C \equiv^k_{sem} \bot$ iff $rd(C) > k$, and $C \equiv^k_{sem} D$ iff*

- *$rd(C) > k$ and $rd(D) > k$, or*
- *$rd(C) \leq k$, $rd(D) \leq k$, and $C \equiv D$.*

3 Unification in $\mathcal{EL}$

In unification, we consider concepts that may contain variables, which can be replaced by concepts. More formally, we introduce a countably infinite set N_V of *concept variables*, which is disjoint with N_C and N_R. An *$\mathcal{EL}$ concept pattern* is an $\mathcal{EL}$ concept that is constructed using $N_C \cup N_V$ as concept names. The semantics of concept patterns is defined as for concepts, i.e., concept variables are treated like concept names when defining the semantics. This way, the notions of subsumption and equivalence (both in the restricted and in the unrestricted setting) transfer from concepts to concept patterns in the obvious way.

A *substitution* σ is a mapping from N_V into the set of all $\mathcal{EL}$ concept patterns such that $dom(\sigma) := \{X \in N_V \mid \sigma(X) \neq X\}$ is finite. This mapping is extended to concept patterns in the obvious way:

– $\sigma(A) := A$ for all $A \in N_C \cup \{\top\}$,
– $\sigma(C \sqcap D) := \sigma(C) \sqcap \sigma(D)$ and $\sigma(\exists r.C) := \exists r.\sigma(C)$.

An $\mathcal{EL}$ *unification problem* is an equation of the form $C \overset{?}{\equiv} D$ where C, D are $\mathcal{EL}$ concept patterns. A *unifier* (or *solution*) of this equation is a substitution σ such that $\sigma(C) \equiv \sigma(D)$.

Example 3. Consider the $\mathcal{EL}$ unification problem

$$\exists r.X \sqcap \exists s.\exists r.A \overset{?}{\equiv} \exists r.\exists s.Y \sqcap \exists s.Y.$$

It is easy to see that (up to equivalence and restricted to the variables X, Y) the substitution σ_{ex} with $dom(\sigma_{ex}) = \{X, Y\}$, $\sigma_{ex}(X) = \exists s.\exists r.A$, and $\sigma_{ex}(Y) = \exists r.A$ is the only unifier of this problem.

3.1 The Unification Type

When considering the unification type of an equational theory or logic, one is interested in the question of whether all unifiers of a given unification problem can be represented as instances of a finite set of unifiers, where the instance relation between unifiers is defined as follows: given unifiers σ, θ of $C \overset{?}{\equiv} D$, we say that θ is an *instance* of σ if there is a substitution λ such that $\theta(X) \equiv \lambda(\sigma(X))$ for all variables X occurring in C or D.

A set of unifiers M of an $\mathcal{EL}$ unification problem $C \overset{?}{\equiv} D$ is *complete* if any unifier of $C \overset{?}{\equiv} D$ is an instance of an element of M. This set is *minimal* if no two distinct elements of M are comparable w.r.t. the instance relation. The unification problem $C \overset{?}{\equiv} D$ has *type zero* if it does not have a minimal complete set of unifiers. Note that this implies that $C \overset{?}{\equiv} D$ does not have a finite complete set of unifiers since such a set could be made minimal by removing unifiers that are instances of others [13]. Saying that $\mathcal{EL}$ has *unification type zero* means that there is an $\mathcal{EL}$ unification problem that has type zero. Thus, the following proposition implies that $\mathcal{EL}$ indeed has unification type zero.

Proposition 4 ([11]). *Let X, Y be concept variables. The $\mathcal{EL}$ unification problem $X \sqcap \exists r.Y \overset{?}{\equiv} \exists r.Y$ has unification type zero.*

The proof of this proposition given in [11], which shows that any complete set of unifiers M of $X \sqcap \exists r.Y \overset{?}{\equiv} \exists r.Y$ is non-minimal, proceeds as follows:

1. It observes that the substitution τ with $\tau(X) = \exists r.A$ and $\tau(Y) = A$ is a unifier, and thus there must be an element σ of M such that τ is an instance of σ. It shows that this unifier σ satisfies $\sigma(X) \not\equiv \top$ and $\sigma(X) \not\equiv \exists r.\top$.

