Extending Unification in $\mathcal{EL}$ towards General TBoxes*

Franz Baader and **Stefan Borgwardt** and **Barbara Morawska**
Theoretical Computer Science, TU Dresden, Germany
{baader,stefborg,morawska}@tcs.inf.tu-dresden.de

Abstract

Unification in Description Logics (DLs) has been proposed as an inference service that can, for example, be used to detect redundancies in ontologies. The inexpressive Description Logic $\mathcal{EL}$ is of particular interest in this context since, on the one hand, several large biomedical ontologies are defined using $\mathcal{EL}$. On the other hand, unification in $\mathcal{EL}$ has recently been shown to be NP-complete, and thus of significantly lower complexity than unification in other DLs of similarly restricted expressive power. However, the unification algorithms for $\mathcal{EL}$ developed so far cannot deal with general concept inclusion axioms (GCIs). This paper makes a considerable step towards addressing this problem, but the GCIs our new unification algorithm can deal with still need to satisfy a certain cycle restriction.

1 Introduction

The DL $\mathcal{EL}$, which offers the constructors conjunction ($\sqcap$), existential restriction ($\exists r.C$), and the top concept ($\top$), has recently drawn considerable attention since, on the one hand, important inference problems such as the subsumption problem are polynomial in $\mathcal{EL}$, even in the presence of GCIs (Brandt 2004; Baader, Brandt, and Lutz 2005). On the other hand, though quite inexpressive, $\mathcal{EL}$ can be used to define biomedical ontologies, such as the large medical ontology SNOMED CT.[1]

Unification in DLs has been proposed in (Baader and Narendran 2001) (for the DL $\mathcal{FL}_0$, which differs from $\mathcal{EL}$ by offering value restrictions ($\forall r.C$) in place of existential restrictions) as a novel inference service that can, for instance, be used to detect redundancies in ontologies. For example, assume that one developer of a medical ontology defines the concept of a *finding of severe head injury* as

$$\exists \mathsf{finding}.(\mathsf{Head\_injury} \sqcap \exists \mathsf{severity}.\mathsf{Severe}), \quad (1)$$

whereas another one represents it as

$$\exists \mathsf{finding}.(\mathsf{Severe\_injury} \sqcap \exists \mathsf{finding\_site}.\mathsf{Head}). \quad (2)$$

These two concept descriptions are not equivalent, but they are nevertheless meant to represent the same concept. They can obviously be made equivalent by treating the concept names Head_injury and Severe_injury as variables, and substituting the first one by $\mathsf{Injury} \sqcap \exists \mathsf{finding\_site}.\mathsf{Head}$ and the second one by $\mathsf{Injury} \sqcap \exists \mathsf{severity}.\mathsf{Severe}$. In this case, we say that the descriptions are unifiable, and call the substitution that makes them equivalent a *unifier*. Intuitively, such a unifier proposes definitions for the concept names that are used as variables: in our example, we know that, if we define Head_injury as $\mathsf{Injury} \sqcap \exists \mathsf{finding\_site}.\mathsf{Head}$ and Severe_injury as $\mathsf{Injury} \sqcap \exists \mathsf{severity}.\mathsf{Severe}$, then the two concept descriptions (1) and (2) are equivalent w.r.t. these definitions. Here equivalence holds without additional GCIs.

To motivate our interest in unification w.r.t. GCIs, assume that the second developer uses the description

$$\exists \mathsf{status}.\mathsf{Emergency} \sqcap \exists \mathsf{finding}.(\mathsf{Severe\_injury} \sqcap \exists \mathsf{finding\_site}.\mathsf{Head}) \quad (3)$$

instead of (2). The descriptions (1) and (3) are not unifiable without additional GCIs, but they are unifiable, with the same unifier as above, if the GCI

$$\exists \mathsf{finding}.\exists \mathsf{severity}.\mathsf{Severe} \sqsubseteq \exists \mathsf{status}.\mathsf{Emergency}$$

is present in a background ontology.

