Unification in the Description Logic $\mathcal{EL}$

Franz Baader and Barbara Morawska

Theoretical Computer Science, TU Dresden, Germany
{baader,morawska}@tcs.inf.tu-dresden.de

Abstract. The Description Logic $\mathcal{EL}$ has recently drawn considerable attention since, on the one hand, important inference problems such as the subsumption problem are polynomial. On the other hand, $\mathcal{EL}$ is used to define large biomedical ontologies. Unification in Description Logics has been proposed as a novel inference service that can, for example, be used to detect redundancies in ontologies. The main result of this paper is that unification in $\mathcal{EL}$ is decidable. More precisely, $\mathcal{EL}$-unification is NP-complete, and thus has the same complexity as $\mathcal{EL}$-matching. We also show that, w.r.t. the unification type, $\mathcal{EL}$ is less well-behaved: it is of type zero, which in particular implies that there are unification problems that have no finite complete set of unifiers.

1 Introduction

Description logics (DLs) [5] are a family of logic-based knowledge representation formalisms, which can be used to represent the conceptual knowledge of an application domain in a structured and formally well-understood way. They are employed in various application domains, such as natural language processing, configuration, databases, and biomedical ontologies, but their most notable success so far is the adoption of the DL-based language OWL [15] as standard ontology language for the semantic web.

In DLs, concepts are formally described by *concept terms*, i.e., expressions that are built from concept names (unary predicates) and role names (binary predicates) using concept constructors. The expressivity of a particular DL is determined by which concept constructors are available in it. From a semantic point of view, concept names and concept terms represent sets of individuals, whereas roles represent binary relations between individuals. For example, using the concept name Woman, and the role name child, the concept of all *women having a daughter* can be represented by the concept term

$$\mathsf{Woman} \sqcap \exists \mathsf{child}.\mathsf{Woman},$$

and the concept of all *women having only daughters* by

$$\mathsf{Woman} \sqcap \forall \mathsf{child}.\mathsf{Woman}.$$

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 Woman subsumes the concept term $\mathsf{Woman} \sqcap \exists \mathsf{child}.\mathsf{Woman}$ since all instances of the second term are also instances of the first term, i.e., the second term is always interpreted as a subset of the first term. With the help of the subsumption algorithm, a newly introduced concept term can automatically be placed at the correct position in the hierarchy of the already existing concept terms.

Two concept terms C, D are *equivalent* ($C \equiv D$) if they subsume each other, i.e., if they always represent the same set of individuals. For example, the terms $\forall \mathsf{child}.\mathsf{Rich} \sqcap \forall \mathsf{child}.\mathsf{Woman}$ and $\forall \mathsf{child}.(\mathsf{Rich} \sqcap \mathsf{Woman})$ are equivalent since the value restriction operator ($\forall r.C$) distributes over the conjunction operator ($\sqcap$). If we replaced the value restriction operator by the existential restriction operator ($\exists r.C$), then this equivalence would no longer hold. However, for this operator, we still have the equivalence

$$\exists \mathsf{child}.\mathsf{Rich} \sqcap \exists \mathsf{child}.(\mathsf{Woman} \sqcap \mathsf{Rich}) \equiv \exists \mathsf{child}.(\mathsf{Woman} \sqcap \mathsf{Rich}).$$

The equivalence test can, for example, be used to find out whether a concept term representing a particular notion has already been introduced, thus avoiding multiple introduction of the same concept into the concept hierarchy. This inference capability is very important if the knowledge base containing the concept terms is very large, evolves during a long time period, and is extended and maintained by several knowledge engineers. However, testing for equivalence of concepts is not always sufficient to find out whether, for a given concept term, there already exists another concept term in the knowledge base describing the same notion. For example, assume that one knowledge engineer has defined the concept of all *women having a daughter* by the concept term

$$\mathsf{Woman} \sqcap \exists \mathsf{child}.\mathsf{Woman}.$$

A second knowledge engineer might represent this notion in a somewhat more fine-grained way, e.g., by using the term $\mathsf{Female} \sqcap \mathsf{Human}$ in place of Woman. The concept terms $\mathsf{Woman} \sqcap \exists \mathsf{child}.\mathsf{Woman}$ and

$$\mathsf{Female} \sqcap \mathsf{Human} \sqcap \exists \mathsf{child}.(\mathsf{Female} \sqcap \mathsf{Human})$$

are not equivalent, but they are meant to represent the same concept. The two terms can obviously be made equivalent by substituting the concept name Woman in the first term by the concept term $\mathsf{Female} \sqcap \mathsf{Human}$. This leads us to *unification of concept terms*, i.e., the question whether two concept terms can be made equivalent by applying an appropriate substitution, where a substitution replaces (some of the) concept names by concept terms. Of course, it is not necessarily the case that unifiable concept terms are meant to represent the same notion. A unifiability test can, however, suggest to the knowledge engineer possible candidate terms.

Unification in DLs was first considered in [9] for a DL called $\mathcal{FL}_0$, which has the concept constructors *conjunction* ($\sqcap$), *value restriction* ($\forall r.C$), and the *top concept* ($\top$). It was shown that unification in $\mathcal{FL}_0$ is decidable and ExpTime-complete, i.e., given an $\mathcal{FL}_0$-unification problem, we can effectively decide whether it has a solution or not, but in the worst-case, any such decision procedure needs exponential time. This result was extended in [7] to a more expressive DL, which additional has the role constructor *transitive closure*. Interestingly, the *unification type* of $\mathcal{FL}_0$ had been determined almost a decade earlier in [1]. In fact, as shown in [9], unification in $\mathcal{FL}_0$ corresponds to unification modulo the equational theory of idempotent Abelian monoids with several homomorphisms. In [1] it was shown that, already for a single homomorphism, unification modulo this theory has unification type zero, i.e., there are unification problems for this theory that do not have a minimal complete set of unifiers. In particular, such unification problems cannot have a finite complete set of unifiers.