2. It then proves that the substitution $\widehat{\sigma}$ defined as
$$\begin{aligned}\widehat{\sigma}(X) &:= \sigma(X) \sqcap \exists r.Z,\\ \widehat{\sigma}(Y) &:= \sigma(Y) \sqcap Z,\end{aligned} \qquad \text{where } Z \text{ is a new variable,}$$
is a unifier of $X \sqcap \exists r.Y \overset{?}{\equiv} \exists r.Y$ that has σ as an instance.
3. Since M is complete, it concludes that there is a unifier θ in M such that $\widehat{\sigma}$ is an instance of θ. Since the instance relation is transitive, this implies that σ is an instance of θ.
4. Using the fact that $\sigma(X) \not\equiv \top$ and $\sigma(X) \not\equiv \exists r.\top$, it proves that θ cannot be equal to σ, and concludes non-minimality of M.

We will show later that this proof also works in the restricted case.

3.2 The Decision Problem

NP-hardness already holds for the case of matching, where we call an $\mathcal{EL}$ unification problem of the form $C \overset{?}{\equiv} D$ a *matching problem* if D does not contain concept variables. A unifier of a matching problem is also called *matcher*. NP-hardness of matching can be shown by a reduction from SAT, i.e., satisfiability of propositional formulae. In the literature, one can actually find two such reductions with different characteristics:

(R1) In [6], a given propositional formula ϕ is translated into an $\mathcal{EL}$ matching problem $C \overset{?}{\equiv} D$ such that the role depth of both C and D is 1, and the number of different role names occurring in C and D is linear in the size of ϕ.

(R2) In [17] (proof of Corollary 6.3.4, pages 185 and 186), a given propositional formula ϕ is translated into an $\mathcal{EL}$ matching problem $C \overset{?}{\equiv} D$ such that the role depth of both C and D is linear in the size of ϕ, and C, D contain only 4 different role names.

For the case of matching, an NP-upper bound was also shown in [6,17]. It took almost ten years before an NP-upper bound for unification in $\mathcal{EL}$ could be proved in [9,11].

Theorem 5 ([9,11]). *Unification in $\mathcal{EL}$ is NP-complete.*

Showing the NP-upper bound for unification in $\mathcal{EL}$ is more involved than proving the lower bound, though the original proof given in [9,11] was simplified in later work [10,1]. The basic idea underlying the proof is to show the following:

1. Every solvable $\mathcal{EL}$ unification problem $C \overset{?}{\equiv} D$ has a local unifier, i.e., a unifier that is a local substitution. Intuitively, local substitutions are built from subconcepts of the concepts C, D (see below for more details).
2. Polynomially large representations of local substitutions can be guessed by a non-deterministic procedure in polynomial time.
3. Given such a representation of a local substitution σ, it can be checked in polynomial time whether σ is a unifier of $C \overset{?}{\equiv} D$.

To be more precise, it is shown in [9,11,1] that a given $\mathcal{EL}$ unification problem $C \overset{?}{\equiv} D$ determines a set of flat atoms $At(C, D)$, whose cardinality is linear in the size of the problem. A *flat atom* is a concept name, a concept variable, or an existential restriction of the form $\exists r.E$, where E is either a concept name or a concept variable. Such an atom is called *non-variable* if it is not a concept variable. The subset of non-variable atoms in $At(C, D)$ is denoted as $At_{nv}(C, D)$.

Local substitutions are induced by *assignments* S, which assign subsets S_X of $At_{nv}(C, D)$ to the concept variables occurring on C, D. To induce a local substitution, such an assignment needs to be *acyclic*, which means that the transitive closure $>_S$ of

$$O_S := \{(X, Y) \mid X, Y \text{ are variables in } C \text{ or } D \text{ and } Y \text{ occurs in } S_X\}$$

is irreflexive (and thus a partial order). Any acyclic assignment S induces a unique substitution σ_S, which can be defined by induction along $>_S$:

- If X is a minimal variable w.r.t. $>_S$, then $\sigma_S(X) := \bigsqcap_{E \in S_X} E$.
- Assume that $\sigma_S(Y)$ is already defined for all Y such that $X >_S Y$. Then $\sigma_S(X) := \bigsqcap_{E \in S_X} \sigma_S(E)$.