In (Baader and Morawska 2009), we were able to show that unification in $\mathcal{EL}$ is of considerably lower complexity than in $\mathcal{FL}_0$: the decision problem in $\mathcal{EL}$ is NP-complete rather than EXPTIME-complete in $\mathcal{FL}_0$. In addition to a brute-force "guess and then test" NP-algorithm (Baader and Morawska 2009), we were able to develop a goal-oriented unification algorithm for $\mathcal{EL}$, in which nondeterministic decisions are only made if they are triggered by "unsolved parts" of the unification problem (Baader and Morawska 2010b), and an algorithm that is based on a reduction to satisfiability in propositional logic (SAT) (Baader and Morawska 2010a), which enables the use of highly-optimized SAT solvers. In (Baader and Morawska 2010b) it was also shown that the approaches for unification of $\mathcal{EL}$-concept descriptions (without any background ontology) can easily be extended to the case of an acyclic TBox as background ontology without really changing the algorithms or increasing their complexity. Basically, by viewing defined concepts as variables, an acyclic TBox can be turned into a unification problem that has as its unique unifier the substitution that replaces the defined concepts by unfolded ver-

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

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

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

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

sions of their definitions. For GCIs, this simple trick is not possible.

In the present paper, we extend the brute-force "guess and then test" NP-algorithm from (Baader and Morawska 2009) to the case of GCIs, which requires the development of a new characterization of subsumption w.r.t. GCIs in $\mathcal{EL}$. Unfortunately, the algorithm is complete only for general TBoxes (i.e., finite sets of GCIs) that satisfy a certain restriction on cycles, which, however, does not prevent all cycles. For example, the cyclic GCI $\exists \mathsf{child}.\mathsf{Human} \sqsubseteq \mathsf{Human}$ satisfies this restriction, whereas the cyclic GCI $\mathsf{Human} \sqsubseteq \exists \mathsf{parent}.\mathsf{Human}$ does not.

Due to space constraints, we cannot present and prove all our results in detail here. Full proofs and a goal-oriented algorithm for unification in $\mathcal{EL}$ w.r.t. cycle-restricted general TBoxes can be found in (Baader, Borgwardt, and Morawska 2011).

2 The Description Logic $\mathcal{EL}$

Starting with a finite set N_C of *concept names* and a finite set N_R of *role names*, *$\mathcal{EL}$-concept descriptions* are built using the concept constructors *top-concept* ($\top$), *conjunction* ($C \sqcap D$), and *existential restriction* ($\exists r.C$ for every $r \in N_R$). *Nested existential restrictions* $\exists r_1.\exists r_2.\cdots\exists r_n.C$ will sometimes also be written as $\exists r_1 r_2 \ldots r_n.C$, where $r_1 r_2 \ldots r_n$ is viewed as a word over the alphabet of role names, i.e., an element of N_R^*.

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a nonempty domain $\Delta^{\mathcal{I}}$ and an interpretation function $\cdot^{\mathcal{I}}$ that assigns binary relations on $\Delta^{\mathcal{I}}$ to role names and subsets of $\Delta^{\mathcal{I}}$ to concept descriptions, as shown in the semantics column of Table 1.

A *general concept inclusion (GCI)* is of the form $C \sqsubseteq D$ for concept descriptions C, D, and a general TBox is a finite set of GCIs. An interpretation $\mathcal{I}$ *satisfies* such a GCI if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, and it is a *model* of the general TBox $\mathcal{T}$ if it satisfies all GCIs in $\mathcal{T}$.

Subsumption asks whether a given GCI $C \sqsubseteq D$ follows from a general TBox $\mathcal{T}$, i.e. whether every model of $\mathcal{T}$ satisfies $C \sqsubseteq D$. In this case we say C is *subsumed* by D w.r.t. $\mathcal{T}$ and write $C \sqsubseteq_{\mathcal{T}} D$. Subsumption in $\mathcal{EL}$ w.r.t. a general TBox is known to be decidable in polynomial time (Brandt 2004). Our unification algorithm will use a polynomial-time subsumption algorithm as a subprocedure. In order to develop the unification algorithm itself, however, we need a structural characterization of subsumption w.r.t. a general TBox. Before we can present this characterization, we need to introduce some new notions.