In this paper, we consider unification in the DL $\mathcal{EL}$. The $\mathcal{EL}$-family consists of inexpressive DLs whose main distinguishing feature is that they provide their users with *existential restrictions* ($\exists r.C$) rather than value restrictions ($\forall r.C$) as the main concept constructor involving roles. The core language of this family is $\mathcal{EL}$, which has the top concept, conjunction, and existential restrictions as concept constructors. This family has recently drawn considerable attention since, on the one hand, the subsumption problem stays tractable (i.e., decidable in polynomial time) in situations where $\mathcal{FL}_0$, the corresponding DL with value restrictions, becomes intractable: subsumption between concept terms is tractable for both $\mathcal{FL}_0$ and $\mathcal{EL}$, but allowing the use of concept definitions or even more expressive terminological formalisms makes $\mathcal{FL}_0$ intractable [2, 16, 4], whereas it leaves $\mathcal{EL}$ tractable [3, 13, 4]. On the other hand, although of limited expressive power, $\mathcal{EL}$ is nevertheless used in applications, e.g., to define biomedical ontologies. For example, both the large medical ontology SNOMED CT[1] and the Gene Ontology[2] can be expressed in $\mathcal{EL}$, and the same is true for large parts of the medical ontology GALEN [18]. The importance of $\mathcal{EL}$ can also be seen from the fact that the new OWL 2 standard[3] contains a sub-profile OWL 2 EL, which is based on (an extension of) $\mathcal{EL}$.

Unification in $\mathcal{EL}$ has, to the best of our knowledge, not been investigated before, but matching (where one side of the equation(s) to be solved does not contain variables) has been considered in [6, 17]. In particular, it was shown in [17] that the decision problem, i.e., the problem of deciding whether a given $\mathcal{EL}$-matching problem has a matcher or not, is NP-complete. Interestingly, $\mathcal{FL}_0$ behaves better w.r.t. matching than $\mathcal{EL}$: for $\mathcal{FL}_0$, the decision problem is tractable [8]. In this paper, we show that, w.r.t. the unification type, $\mathcal{FL}_0$ and $\mathcal{EL}$ behave the same: just as $\mathcal{FL}_0$, the DL $\mathcal{EL}$ has unification type zero. However, w.r.t. the decision problem, $\mathcal{EL}$ behaves much better than $\mathcal{FL}_0$: $\mathcal{EL}$-unification is NP-complete, and thus has the same complexity as $\mathcal{EL}$-matching.

---

1 http://www.ihtsdo.org/snomed-ct/
2 http://www.geneontology.org/
3 See http://www.w3.org/TR/owl2-profiles/

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

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

In the next section, we define the DL $\mathcal{EL}$ and unification in $\mathcal{EL}$ more formally. In Section 3, we recall the characterisation of subsumption and equivalence in $\mathcal{EL}$ from [17], and in Section 4 we use this to show that unification in $\mathcal{EL}$ has type zero. In Section 5, we show that unification in $\mathcal{EL}$ is NP-complete, and in Section 6 we point out that our results for $\mathcal{EL}$-unification imply that unification modulo the equational theory of semilattices with monotone operators [19] is NP-complete and of unification type zero.

More information about Description Logics can be found in [5], and about unification theory in [12].

2 Unification in $\mathcal{EL}$

First, we define the syntax and semantics of $\mathcal{EL}$-concept terms as well as the subsumption and the equivalence relation on these terms.

Starting with a set N_{con} of concept names and a set N_{role} of role names, *$\mathcal{EL}$-concept terms* are built using the concept constructors top concept ($\top$), conjunction ($\sqcap$), and existential restriction ($\exists r.C$). The semantics of $\mathcal{EL}$ is defined in the usual way, using the notion of an interpretation $\mathcal{I} = (\mathcal{D}_{\mathcal{I}}, \cdot^{\mathcal{I}})$, which consists of a nonempty domain $\mathcal{D}_{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ that assigns binary relations on $\mathcal{D}_{\mathcal{I}}$ to role names and subsets of $\mathcal{D}_{\mathcal{I}}$ to concept terms, as shown in the semantics column of Table 1.

The concept term C *is subsumed by* the concept term D (written $C \sqsubseteq D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all interpretations $\mathcal{I}$. We say that C *is equivalent to* D (written $C \equiv D$) iff $C \sqsubseteq D$ and $D \sqsubseteq C$, i.e., iff $C^{\mathcal{I}} = D^{\mathcal{I}}$ holds for all interpretations $\mathcal{I}$. The concept term C *is strictly subsumed by* the concept term D (written $C \sqsubset D$) iff $C \sqsubseteq D$ and $C \not\equiv D$.

A *concept definition* is of the form $A \doteq C$ where A is a concept name and C is a concept term. A *TBox* $\mathcal{T}$ is a finite set of concept definitions such that no concept name occurs more than once on the left-hand side of a concept definition in $\mathcal{T}$. The TBox $\mathcal{T}$ is called *acyclic* if there are no cyclic dependencies between its concept definitions. The interpretation $\mathcal{I}$ is a model of the TBox $\mathcal{T}$ iff $A^{\mathcal{I}} = C^{\mathcal{I}}$ holds for all concept definitions $A \doteq C$ in $\mathcal{T}$. Subsumption and equivalence w.r.t. a TBox are defined as follows: $C \sqsubseteq_{\mathcal{T}} D$ ($C \equiv_{\mathcal{T}} D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ ($C^{\mathcal{I}} = D^{\mathcal{I}}$) holds for all models $\mathcal{I}$ of $\mathcal{T}$. Subsumption and equivalence w.r.t. an acyclic TBox can be reduced to subsumption and equivalence of concept terms (without TBox) by *expanding* the concept terms w.r.t. the TBox, i.e., by replacing defined concepts (i.e., concept names occurring on the left-hand side of a definition) by their definitions (i.e., the corresponding right-hand sides) until all defined concepts have been replaced. This expansion process may, however, result in an exponential blow-up [10].

In order to define unification of concept terms, we first introduce the notion of a substitution operating on concept terms. To this purpose, we partition the set of concepts names into a set N_v of concept variables (which may be replaced by substitutions) and a set N_c of concept constants (which must not be replaced by substitutions). Intuitively, N_v are the concept names that have possibly been given another name or been specified in more detail in another concept term describing the same notion. The elements of N_c are the ones of which it is assumed that the same name is used by all knowledge engineers (e.g., standardised names in a certain domain).

A *substitution* σ is a mapping from N_v into the set of all $\mathcal{EL}$-concept terms. This mapping is extended to concept terms in the obvious way, i.e.,

- $\sigma(A) := A$ for all $A \in N_c$,
- $\sigma(\top) := \top$,
- $\sigma(C \sqcap D) := \sigma(C) \sqcap \sigma(D)$, and
- $\sigma(\exists r.C) := \exists r.\sigma(C)$.