A substitution σ is called *local* if it is of this form, i.e., if there is an acyclic assignment S such that $\sigma = \sigma_S$. If the unifier σ of $C \overset{?}{\equiv} D$ is a local substitution, then it is called a *local unifier.*

Whereas assignments are of size polynomial in the size of the input unification problem $C \overset{?}{\equiv} D$ (and thus can be guessed by an NP procedure), local substitutions may assign exponentially large concepts to a variable. Nevertheless, given an acyclic assignment S, one can check in polynomial time whether σ_S is a unifier, basically by viewing the assignment as an acyclic TBox (see [11] for details).

The proof that every solvable $\mathcal{EL}$ unification problem $C \overset{?}{\equiv} D$ has a local unifier given in [11] proceeds in two steps. First, it uses component-wise subsumption to define an order on substitutions:

$$\sigma \succeq \theta \text{ iff } \sigma(X) \sqsubseteq \theta(X) \text{ holds for all variables } X \text{ occurring in } C, D,$$
$$\sigma \succ \theta \text{ iff } \sigma \succeq \theta \text{ and } \theta \not\succeq \sigma.$$

The unifier σ of $C \overset{?}{\equiv} D$ is called *minimal* if there is no unifier θ of $C \overset{?}{\equiv} D$ such that $\sigma \succ \theta$. It is then shown in [11] that the pre-order $\succ$ is well-founded, which immediately implies that very solvable $\mathcal{EL}$ unification problem $C \overset{?}{\equiv} D$ has a minimal unifier. The second step, which is considerably more involved, is then to prove that every minimal unifier is local (see Proposition 5.11 in [11]). A direct proof that every solvable $\mathcal{EL}$ unification problem has a local unifier that does not use minimal unifiers can be found in [1].

4 Syntactically Restricted Unification in $\mathcal{EL}$

For an integer $k \geq 1$, a *syntactically k-restricted unification problem* is an equation of the form $C \overset{?}{\equiv}^k_{syn} D$, where C, D are $\mathcal{EL}$ concept patterns. A unifier of this equation (also called *syntactically k-restricted unifier*) is a substitution σ such that $\sigma(C) \equiv^k_{syn} \sigma(D)$.

If we view the unification problem in Example 3 as a syntactically k-restricted unification problem, then it does not have a solution for $k \leq 2$, but the substitution σ_{ex} is a syntactically k-restricted unifier of this problem for all $k \geq 3$.

The following lemma is an immediate consequence of the definition of $\equiv^k_{syn}$ and the fact that $rd(\sigma(X)) \leq rd(\sigma(D))$ for every substitution σ and every concept variable X occurring in the $\mathcal{EL}$ concept D.

Lemma 6. *If σ is a syntactically k-restricted unifier of $C \overset{?}{\equiv}{}^k_{syn} D$, then σ is a unifier of $C \overset{?}{\equiv} D$ and $rd(\sigma(X)) \leq k$ for every concept variable X occurring in C or D.*

Consequently, the instance relation on syntactically k-restricted unifiers (which is defined using $\equiv^k_{syn}$) coincides with the instance relation on unifiers (which is defined using $\equiv$). However, the set of syntactically k-restricted unifiers may of course be a strict subset of the set of unrestricted unifiers. For this reason, it is not immediately clear that type zero transfers from unification to syntactically k-restricted unification. To see that this is nevertheless the case, one needs to look more closely at the proof of Proposition 4 sketched in the previous section.

Proposition 7. *Let X, Y be concept variables. The syntactically k-restricted $\mathcal{EL}$ unification problem $X \sqcap \exists r.Y \overset{?}{\equiv}{}^k_{syn} \exists r.Y$ has unification type zero for every $k \geq 1$.*

Proof. Similarly to the proof of Proposition 4, we consider an arbitrary complete set M of syntactically k-restricted unifiers of $X \sqcap \exists r.Y \overset{?}{\equiv}{}^k_{syn} \exists r.Y$. We show that all the steps made in this proof also work in the syntactically k-restricted case.