An $\mathcal{EL}$-concept description is an *atom* if it is an existential restriction or a concept name. The atoms of an $\mathcal{EL}$-concept description C are the subdescriptions of C that are atoms, and the top-level atoms of C are the atoms occurring in the top-level conjunction of C. Obviously, any $\mathcal{EL}$-concept description is the conjunction of its top-level atoms, where the empty conjunction corresponds to $\top$. The atoms of a general TBox $\mathcal{T}$ are the atoms of all the concept descriptions occurring in $\mathcal{T}$.

We say that a subsumption between two atoms is *structural* if their top-level structure is compatible. To be more precise, we define structural subsumption between atoms as follows: the atom C is *structurally subsumed* by the atom D w.r.t. $\mathcal{T}$ ($C \sqsubseteq^{\mathsf{s}}_{\mathcal{T}} D$) iff either

- $C = D$ is a concept name, or
- $C = \exists r.C'$, $D = \exists r.D'$, and $C' \sqsubseteq_{\mathcal{T}} D'$.

It is easy to see that subsumption w.r.t. $\emptyset$ between two atoms implies structural subsumption w.r.t. $\mathcal{T}$, which in turn implies subsumption w.r.t. $\mathcal{T}$. The unification algorithm presented in this paper crucially depends on the following characterization of subsumption w.r.t. general TBoxes:

Lemma 1. *Let $\mathcal{T}$ be a general TBox and $C_1, \ldots, C_n$, $D_1, \ldots, D_m$ atoms. Then $C_1 \sqcap \cdots \sqcap C_n \sqsubseteq_{\mathcal{T}} D_1 \sqcap \cdots \sqcap D_m$ iff for every $j \in \{1, \ldots, m\}$*

1. *there is an index $i \in \{1, \ldots, n\}$ such that $C_i \sqsubseteq^{\mathsf{s}}_{\mathcal{T}} D_j$, or*
2. *there are atoms $A_1, \ldots, A_k, B$ of $\mathcal{T}$ ($k \geq 0$) such that*
   a) *$A_1 \sqcap \cdots \sqcap A_k \sqsubseteq_{\mathcal{T}} B$,*
   b) *for every $\eta \in \{1, \ldots, k\}$ there is $i \in \{1, \ldots, n\}$ with $C_i \sqsubseteq^{\mathsf{s}}_{\mathcal{T}} A_\eta$, and*
   c) *$B \sqsubseteq^{\mathsf{s}}_{\mathcal{T}} D_j$.*

Our proof of this lemma in (Baader, Borgwardt, and Morawska 2011) is based on a new Gentzen-style proof calculus for subsumption w.r.t. a general TBox, which is similar to the one developed in (Hofmann 2005) for subsumption w.r.t. cyclic and general TBoxes.

As mentioned in the introduction, our unification algorithm is complete only for general TBoxes that satisfy a certain restriction on cycles.

Definition 2. The general TBox $\mathcal{T}$ is called *cycle-restricted* iff there is no nonempty word $w \in N_R^+$ and $\mathcal{EL}$-concept description C such that $C \sqsubseteq_{\mathcal{T}} \exists w.C$.

In (Baader, Borgwardt, and Morawska 2011) we show that a given general TBox can easily be tested for cycle-restrictedness. The main idea is that it is sufficient to consider the cases where C is a concept name or $\top$.

Lemma 3. *Let $\mathcal{T}$ be a general TBox. It can be decided in time polynomial in the size of $\mathcal{T}$ whether $\mathcal{T}$ is cycle-restricted or not.*

3 Unification in $\mathcal{EL}$ w.r.t. General TBoxes

We partition the set N_C 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). A *substitution* σ maps every concept variable to an $\mathcal{EL}$-concept description. It can be extended to concept descriptions in the usual 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 $\mathcal{EL}$-concept description C is *ground* if it does not contain variables. Obviously, a ground concept description is not modified by applying a substitution. A general TBox is *ground* if it does not contain variables.