Definition 1. *An $\mathcal{EL}$-*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 $\mathcal{EL}$-concept terms. The substitution σ is a* unifier *(or* solution*) of Γ iff $\sigma(C_i) \equiv \sigma(D_i)$ for $i = 1, \ldots, n$. In this case, Γ is called* solvable *or* unifiable.

When we say that $\mathcal{EL}$-unification is *decidable (NP-complete)*, then we mean that the following decision problem is decidable (NP-complete): given an $\mathcal{EL}$-unification problem Γ, decide whether Γ is solvable or not.

As usual, unifiers can be compared using the instantiation preorder $\lessdot$. Let Γ be an $\mathcal{EL}$-unification problem, V the set of variables occurring in Γ, and σ, θ two unifiers of this problem. We define

$$\sigma \lessdot \theta \text{ iff there is a substitution } \lambda \text{ such that } \theta(X) \equiv \lambda(\sigma(X)) \text{ for all } X \in V.$$

If $\sigma \lessdot \theta$, then we say that θ is an *instance* of σ.

Definition 2. *Let Γ be an $\mathcal{EL}$-unification problem. The set of substitutions M is called a* complete set of unifiers *for Γ iff it satisfies the following two properties:*

1. *every element of M is a unifier of Γ;*
2. *if θ is a unifier of Γ, then there exists a unifier $\sigma \in M$ such that $\sigma \lessdot \theta$.*

The set M is called a minimal complete set of unifiers *for Γ iff it additionally satisfies*

3. if $\sigma, \theta \in M$, then $\sigma \lessdot \theta$ implies $\sigma = \theta$.

The unification type of a given unification problem is determined by the existence and cardinality of such a minimal complete set.

Definition 3. *Let Γ be an $\mathcal{EL}$-unification problem. This problem has type* unitary *(*finitary*,* infinitary*) iff it has a minimal complete set of unifiers of cardinality* 1 *(finite cardinality, infinite cardinality). If Γ does not have a minimal complete set of unifiers, then it is of type* zero*.*

Note that the set of all unifiers of a given $\mathcal{EL}$-unification problem is always a complete set of unifiers. However, this set is usually infinite and redundant (in the sense that some unifiers are instances of others). For a unitary or finitary $\mathcal{EL}$-unification problem, all unifiers can be represented by a finite complete set of unifiers, whereas for problems of type infinitary or zero this is no longer possible. In fact, if a problem has a finite complete set of unifiers M, then it also has a finite *minimal* complete set of unifiers, which can be obtained by iteratively removing redundant elements from M. For an infinite complete set of unifiers, this approach of removing redundant unifiers may be infinite, and the set reached in the limit need no longer be complete. This is what happens for problems of type zero. The difference between infinitary and type zero is that a unification problem of type zero cannot even have a non-redundant complete set of unifiers, i.e., every complete set of unifiers must contain different unifiers σ, θ such that $\sigma \lessdot \theta$.

When we say that $\mathcal{EL}$ *has unification type zero*, we mean that there exists an $\mathcal{EL}$-unification problem that has type zero. Before we can prove that this is indeed the case, we must first have a closer look at equivalence in $\mathcal{EL}$.

3 Equivalence and subsumption in $\mathcal{EL}$

In order to characterise equivalence of $\mathcal{EL}$-concept terms, the notion of a reduced $\mathcal{EL}$-concept term is introduced in [17]. A given $\mathcal{EL}$-concept term can be transformed into an equivalent reduced term by applying the following rules modulo associativity and commutativity of conjunction:

$C \sqcap \top \rightarrow C$	for all $\mathcal{EL}$-concept terms C
$A \sqcap A \rightarrow A$	for all concept names $A \in N_{con}$
$\exists r.C \sqcap \exists r.D \rightarrow \exists r.C$	for all $\mathcal{EL}$-concept terms C, D with $C \sqsubseteq D$

Obviously, these rules are equivalence preserving. We say that the $\mathcal{EL}$-concept term C is *reduced* if none of the above rules is applicable to it (modulo associativity and commutativity of $\sqcap$). The $\mathcal{EL}$-concept term D is a *reduced form* of C if D is reduced and can be obtained from C by applying the above rules (modulo associativity and commutativity of $\sqcap$). The following theorem is an easy consequence of Theorem 6.3.1 on page 181 of [17].

Theorem 1. *Let C, D be $\mathcal{EL}$-concept terms, and $\widehat{C}, \widehat{D}$ reduced forms of C, D, respectively. Then $C \equiv D$ iff $\widehat{C}$ is identical to $\widehat{D}$ up to associativity and commutativity of $\sqcap$.*

This theorem can also be used to derive a recursive characterisation of subsumption in $\mathcal{EL}$. In fact, if $C \sqsubseteq D$, then $C \sqcap D \equiv C$, and thus C and $C \sqcap D$ have the same reduced form. Thus, during reduction, all concept names and existential restrictions of D must be "eaten up" by corresponding concept names and existential restrictions of C.

Corollary 1. *Let $C = A_1 \sqcap \ldots \sqcap A_k \sqcap \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_m.C_m$ and $D = B_1 \sqcap \ldots \sqcap B_\ell \sqcap \exists s_1.D_1 \sqcap \ldots \sqcap \exists s_n.D_n$, 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 $j, 1 \leq j \leq n$, there exists an $i, 1 \leq i \leq m$, such that $r_i = s_j$ and $C_i \sqsubseteq D_j$.*

Note that this corollary also covers the cases where some of the numbers k, ℓ, m, n are zero. The empty conjunction should then be read as $\top$. The following lemma, which is an immediate consequence of this corollary, will be used in our proof that $\mathcal{EL}$ has unification type zero.

Lemma 1. *If C, D are reduced $\mathcal{EL}$-concept terms such that $\exists r.D \sqsubseteq C$, then C is either $\top$, or of the form $C = \exists r.C_1 \sqcap \ldots \sqcap \exists r.C_n$ where $n \geq 1$; $C_1, \ldots, C_n$ are reduced and pairwise incomparable w.r.t. subsumption; and $D \sqsubseteq C_1, \ldots, D \sqsubseteq C_n$. Conversely, if C, D are $\mathcal{EL}$-concept terms such that $C = \exists r.C_1 \sqcap \ldots \sqcap \exists r.C_n$ and $D \sqsubseteq C_1, \ldots, D \sqsubseteq C_n$, then $\exists r.D \sqsubseteq C$.*

In the proof of decidability of $\mathcal{EL}$-unification, we will make use of the fact that the inverse strict subsumption order is well-founded.