1. The unifier τ with $\tau(X) = \exists r.A$ and $\tau(Y) = A$ considered in the first step of that proof is clearly also a syntactically k-restricted unifier for every $k \geq 1$. Since M is complete for such unifiers, τ is an instance[1] of an element $\sigma \in M$. As before, we conclude that σ satisfies $\sigma(X) \not\equiv \top$ and $\sigma(X) \not\equiv \exists r.\top$.
2. Since σ is a syntactically k-restricted unifier, the same is true for the substitution $\widehat{\sigma}$ constructed in the second step. In fact, we already know that $\widehat{\sigma}$ is a unifier, and the concepts

   $$\widehat{\sigma}(X \sqcap \exists r.Y) = \sigma(X) \sqcap \exists r.Z \sqcap \exists r.(\sigma(Y) \sqcap Z) \text{ and } \widehat{\sigma}(\exists r.Y) = \exists r.(\sigma(Y) \sqcap Z)$$

   have role depth 1 since σ is a syntactically k-restricted unifier of $X \sqcap \exists r.Y \overset{?}{\equiv}{}^k_{syn} \exists r.Y$, which implies $rd(\sigma(X)) \leq 1$ and $rd(\sigma(Y)) = 0$.
3. The third step works as above, but we now also know that the unifier $\theta \in M$ that has σ as an instance is syntactically k-restricted.
4. The proof that σ and θ cannot be equal again works as for the unrestricted case since the same notion of instance is employed, due to Lemma 6.

[1] Recall that, by Lemma 6, we can assume that this is the classical instance relation.

This finishes the proof that M cannot be minimal. Since M was chosen as an arbitrary complete set of syntactically k-restricted unifiers of $X \sqcap \exists r.Y \overset{?}{\equiv}{}^{k}_{syn} \exists r.Y$, this shows that this unification problem has unification type zero. □

Theorem 8. *Syntactically k-restricted $\mathcal{EL}$ unification has unification type zero for any $k \geq 1$.*

Let us now turn to the complexity of the decision problem. In this context, one can either assume that the bound k is fixed, or that it is part of the input. In the latter case, one can distinguish between unary and binary coding of the number k. We want to show that the same reductions from SAT as in the unrestricted setting can be used to show NP-hardness of the matching problem. For this, the observations made in the next lemma are useful.

Lemma 9. *Let $C \overset{?}{\equiv}{}^{k}_{syn} D$ be a syntactically k-restricted $\mathcal{EL}$ matching problem.*

1. *If $rd(D) > k$, then $C \overset{?}{\equiv}{}^{k}_{syn} D$ does not have a solution.*
2. *Otherwise, a substitution σ is a syntactically k-restricted matcher of $C \overset{?}{\equiv}{}^{k}_{syn} D$ iff it is a matcher of $C \overset{?}{\equiv} D$.*

Consequently, the reduction (R1) works for any fixed $k \geq 1$ since the concept D used in that reduction has role depth 1. However, this reduction requires an unbounded supply of role names.

Proposition 10. *For every fixed $k \geq 1$, syntactically k-restricted matching is NP-hard if the number of available role names is not bounded by a finite number.*

The reduction (R2) requires only four role names, but the role depth of D is linear in the size of the propositional formula. Thus, this reduction only works if we assume the bound k to be part of the input, but the reduction is polynomial even if unary representation of k is assumed.

Proposition 11. *Syntactically k-restricted matching is NP-hard even if only four role names are available and k is encoded in unary.*

We show the NP-upper bound for the most complex case of unification where k is part of the input and assumed to be encoded in binary. The idea is to employ basically the same NP procedure as in the unrestricted case, but additionally check whether the restriction of the role depth is satisfied. Given a syntactically k-restricted unification problem $C \overset{?}{\equiv}{}^{k}_{syn} D$, we

1. guess a local substitution σ in non-deterministic polynomial time;
2. check whether $\sigma(C) \equiv \sigma(D)$ and $rd(\sigma(C)) \leq k$.

To see that the second test checking the bound on the role depth can also be realized in polynomial time, it is sufficient to prove the following lemma.

Lemma 12. *Let S be an acyclic assignment for the unification problem $C \overset{?}{\equiv} D$. Then we can compute $rd(\sigma_S(X))$ for every concept variable X occurring in C, D in time polynomial in the size of $C \overset{?}{\equiv} D$.*

Proof. This can easily be shown by induction along $>_S$, following the inductive definition of the substitution σ_S. $\square$

It remains to show that this NP procedure is complete, i.e., that it is really the case that any solvable syntactically k-restricted unification problem $C \,{}^?{\equiv}^k_{syn}\, D$ has a local substitution as a solution. We already know (see Section 3) that, for any unifier σ of $C \,{}^?{\equiv}\, D$, there is a minimal unifier θ of $C \,{}^?{\equiv}\, D$ such that $\sigma \succeq \theta$. In addition, θ is local. Using the fact that $E \sqsubseteq F$ implies $rd(E) \geq rd(F)$, we can conclude that, if σ is a syntactically k-restricted unifier, then so is θ. This completes the proof of completeness.