Definition 4. Let $\mathcal{T}$ be a general TBox that is ground. An *$\mathcal{EL}$-unification problem* w.r.t. $\mathcal{T}$ is a finite set $\Gamma = \{C_1 \sqsubseteq^? D_1, \ldots, C_n \sqsubseteq^? D_n\}$ of subsumptions between $\mathcal{EL}$-concept descriptions. A substitution σ is a *unifier* of Γ w.r.t. $\mathcal{T}$ if σ *solves* all the subsumptions in Γ, i.e., if $\sigma(C_1) \sqsubseteq_{\mathcal{T}} \sigma(D_1), \ldots, \sigma(C_n) \sqsubseteq_{\mathcal{T}} \sigma(D_n)$. We say that Γ is *unifiable* w.r.t. $\mathcal{T}$ if it has a unifier.

Two remarks regarding this definition are in order. First, note that the previous papers on unification in DLs used equivalences $C \equiv^? D$ instead of subsumptions $C \sqsubseteq^? D$. This difference is, however, irrelevant since $C \equiv^? D$ can be seen as a shorthand for the two subsumptions $C \sqsubseteq^? D$ and $D \sqsubseteq^? C$, and $C \sqsubseteq^? D$ has the same unifiers as $C \sqcap D \equiv^? C$.

Second, note that we have restricted the background general TBox $\mathcal{T}$ to be ground. This is not without loss of generality. In fact, if $\mathcal{T}$ contained variables, then we would need to apply the substitution also to its GCIs, and instead of requiring $\sigma(C_i) \sqsubseteq_{\mathcal{T}} \sigma(D_i)$ we would thus need to require $\sigma(C_i) \sqsubseteq_{\sigma(\mathcal{T})} \sigma(D_i)$, which would change the nature of the problem considerably. The treatment of unification w.r.t. acyclic TBoxes in (Baader and Morawska 2010b) actually considers a more general setting, where some of the primitive concepts occurring in the TBox may be variables. The restriction to ground general TBoxes is, however, appropriate for the application scenario sketched in the introduction. In this scenario, there is a fixed background ontology, given by a general TBox, which is extended with definitions of new concepts by several knowledge engineers. Unification w.r.t. the background ontology is used to check whether some of these new definitions actually are redundant, i.e., define the same intuitive concept. Here, some of the primitive concepts newly introduced by one knowledge engineer may be further defined by another one, but we assume that the knowledge engineers use the vocabulary from the background ontology unchanged, i.e., they define *new* concepts rather than adding definitions for concepts that already occur in the background ontology. An instance of this scenario can, e.g., be found in (Campbell et al. 2007), where different extensions of SNOMED CT are checked for overlaps, albeit not by using unification, but by simply testing for equivalence.

In the remainder of this section we will show that $\mathcal{EL}$-unification w.r.t. cycle-restricted TBoxes is NP-complete. NP-hardness is an immediate consequence of the fact that $\mathcal{EL}$-unification is NP-complete w.r.t. the empty TBox (Baader and Morawska 2009). Thus, it is enough to show that $\mathcal{EL}$-unification is still in NP w.r.t. cycle-restricted TBoxes.

Preprocessing To simplify the description of the NP-algorithm, it is convenient to first normalize the TBox and the unification problem appropriately.

An atom is called *flat* if it is a concept name or an existential restriction of the form $\exists r.A$ for a concept name A. The general TBox $\mathcal{T}$ is called *flat* if it contains only GCIs of the form $A \sqcap B \sqsubseteq C$, where A, B are flat atoms or $\top$ and C is a flat atom. The unification problem Γ is called *flat* if it contains only flat subsumptions of the form $C_1 \sqcap \cdots \sqcap C_n \sqsubseteq^? D$, where $n \geq 0$ and $C_1, \ldots, C_n, D$ are flat atoms.[2]

Let Γ be a unification problem and $\mathcal{T}$ a general TBox. By introducing auxiliary variables and concept names, respectively, Γ and $\mathcal{T}$ can be transformed in polynomial time into a flat unification problem Γ' and a flat general TBox $\mathcal{T}'$ such that the unifiability status remains unchanged, i.e., Γ has a unifier w.r.t. $\mathcal{T}$ iff Γ' has a unifier w.r.t. $\mathcal{T}'$. In addition, if $\mathcal{T}$ was cycle-restricted, then so is $\mathcal{T}'$ (see (Baader, Borgwardt, and Morawska 2011) for details). Thus, we can assume without loss of generality that the input unification problem and general TBox are flat.