Proposition 1. *There is no infinite sequence $C_0, C_1, C_2, C_3, \ldots$ of $\mathcal{EL}$-concept terms such that $C_0 \sqsubset C_1 \sqsubset C_2 \sqsubset C_3 \sqsubset \cdots$.*

Proof. We define the *role depth* of an $\mathcal{EL}$-concept term C as the maximal nesting of existential restrictions in C. Let n_0 be the role depth of C_0. Since $C_0 \sqsubseteq C_i$ for $i \geq 1$, it is an easy consequence of Corollary 1 that the role depth of C_i is bounded by n_0, and that C_i contains only concept and role names occurring in C_0. In addition, it is known that, for a given natural number n_0 and finite sets of concept names $\mathcal{C}$ and role names $\mathcal{R}$, there are, up to equivalence, only finitely many $\mathcal{EL}$-concept term built using concept names from $\mathcal{C}$ and role names from $\mathcal{R}$ and of a role depth bounded by n_0 [11]. Consequently, there are indices $i < j$ such that $C_i \equiv C_j$. This contradicts our assumption that $C_i \sqsubset C_j$. □

4 An $\mathcal{EL}$-unification problem of type zero

To show that $\mathcal{EL}$ has unification type zero, we exhibit an $\mathcal{EL}$-unification problem that has this type.

Theorem 2. *Let X, Y be variables. The $\mathcal{EL}$-unification problem $\Gamma := \{X \sqcap \exists r.Y \equiv^? \exists r.Y\}$ has unification type zero.*

Proof. It is enough to show that any complete set of unifiers for this problem is redundant, i.e., contains two different unifiers that are comparable w.r.t. the instantiation preorder. Thus, let M be a complete set of unifiers for Γ.

First, note that M must contain a unifier that maps X to an $\mathcal{EL}$-concept term not equivalent to $\top$ or $\exists r.\top$. In fact, consider a substitution τ such that $\tau(X) = \exists r.A$ and $\tau(Y) = A$. Obviously, τ is a unifier of Γ. Thus, M must contain a unifier σ such that $\sigma \lessdot \tau$. In particular, this means that there is a substitution λ such that $\exists r.A = \tau(X) \equiv \lambda(\sigma(X))$. Obviously, $\sigma(X) \equiv \top$ ($\sigma(X) \equiv \exists r.\top$) would imply $\lambda(\sigma(X)) \equiv \top$ ($\lambda(\sigma(X)) \equiv \exists r.\top$), and thus $\exists r.A \equiv \top$ ($\exists r.A \equiv \exists r.\top$), which is, however, not the case.

Thus, let $\sigma \in M$ be such that $\sigma(X) \not\equiv \top$ and $\sigma(X) \not\equiv \exists r.\top$. Without loss of generality, we assume that $C := \sigma(X)$ and $D := \sigma(Y)$ are reduced. Since σ is a unifier of Γ, we have $\exists r.D \sqsubseteq C$. Consequently, Lemma 1 yields that C is of the form $C = \exists r.C_1 \sqcap \ldots \sqcap \exists r.C_n$ where $n \geq 1$, $C_1, \ldots, C_n$ are reduced and pairwise incomparable w.r.t. subsumption, and $D \sqsubseteq C_1, \ldots, D \sqsubseteq C_n$.

We use σ to construct a new unifier $\widehat{\sigma}$ as follows:

$$\begin{aligned}\widehat{\sigma}(X) &:= \exists r.C_1 \sqcap \ldots \sqcap \exists r.C_n \sqcap \exists r.Z\\ \widehat{\sigma}(Y) &:= D \sqcap Z\end{aligned}$$

where Z is a new variable (i.e., one not occurring in C, D). The second part of Lemma 1 implies that $\widehat{\sigma}$ is indeed a unifier of Γ.

Next, we show that $\widehat{\sigma} \lessdot \sigma$. To this purpose, we consider the substitution λ that maps Z to C_1, and does not change any of the other variables. Then we have $\lambda(\widehat{\sigma}(X)) = \exists r.C_1 \sqcap \ldots \sqcap \exists r.C_n \sqcap \exists r.C_1 \equiv \exists r.C_1 \sqcap \ldots \sqcap \exists r.C_n = \sigma(X)$ and $\lambda(\widehat{\sigma}(Y)) = D \sqcap C_1 \equiv D = \sigma(Y)$. Note that the second equivalence holds since we have $D \sqsubseteq C_1$.

Since M is complete, there exists a unifier $\theta \in M$ such that $\theta \lessdot \widehat{\sigma}$. Transitivity of the relation $\lessdot$ thus yields $\theta \lessdot \sigma$. Since σ and θ both belong to M, we have completed the proof of the theorem once we have shown that $\sigma \neq \theta$. Assume to the contrary that $\sigma = \theta$. Then we have $\sigma \lessdot \widehat{\sigma}$, and thus there exists a substitution μ such that $\mu(\sigma(X)) \equiv \widehat{\sigma}(X)$, i.e.,

$$\exists r.\mu(C_1) \sqcap \ldots \sqcap \exists r.\mu(C_n) \equiv \exists r.C_1 \sqcap \ldots \sqcap \exists r.C_n \sqcap \exists r.Z. \tag{1}$$

Recall that the concept terms $C_1, \ldots, C_n$ are reduced and pairwise incomparable w.r.t. subsumption. In addition, since $\sigma(X) = \exists r.C_1 \sqcap \ldots \sqcap \exists r.C_n$ is reduced and not equivalent to $\exists r.\top$, none of the concept terms $C_1, \ldots, C_n$ can be equivalent to $\top$. Finally, Z is a concept name that does not occur in $C_1, \ldots, C_n$. All this implies that $\exists r.C_1 \sqcap \ldots \sqcap \exists r.C_n \sqcap \exists r.Z$ is reduced. Obviously, any reduced form for $\exists r.\mu(C_1) \sqcap \ldots \sqcap \exists r.\mu(C_n)$ is a conjunction of at most n existential restrictions. Thus, Theorem 1 shows that the above equivalence (1) actually cannot hold.