Theorem 13. *Syntactically k-restricted unification in $\mathcal{EL}$ is NP-complete.*

5 Semantically Restricted Unification in $\mathcal{EL}$

For an integer $k \geq 1$, a *semantically k-restricted unification problem* is an equation of the form $C \,{}^?{\equiv}^k_{sem}\, D$, where C, D are $\mathcal{EL}$ concept patterns. A unifier of this equation (also called *semantically k-restricted unifier*) is a substitution σ such that $\sigma(C) \equiv^k_{sem} \sigma(D)$.

In contrast to the syntactic case, solvability in the semantically k-restricted setting does not imply solvability in the unrestricted setting. For example, $\exists r.X \,{}^?{\equiv}\, \exists s.X$ for distinct role names r, s clearly does not have a solution, but the substitution σ with $\sigma(X) := \exists r.A$ is a solution if we view this equation as a semantically 1-restricted unification problem. More generally, the following holds.

Lemma 14. *If C, D both contain concept variables, then $C \,{}^?{\equiv}^k_{sem}\, D$ always has a solution.*

Proof. Assume that the variable X occurs in C and the variable Y occurs in D, and let E be an $\mathcal{EL}$ concept with $rd(E) > k$. Let θ be the substitution defined as $\theta(X) := \theta(Y) := E$ and $\theta(Z) := \top$ for all variables Z in C, D that are different from X and Y. Then $\theta(C) \equiv^k_{sem} \bot \equiv^k_{sem} \theta(D)$. $\square$

We call a solution σ of $C \,{}^?{\equiv}^k_{sem}\, D$ *trivial* if $\sigma(C) \equiv^k_{sem} \bot \equiv^k_{sem} \sigma(D)$. The non-trivial semantically k-restricted unifiers are just the syntactically k-restricted unifiers. This is an easy consequence of Lemma 2.

Lemma 15. *The substitution σ is a non-trivial semantically k-restricted unifier of $C \,{}^?{\equiv}^k_{sem}\, D$ iff σ is a syntactically k-restricted unifier of $C \,{}^?{\equiv}^k_{syn}\, D$.*

If we view the unification problem in Example 3 as a semantically k-restricted unification problem, then it does not have a non-trivial solution for $k \leq 2$, but setting $\sigma(X) := \sigma(Y) := \exists r.\exists r.A$ yields a trivial solution for $k \leq 2$. The substitution σ_{ex} is a non-trivial semantically k-restricted unifier of this problem for all $k \geq 3$.

Matching problems $C \,{}^?{\equiv}^k_{sem}\, D$ can only have trivial solutions if $rd(D) > k$.

Lemma 16. *Let $C \overset{?}{\equiv}^k_{sem} D$ be a matching problem such that $rd(D) \leq k$. Then the set of solutions of $C \overset{?}{\equiv}^k_{sem} D$ coincides with the set of solutions of $C \overset{?}{\equiv}^k_{syn} D$.*

Using this lemma, it is easy to see that the NP-hardness results shown in the previous section also hold in the semantically restricted case. We state here only the general NP-hardness result, but note that the more fine-grained complexity results given in Propositions 10 and 11 also apply.

Proposition 17. *Semantically k-restricted matching is NP-hard.*

To show the NP-upper bound for unification, we can proceed as follows for a given semantically k-restricted unification problem $C \overset{?}{\equiv}^k_{sem} D$:

1. Check whether $C \overset{?}{\equiv}^k_{sem} D$ has a trivial solution. This is the case if the following holds for all $E \in \{C, D\}$: E contains a variable or $rd(E) > k$. If this test succeeds, then answer "unifiable." Otherwise, proceed with the next step.
2. If $C \overset{?}{\equiv}^k_{sem} D$ does not have a trivial solution, then check whether it has a non-trivial solution by testing whether $C \overset{?}{\equiv}^k_{syn} D$ has a solution, using the NP procedure described in the previous section.