To sum up, we have shown that M contains two distinct unifiers σ, θ such that $\theta \lessdot \sigma$. Since M was an arbitrary complete set of unifiers for Γ, this shows that this unification problem cannot have a minimal complete set of unifiers. □

5 The decision problem

Before we can describe our decision procedure for $\mathcal{EL}$-unification, we must introduce some notation. An $\mathcal{EL}$-concept term is called an *atom* iff it is a concept name (i.e., concept constant or concept variable) or an existential restriction $\exists r.D$. Obviously, any $\mathcal{EL}$-concept term is (equivalent to) a conjunction of atoms, where the empty conjunction is $\top$. The set $At(C)$ of *atoms of an $\mathcal{EL}$-concept term C* is defined inductively: if $C = \top$, then $At(C) := \emptyset$; if C is a concept name, then $At(C) := \{C\}$; if $C = \exists r.D$ then $At(C) := \{C\} \cup At(D)$; if $C = C_1 \sqcap C_2$, then $At(C) := At(C_1) \cup At(C_2)$.

Concept names and existential restrictions $\exists r.D$ where D is a concept name or $\top$ are called *flat atoms*. The $\mathcal{EL}$-unification problem Γ is *flat* iff it only contains equations of the following form:

- $X \equiv^? C$ where X is a variable and C is a non-variable flat atom;
- $X_1 \sqcap \ldots \sqcap X_m \equiv^? Y_1 \sqcap \ldots \sqcap Y_n$ where $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ are variables.

By introducing new concept variables and eliminating $\top$, any $\mathcal{EL}$-unification problem Γ can be transformed in polynomial time into a flat $\mathcal{EL}$-unification problem Γ' such that Γ is solvable iff Γ' is solvable. Thus, we may assume without loss of generality that our input $\mathcal{EL}$-unification problems are flat. Given a flat $\mathcal{EL}$-unification problem $\Gamma = \{C_1 \equiv^? D_1, \ldots, C_n \equiv^? D_n\}$, we call the atoms of $C_1, D_1, \ldots, C_n, D_n$ the *atoms of Γ*.

The unifier σ of Γ is called *reduced* (*ground*) iff, for all concept variables X occurring in Γ, the $\mathcal{EL}$-concept term $\sigma(X)$ is reduced (does not contain variables). Obviously, Γ is solvable iff it has a reduced ground unifier. Given a ground unifier σ of Γ, we consider the set $At(\sigma)$ of all atoms of $\sigma(X)$, where X ranges over all variables occurring in Γ. We call the elements of $At(\sigma)$ the *atoms of σ*.

Given $\mathcal{EL}$-concept terms C, D, we define $C >_{is} D$ iff $C \sqsubset D$. Proposition 1 says that the strict order $>_{is}$ defined this way is well-founded. This order is monotone in the following sense.

Lemma 2. *Let C, D, D' be $\mathcal{EL}$-concept terms such that $D >_{is} D'$ and C is reduced and contains at least one occurrence of D. If C' is obtained from C by replacing all occurrences of D by D', then $C >_{is} C'$.*

Proof. We prove the lemma by induction on the size of C. If $C = D$, then $C' = D'$, and thus $C = D >_{is} D' = C'$. Thus, assume that $C \neq D$. In this case, C obviously cannot be a concept name. If $C = \exists r.C_1$, then D occurs in C_1. By induction, we can assume that $C_1 >_{is} C_1'$, where C_1' is obtained from C_1 by replacing all occurrences of D by D'. Thus, we have $C = \exists r.C_1 >_{is} \exists r.C_1' = C'$ by Corollary 1. Finally, assume that $C = C_1 \sqcap \ldots \sqcap C_n$ for $n > 1$ atoms $C_1, \ldots, C_n$. Since C is reduced, these atoms are incomparable w.r.t. subsumption, and since D occurs in C we can assume without loss of generality that D occurs in C_1. Let $C_1', \ldots, C_n'$ be respectively obtained from $C_1, \ldots, C_n$ by replacing every occurrence of D by D', and then reducing the concept term obtained this way. By induction, we have $C_1 >_{is} C_1'$. Assume that $C \not>_{is} C'$.

Since the concept constructors of $\mathcal{EL}$ are monotone w.r.t. subsumption $\sqsubseteq$, we have $C \sqsubseteq C'$, and thus $C \not>_{is} C'$ means that $C \equiv C'$. Consequently, $C = C_1 \sqcap \ldots \sqcap C_n$ and the reduced form of $C'_1 \sqcap \ldots \sqcap C'_n$ must be equal up to associativity and commutativity of $\sqcap$. If $C'_1 \sqcap \ldots \sqcap C'_n$ is not reduced, then its reduced form is actually a conjunction of $m < n$ atoms, which contradicts $C \equiv C'$. If $C'_1 \sqcap \ldots \sqcap C'_n$ is reduced, then $C_1 >_{is} C'_1$ implies that there is an $i \neq 1$ such that $C_i \equiv C'_1$. However, then $C_i \equiv C'_1 \sqsupset C_1$ contradicts the fact that the atoms $C_1, \ldots, C_n$ are incomparable w.r.t. subsumption. □

We use the order $>_{is}$ on $\mathcal{EL}$-concept terms to define a well-founded order on ground unifiers. Since $>_{is}$ is well-founded, its multiset extension $>_m$ is also well-founded. Given a ground unifier σ of Γ, we consider the multiset $S(\sigma)$ of all $\mathcal{EL}$-concept terms $\sigma(X)$, where X ranges over all concept variables occurring in Γ. For two ground unifiers σ, θ of Γ, we define $\sigma \succ \theta$ iff $S(\sigma) >_m S(\theta)$. The ground unifier σ of Γ is *minimal* iff there is no ground unifier θ of Γ such that $\sigma \succ \theta$. The following proposition is an easy consequence of the fact that $\succ$ is well-founded.

Proposition 2. *Let Γ be an $\mathcal{EL}$-unification problem. Then Γ is solvable iff it has a minimal reduced ground unifier.*

In the following, we show that minimal reduced ground unifiers of flat $\mathcal{EL}$-unification problems satisfy properties that make it easy to check (with an NP-algorithm) whether such a unifier exists or not.

Lemma 3. *Let Γ be a flat $\mathcal{EL}$-unification problem and γ a minimal reduced ground unifier of Γ. If C is an atom of γ, then there is a non-variable atom D of Γ such that $C \equiv \gamma(D)$.*

Proof. Since γ is ground, C is either a concept constant or an existential restriction. First, assume that $C = A$ for a concept constant A, but there is no non-variable atom D of Γ such that $A \equiv \gamma(D)$. This simply means that A does not occur in Γ. Let γ' be the substitution obtained from γ by replacing every occurrence of A by $\top$. Since equivalence in $\mathcal{EL}$ is preserved under replacing concept names by $\top$, and since A does not occur in Γ, it is easy to see that γ' is also a unifier of Γ. However, since $\gamma \succ \gamma'$, this contradicts our assumption that γ is minimal.

Second, assume that $C = \exists r.C_1$, but there is no non-variable atom D of Γ such that $C \equiv \gamma(D)$. We assume that C is maximal (w.r.t. subsumption) with this property, i.e., for every atom C' of γ with $C \sqsubset C'$, there is a non-variable atom D' of Γ such that $C' \equiv \gamma(D')$. Let $D_1, \ldots, D_n$ be all the atoms of Γ with $C \sqsubseteq \gamma(D_i)$ $(i = 1, \ldots, n)$. By our assumptions on C, we actually have $C \sqsubset \gamma(D_i)$ and, by Lemma 1, the atom D_i is also an existential restriction $D_i = \exists r.D'_i$ $(i = 1, \ldots, n)$. The conjunction $\widehat{D} := \gamma(D_1) \sqcap \ldots \sqcap \gamma(D_n)$ obviously subsumes C. We claim that this subsumption relationship is actually strict. In fact, if $n = 0$, then $\widehat{D} = \top$, and since C is an atom, it is not equivalent to $\top$. If $n \geq 1$, then $C = \exists r.C_1 \sqsupseteq \exists r.\gamma(D'_1) \sqcap \ldots \sqcap \exists r.\gamma(D_n)$ would imply (by

Corollary 1) that there is an $i, 1 \leq i \leq n$, with $C_1 \sqsupseteq \gamma(D_i')$. However, this would yield $C = \exists r.C_1 \sqsupseteq \exists r.\gamma(D_i') = \gamma(D_i)$, which contradicts the fact that $C \sqsubset \gamma(D_i)$. Thus, we have shown that $C \sqsubset \widehat{D}$. The substitution γ' is obtained from γ by replacing every occurrence of C by $\widehat{D}$. Lemma 2 implies that $\gamma \succ \gamma'$. Thus, to obtain the desired contradiction, it is sufficient to show that γ' is a unifier of Γ.

First, consider an equation of the form $X \equiv^? E$ in Γ, where X is a variable and E is a non-variable flat atom. If E is a concept constant, then $\gamma(X) = E$, and thus $\gamma'(X) = \gamma(X)$, which shows that γ' solves this equation. Thus, assume that $E = \exists r.E'$. Since γ is reduced, we actually have $\gamma(X) = \exists r.\gamma(E')$. If C occurs in $\gamma(E')$, then each replacement of C by $\widehat{D}$ in $\gamma(E')$ is matched by the corresponding replacement in $\gamma(X)$. Thus, in this case γ' again solves the equation. Finally, assume that $C = \gamma(X)$. But then $C \equiv \gamma(E)$ for a non-variable atom E of Γ, which contradicts our assumption on C

Second, consider an equation of the form $X_1 \sqcap \ldots \sqcap X_m \equiv^? Y_1 \sqcap \ldots \sqcap Y_n$ where $X_1, \ldots, X_m, Y_1, \ldots, Y_n$ are variables. Then $L := \gamma(X_1 \sqcap \ldots \sqcap X_m)$ and $R := \gamma(Y_1 \sqcap \ldots \sqcap Y_n)$ reduce to the same reduced $\mathcal{EL}$-concept term J. Let L', R', J' be the $\mathcal{EL}$-concept terms respectively obtained from L, R, J by replacing every occurrence of C by $\widehat{D}$. We prove that $L' = \gamma'(X_1 \sqcap \ldots \sqcap X_m)$ and $R' = \gamma'(Y_1 \sqcap \ldots \sqcap Y_n)$ both reduce to J', which shows that γ' solves this equation. It is enough to show that the reductions are invariant under the replacement of C by $\widehat{D}$. Obviously, all the interesting reductions are of the form $E_1 \sqcap E_2 \rightarrow E_1$ where E_1, E_2 are existential restrictions such that $E_1 \sqsubseteq E_2$. Since γ is reduced, we can assume that E_1, E_2 are reduced. Let E_1', E_2' be respectively obtained from E_1, E_2 by replacing every occurrence of C by $\widehat{D}$. We must show that $E_1' \sqcap E_2'$ reduces to E_1'. For this, it is enough to show that $E_1' \sqsubseteq E_2'$. Assume that an occurrence of C in E_1 is actually needed to have the subsumption $E_1 \sqsubseteq E_2$. Then there is an existential restriction C' in E_2 such that $C \sqsubseteq C'$. If $C = C'$, then both are replaced by $\widehat{D}$, and thus this replacement is harmless. Otherwise, $C \sqsubset C'$. Since C' is an atom of γ, maximality of C yields that there is a non-variable atom D' of Γ such that $C' \equiv \gamma(D')$. Now $C \sqsubset C' \equiv \gamma(D')$ implies that there is an $i, 1 \leq i \leq n$, such that $D' = D_i$. Thus, C' is actually one of the conjuncts of $\widehat{D}$, which again shows that replacing C by $\widehat{D}$ is harmless. Thus, we have shown that $E_1' \sqsubseteq E_2'$, which completes the proof of the lemma. □

The next proposition is an easy consequence of this lemma.

Proposition 3. *Let Γ be a flat $\mathcal{EL}$-unification problem and γ a minimal reduced ground unifier of Γ. If X is a concept variable occurring in Γ, then $\gamma(X) \equiv \top$ or there are non-variable atoms $D_1, \ldots, D_n$ ($n \geq 1$) of Γ such that $\gamma(X) \equiv \gamma(D_1) \sqcap \ldots \sqcap \gamma(D_n)$.*

Proof. If $\gamma(X) \not\equiv \top$, then it is a non-empty conjunction of atoms, i.e., there are atoms $C_1, \ldots, C_n$ ($n \geq 1$) such that $\gamma(X) = C_1 \sqcap \ldots \sqcap C_n$. Then $C_1, \ldots, C_n$ are atoms of γ, and thus Lemma 3 yields non-variable atoms $D_1, \ldots, D_n$ of Γ such that $C_i \equiv \gamma(D_i)$ for $i = 1, \ldots n$. Consequently, $\gamma(X) \equiv \gamma(D_1) \sqcap \ldots \sqcap \gamma(D_n)$. □

This proposition suggests the following *non-deterministic algorithm for deciding solvability of a given flat $\mathcal{EL}$-unification problem Γ:*