Overall, this yields an NP procedure for testing solvability of semantically k-restricted unification problems.

Theorem 18. *Semantically k-restricted unification in $\mathcal{EL}$ is NP-complete.*

Unification type zero can be shown by a simple adaptation of the approach employed above for the syntactically k-restricted case. The proof uses the fact that, if θ is an instance of σ, then $rd(\theta(X)) \geq rd(\sigma(X))$ for all variables occurring in the unification problem. Consequently, a trivial solution cannot have a non-trivial solution as an instance.

Theorem 19. *Semantically k-restricted $\mathcal{EL}$ unification has unification type zero for any $k \geq 1$.*

6 Conclusion

We have investigated both a semantically and a syntactically restricted variant of unification in $\mathcal{EL}$, where either the role depth of concepts or the length of role paths in interpretations is restricted by a natural number $k \geq 1$. In contrast to the case of $\mathcal{FL}_0$, for $\mathcal{EL}$ these restrictions do not lead to an improvement of the unification type or the complexity of the decision problem. For $\mathcal{FL}_0$, the "bad" behaviour w.r.t. the unification type and the complexity of the decision problem depends on the ability to nest value restrictions arbitrarily deep. In contrast, for $\mathcal{EL}$ the "bad" behaviour already shows up for $k = 1$, where no nesting of existential restrictions is allowed.

Nevertheless, it may still make sense to consider the syntactically restricted variant of unification also for $\mathcal{EL}$. In fact, in our experiments with the system UEL, which implements several unification algorithms for the DL $\mathcal{EL}$ [8], we have observed that the algorithms usually yield many different unifiers, and it is hard to choose one that is appropriate for the application at hand (e.g., when generating new concepts using unification [2]). For this reason, we added additional constraints to the unification problem to ensure that the generated concepts are of a similar shape as the concepts already present in the ontology [2]. It makes sense also to use a restriction on the role depth as such an additional constraint since the role depth of the (unfolded) concepts occurring in real-world ontologies is usually rather small. This claim is supported by our experiments with the medical ontology SNOMED CT,[2] which has a maximal role depth of 10, and the acyclic ontologies in Bioportal 2017,[3] where a large majority also has a role depth of at most 10. The NP procedure for syntactically k-restricted unification described in Section 4 is, however, not very useful in this setting since it basically first computes all local unifiers of the unrestricted problem, and then throws away the ones that lead to a unified concept whose role depth is too large. Thus, it would be interesting to find a procedure for syntactically k-restricted unification in $\mathcal{EL}$ that directly generates the local unifiers that are syntactically k-restricted, without first generating all local unifiers.

Both for $\mathcal{FL}_0$ and for $\mathcal{EL}$, decidability of unification in the presence of terminologies (TBoxes) consisting of general concept inclusions is an open problem. In the restricted setting, decidability should follow from the fact that, up to equivalence, there can be only finitely many concepts of a bounded role depth if only finitely many concept names and role names are available. However, for $\mathcal{EL}$ the number of such concepts increase by one exponent with every increase of the role depth bound. Thus, it would be interesting to see whether an elementary decision procedure can be obtained for $\mathcal{EL}$ in the restricted setting.

Acknowledgements

Franz Baader was partially supported by DFG TRR 248 (cpec, grant 389792660), and Maryam Rostamigiv by a DAAD Short-Term Grant, 2021 (57552336). The authors should like to thank Patrick Koopmann for determining the maximal role depth of concepts in ontologies from Bioportal 2017 and in SNOMED CT.

References

1. Franz Baader, Stefan Borgwardt, and Barbara Morawska. Extending unification in $\mathcal{EL}$ towards general TBoxes. In *Proc. of the 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2012)*, pages 568–572. AAAI Press/The MIT Press, 2012.