1. For every variable X occurring in Γ, guess a finite, possibly empty, set S_X of non-variable atoms of Γ.
2. We say that the variable X *directly depends on* the variable Y if Y occurs in an atom of S_X. Let *depends on* be the transitive closure of *directly depends on*. If there is a variable that depends on itself, then the algorithm returns "fail." Otherwise, there exists a strict linear order $>$ on the variables occurring in Γ such that $X > Y$ if X depends on Y.
3. We define the substitution σ along the linear order $>$:
   - If X is the least variable w.r.t. $>$, then S_X does not contain any variables. We define $\sigma(X)$ to be the conjunction of the elements of S_X, where the empty conjunction is $\top$.
   - Assume that $\sigma(Y)$ is defined for all variables $Y < X$. Then S_X only contains variables Y for which $\sigma(Y)$ is already defined. If S_X is empty, then we define $\sigma(X) := \top$. Otherwise, let $S_X = \{D_1, \ldots, D_n\}$. We define $\sigma(X) := \sigma(D_1) \sqcap \ldots \sqcap \sigma(D_n)$.
4. Test whether the substitution σ computed in the previous step is a unifier of Γ. If this is the case, then return σ; otherwise, return "fail."

This algorithm is trivially *sound* since it only returns substitutions that are unifiers of Γ. In addition, it obviously always terminates. Thus, to show correctness of our algorithm, it is sufficient to show that it is complete.

Lemma 4 (completeness). *If Γ is solvable, then there is a way of guessing in Step 1 subsets S_X of the non-variable atoms of Γ such that the* depends on *relation determined in Step 2 is acyclic and the substitution σ computed in Step 3 is a unifier of Γ.*

Proof. If Γ is solvable, then it has a minimal reduced ground unifier γ. By Proposition 3, for every variable X occurring in Γ we have $\gamma(X) \equiv \top$ or there are non-variable atoms $D_1, \ldots, D_n$ ($n \geq 1$) of Γ such that $\gamma(X) \equiv \gamma(D_1) \sqcap \ldots \sqcap \gamma(D_n)$. If $\gamma(X) \equiv \top$, then we define $S_X := \emptyset$. Otherwise, we define $S_X := \{D_1, \ldots, D_n\}$.

We show that the relation *depends on* induced by these sets S_X is acyclic, i.e., there is no variable X such that X depends on itself. If X directly depends on Y, then Y occurs in an element of S_X. Since S_X consists of non-variable atoms of the flat unification problem Γ, this means that there is a role name r such that $\exists r.Y \in S_X$. Consequently, we have $\gamma(X) \sqsubseteq \exists r.\gamma(Y)$. Thus, if X depends on X, then there are $k \geq 1$ role names $r_1, \ldots, r_k$ such that $\gamma(X) \sqsubseteq \exists r_1. \cdots \exists r_k.\gamma(X)$. This is clearly not possible since $\gamma(X)$ cannot be subsumed by an $\mathcal{EL}$-concept term whose role depth is larger than the role depth of $\gamma(X)$.

To show that the substitution σ induced by the sets S_X is a unifier of Γ, we prove that σ is equivalent to γ, i.e., $\sigma(X) \equiv \gamma(X)$ holds for all variables X occurring in Γ. The substitution σ is defined along the linear order $>$. If X is the least variable w.r.t. $>$, then S_X does not contain any variables. If S_X is empty, then $\sigma(X) = \top \equiv \gamma(X)$. Otherwise, let $S_X = \{D_1, \ldots, D_n\}$. Since the atoms D_i do not contain variables, we have $D_i = \gamma(D_i)$. Thus, the definitions of S_X and of σ yield $\sigma(X) = D_1 \sqcap \ldots \sqcap D_n = \gamma(D_1) \sqcap \ldots \sqcap \gamma(D_n) \equiv \gamma(X)$.

Assume that $\sigma(Y) \equiv \gamma(Y)$ holds for all variables $Y < X$. If $S_X = \emptyset$, then we have again $\sigma(X) = \top \equiv \gamma(X)$. Otherwise, let $S_X = \{D_1, \ldots, D_n\}$. Since the atoms D_i contain only variables that are smaller than X, we have $\sigma(D_i) \equiv \gamma(D_i)$ by induction. Thus, the definitions of S_X and of σ yield $\sigma(X) = \sigma(D_1) \sqcap \ldots \sqcap \sigma(D_n) \equiv \gamma(D_1) \sqcap \ldots \sqcap \gamma(D_n) \equiv \gamma(X)$. □

Note that our proof of completeness actually shows that, up to equivalence, the algorithm returns all minimal reduced ground unifiers of Γ.

Theorem 3. *$\mathcal{EL}$-unification is NP-complete.*

Proof. NP-hardness follows from the fact that $\mathcal{EL}$-matching is NP-complete [17]. To show that the problem can be decided by a non-deterministic polynomial-time algorithm, we analyse the complexity of our algorithm. Obviously, guessing the sets S_X (Step 1) can be done within NP. Computing the *depends on* relation and checking it for acyclicity (Step 2) is clearly polynomial.

Steps 3 and 4 are more problematic. In fact, since a variable may occur in different atoms of Γ, the substitution σ computed in Step 3 may be of exponential size. This is actually the same reason that makes a naive algorithm for syntactic unification compute an exponentially large most general unifier [12]. As in the case of syntactic unification, the solution to this problem is basically structure sharing. Instead of computing the substitution σ explicitly, we view its definition as an acyclic TBox. To be more precise, for every concept variable X occurring in Γ, the TBox $\mathcal{T}_\sigma$ contains the concept definition $X \doteq \top$ if $S_X = \emptyset$ and $X \doteq D_1 \sqcap \ldots \sqcap D_n$ if $S_X = \{D_1, \ldots, D_n\}$ $(n \geq 1)$. Instead of computing σ in Step 3, we compute $\mathcal{T}_\sigma$. Because of the acyclicity test in Step 2, we know that $\mathcal{T}_\sigma$ is an acyclic TBox. The size of $\mathcal{T}_\sigma$ is obviously polynomial in the size of Γ, and thus this modified Step 3 is polynomial. It is easy to see that applying the substitution σ is the same as expanding the concept terms C, D w.r.t. the TBox $\mathcal{T}_\sigma$. This implies that, for every equation $C \equiv^? D$ in Γ, we have $C \equiv_{\mathcal{T}_\sigma} D$ iff $\sigma(C) \equiv \sigma(D)$. Thus, testing whether σ is a unifier of Γ can be reduced to testing whether $C \equiv_{\mathcal{T}_\sigma} D$ holds for every equation $C \equiv^? D$ in Γ. Since subsumption (and thus equivalence) in $\mathcal{EL}$ w.r.t. acyclic TBoxes can be decided in polynomial time [3],[4] this completes the proof of the theorem. □

6 Unification in semilattices with monotone operators

Unification problems and their types were originally not introduced for Description Logics, but for equational theories [12]. In this section, we show that the above results for unification in $\mathcal{EL}$ can actually be viewed as results for an equational theory. As shown in [19], the equivalence problem for $\mathcal{EL}$-concept terms corresponds to the word problem for the equational theory of semilattices with monotone operators. In order to define this theory, we consider a signature Σ_{SLmO} consisting of a binary function symbol $\wedge$, a constant symbol 1, and finitely many unary function symbols $f_1, \ldots, f_n$. Terms can then be built using these symbols and additional variable symbols and free constant symbols.

[4] Of course, the polynomial-time subsumption algorithm does not expand the TBox.

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

$$SLmO := \{x \wedge (y \wedge z) = (x \wedge y) \wedge z,\ x \wedge y = y \wedge x,\ x \wedge x = x,\ x \wedge 1 = x\} \cup \\ \{f_i(x \wedge y) \wedge f_i(y) = f_i(x \wedge y) \mid 1 \leq i \leq n\}$$

A given $\mathcal{EL}$-concept term C using only roles $r_1, \ldots, r_n$ can be translated into a term t_C over the signature Σ_{SLmO} by replacing each concept constant A by a corresponding free constants a, each concept variable X by a corresponding variable x, $\top$ by 1, $\sqcap$ by $\wedge$, and $\exists r_i$ by f_i. For example, the $\mathcal{EL}$-concept term $C = A \sqcap \exists r_1.\top \sqcap \exists r_3.(X \sqcap B)$ is translated into $t_C = a \wedge f_1(1) \wedge f_3(x \wedge b)$. Conversely, any term over the signature Σ_{SLmO} can be translated back into an $\mathcal{EL}$-concept term.

Lemma 5. *Let C, D be $\mathcal{EL}$-concept term using only roles $r_1, \ldots, r_n$. Then $C \equiv D$ iff $t_C =_{SLmO} t_D$.*

As an immediate consequence of this lemma, we have that unification in the DL $\mathcal{EL}$ corresponds to unification modulo the equational theory $SLmO$. Thus, Theorem 2 implies that $SLmO$ has unification type zero, and Theorem 3 implies that $SLmO$-unification is NP-complete.

Corollary 2. *The equational theory SLmO of semilattices with monotone operators has unification type zero, and deciding solvability of an SLmO-unification problem is an NP-complete problem.*

7 Conclusion

In this paper, we have shown that unification in the DL $\mathcal{EL}$ is of type zero and NP-complete. There are interesting differences between the behaviour of $\mathcal{EL}$ and the closely related DL $\mathcal{FL}_0$ w.r.t. unification and matching. Though the unification types coincide for these two DLs, the complexities of the decision problems differ: $\mathcal{FL}_0$-unification is ExpTime-complete, and thus considerably harder than $\mathcal{EL}$-unification. In contrast, $\mathcal{FL}_0$-matching is polynomial, and thus considerably easier than $\mathcal{EL}$-matching, which is NP-complete.

It is well-known that there is a close connection between modal logics and DLs [5]. For example, the DL $\mathcal{ALC}$, which can be obtained by adding negation to $\mathcal{EL}$ or $\mathcal{FL}_0$, corresponds to the basic (multi-)modal logic K. Decidability of unification in K is a long-standing open problem. Recently, undecidability of unification in some extensions of K (for example, by the universal modality) was shown in [20]. The undecidability results in [20] also imply undecidability of unification in some expressive DLs (e.g., $\mathcal{SHIQ}$). The unification types of some modal (and related) logics have been determined by Ghilardi; for example in [14] he shows that $\mathsf{K4}$ and $\mathsf{S4}$ have unification type finitary. Unification in sub-Boolean modal logics (i.e., modal logics that are not closed under all Boolean operations, such as the modal logic equivalent of $\mathcal{EL}$) has, to the best of our knowledge, not been considered in the modal logic literature.

References

1. F. Baader. Unification in commutative theories. *J. of Symbolic Computation*, 8(5), 1989.
2. F. Baader. Terminological cycles in KL-ONE-based knowledge representation languages. In *Proc. AAAI'90*, 1990.
3. F. Baader. Terminological cycles in a description logic with existential restrictions. In *Proc. IJCAI'03*, 2003.
4. F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proc. IJCAI'05*, 2005.
5. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
6. F. Baader and R. Küsters. Matching in description logics with existential restrictions. In *Proc. KR'00*, 2000.
7. F. Baader and R. Küsters. Unification in a description logic with transitive closure of roles. In *Proc. LPAR'01*, Springer LNAI 2250, 2001.
8. F. Baader, R. Küsters, A. Borgida, and D. L. McGuinness. Matching in description logics. *J. of Logic and Computation*, 9(3), 1999.
9. F. Baader and P. Narendran. Unification of concepts terms in description logics. *J. of Symbolic Computation*, 31(3), 2001.
10. F. Baader and W. Nutt. Basic description logics. In [5], 2003.
11. F. Baader, B. Sertkaya, and A.-Y. Turhan. Computing the least common subsumer w.r.t. a background terminology. *J. of Applied Logic*, 5(3), 2007.
12. Franz Baader and Wayne Snyder. Unification theory. In *Handbook of Automated Reasoning*, volume I. Elsevier Science Publishers, 2001.
13. S. Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In *Proc. ECAI'04*, 2004.
14. S. Ghilardi. Best solving modal equations. *Ann. Pure Appl. Logic*, 102(3), 2000.
15. I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1), 2003.
16. Y. Kazakov and H. de Nivelle. Subsumption of concepts in $\mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete. In *Proc. DL'03*. CEUR Electronic Workshop Proceedings, http://CEUR-WS.org/Vol-81/, 2003.
17. R. Küsters. *Non-standard Inferences in Description Logics*, Springer LNAI 2100, 2001.
18. A. Rector and I. Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proc. AAAI'97*, 1997.
19. V. Sofronie-Stokkermans. Locality and subsumption testing in $\mathcal{EL}$ and some of its extensions. In *Proc. AiML'08*, 2008.
20. F. Wolter and M. Zakharyaschev. Undecidability of the unification and admissibility problems for modal and description logics, *ACM Trans. Comput. Log.*, 9(4), 2008.