[2] https://www.snomed.org/

[3] https://zenodo.org/record/439510

2. Franz Baader, Stefan Borgwardt, and Barbara Morawska. Constructing SNOMED CT concepts via disunification. LTCS-Report 17-07, Chair for Automata Theory, Institute for Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany, 2017. `https://lat.inf.tu-dresden.de/research/reports/2017/BaBM-LTCS-17-07.pdf`.
3. Franz Baader, Oliver Fernández Gil, and Maryam Rostamigiv. Restricted unification in the description logic $\mathcal{FL}_0$. In Boris Konev and Giles Reger, editors, *Proc. of the 13th International Symposium on Frontiers of Combining Systems (FroCoS 2021)*, volume 12941 of *Lecture Notes in Computer Science*. Springer, 2021. To appear. A long version of this paper containing detailed proofs has been published as technical report [4].
4. Franz Baader, Oliver Fernández Gil, and Maryam Rostamigiv. Restricted unification in the DL $\mathcal{FL}_0$ (extended version). LTCS-Report 21-02, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany, 2021. `https://lat.inf.tu-dresden.de/research/reports/2021/BaGiRo21.pdf`.
5. Franz Baader and Silvio Ghilardi. Unification in modal and description logics. *Log. J. IGPL*, 19(6):705–730, 2011.
6. Franz Baader and Ralf Küsters. Matching in description logics with existential restrictions. In *Proc. of the 7th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2000)*, pages 261–272, 2000.
7. Franz Baader, Ralf Küsters, and Ralf Molitor. Computing least common subsumers in description logics with existential restrictions. In *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI'99)*, pages 96–101, 1999.
8. Franz Baader, Julian Mendez, and Barbara Morawska. UEL: Unification solver for the description logic $\mathcal{EL}$ – system description. In *Proc. of the 6th International Joint Conference on Automated Reasoning (IJCAR 2012)*, volume 7364 of *Lecture Notes in Artificial Intelligence*, pages 45–51. Springer, 2012.
9. Franz Baader and Barbara Morawska. Unification in the description logic $\mathcal{EL}$. In Ralf Treinen, editor, *Proc. of the 20th Int. Conf. on Rewriting Techniques and Applications (RTA 2009)*, volume 5595 of *Lecture Notes in Computer Science*, pages 350–364. Springer-Verlag, 2009.
10. Franz Baader and Barbara Morawska. SAT encoding of unification in $\mathcal{EL}$. In C. Fermüller and A. Voronkov, editors, *Proc. of the 17th Int. Conf. on Logic for Programming, Artifical Intelligence, and Reasoning (LPAR-17)*, volume 6397 of *Lecture Notes in Computer Science*, pages 97–111, Yogyakarta (Indonesia), 2010. Springer-Verlag.
11. Franz Baader and Barbara Morawska. Unification in the description logic $\mathcal{EL}$. *Logical Methods in Computer Science*, 6(3), 2010.
12. Franz Baader and Paliath Narendran. Unification of concept terms in description logics. *J. Symbolic Computation*, 31(3):277–305, 2001.
13. Franz Baader and Wayne Snyder. Unification theory. In J.A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, pages 447–533. Elsevier Science Publishers, 2001.
14. Philippe Balbiani, Cigdem Gencer, Maryam Rostamigiv, and Tinko Tinchev. About the unification type of $\mathbf{K} + \Box\Box\bot$. In *Proc. of the 34th International Workshop on Unification (UNIF 2020)*, pages 4:1–4:6. RISC-Linz, 2020.
15. Emil Jerabek. Blending margins: The modal logic K has nullary unification type. *J. Logic and Computation*, 25(5):1231–1240, 2015.
16. Ajay Kumar Eeralla and Christopher Lynch. Bounded ACh unification. *Math. Struct. Comput. Sci.*, 30(6):664–682, 2020.

17. Ralf Küsters. *Non-standard Inferences in Description Logics*, volume 2100 of *Lecture Notes in Artificial Intelligence.* Springer-Verlag, 2001.
18. Paliath Narendran. Solving linear equations over polynomial semirings. In *Proc. of the 11th Annual IEEE Symposium on Logic in Computer Science (LICS 1996)*, pages 466–472. IEEE Computer Society, 1996.
19. Rafael Peñaloza and Anni-Yasmin Turhan. A practical approach for computing generalization inferences in $\mathcal{EL}$. In Grigoris Antoniou, Marko Grobelnik, Elena Paslaru Bontas Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter De Leenheer, and Jeff Z. Pan, editors, *Proc. of the 8th Extended Semantic Web Conference (ESWC 2011)*, volume 6643 of *Lecture Notes in Computer Science*, pages 410–423. Springer, 2011.