Local Unifiers The main idea underlying the "in NP" result in (Baader and Morawska 2009) is to show that any $\mathcal{EL}$-unification problem that is unifiable w.r.t. the empty TBox has a so-called local unifier. Here, we generalize the notion of a local unifier to the case of unification w.r.t. cycle-restricted TBoxes, and show that a similar locality result holds in this case.

Let $\mathcal{T}$ be a flat cycle-restricted TBox and Γ a flat unification problem. The atoms of Γ are the atoms of all the concept descriptions occurring in Γ. We define

$$\begin{aligned} \mathrm{At} &:= \{C \mid C \text{ is an atom of } \mathcal{T} \text{ or of } \Gamma\} \text{ and} \\ \mathrm{At}_{\mathsf{nv}} &:= \mathrm{At} \setminus N_v \quad \text{(non-variable atoms)}. \end{aligned}$$

Every assignment S of subsets S_X of $\mathrm{At}_{\mathsf{nv}}$ to the variables X in N_v induces the following relation $>_S$ on N_v: $>_S$ is the transitive closure of

$$\{(X, Y) \in N_v \times N_v \mid Y \text{ occurs in an element of } S_X\}.$$

We call the assignment S *acyclic* if $>_S$ is irreflexive (and thus a strict 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 element of N_v w.r.t. $>_S$, then we define $\sigma_S(X) := \bigsqcap_{D \in S_X} D$.
- Assume that $\sigma(Y)$ is already defined for all Y such that $X >_S Y$. Then we define $\sigma_S(X) := \bigsqcap_{D \in S_X} \sigma_S(D)$.

We call a substitution σ *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 Γ w.r.t. $\mathcal{T}$ is a local substitution, then we call it a *local unifier* of Γ w.r.t. $\mathcal{T}$.

Theorem 5. *Let $\mathcal{T}$ be a flat cycle-restricted TBox and Γ a flat unification problem. If Γ has a unifier w.r.t. $\mathcal{T}$, then it also has a local unifier w.r.t. $\mathcal{T}$.*

[2] If $n = 0$, then we have an empty conjunction on the left-hand side, which as usual stands for $\top$.

This theorem immediately implies that unification in $\mathcal{EL}$ w.r.t. cycle-restricted TBoxes is decidable within NP. In fact, one can guess an acyclic assignment S in polynomial time. To check whether the induced local substitution σ_S is a unifier of Γ w.r.t. $\mathcal{T}$, we build the general TBox

$$\mathcal{T}_S := \{X \sqsubseteq \prod_{D \in S_X} D,\ \prod_{D \in S_X} D \sqsubseteq X \mid X \in N_v\},$$

and then check in polynomial time whether $C \sqsubseteq_{\mathcal{T} \cup \mathcal{T}_S} D$ holds for all $C \sqsubseteq^? D \in \Gamma$. It is easy to show that this is the case iff $\sigma_S(C) \sqsubseteq_{\mathcal{T}} \sigma_S(D)$ for all $C \sqsubseteq^? D \in \Gamma$.

Corollary 6. *Unification in $\mathcal{EL}$ w.r.t. cycle-restricted TBoxes is in NP.*

Proof of Theorem 5 Assume that γ is a unifier of Γ w.r.t. $\mathcal{T}$. We define the assignment S^γ induced by γ as

$$S^\gamma_X := \{D \in \mathrm{At}_{\mathsf{nv}} \mid \gamma(X) \sqsubseteq_{\mathcal{T}} \gamma(D)\}.$$

The following lemma is the only place in the proof of Theorem 5 where cycle-restrictedness of $\mathcal{T}$ is needed. Later we will give an example (Example 9) that demonstrates that the theorem actually does not hold if this restriction is removed.

Lemma 7. *The assignment S^γ is acyclic.*

Proof. Assume that S^γ is cyclic. Then there are variables $X_1, \ldots, X_n$ and role names $r_1, \ldots, r_{n-1}$ ($n \geq 2$) such that $X_1 = X_n$ and $\exists r_i.X_{i+1} \in S^\gamma(X_i)$ ($i = 1, \ldots, n-1$). But then we have $\gamma(X_i) \sqsubseteq_{\mathcal{T}} \exists r_i.\gamma(X_{i+1})$ for $i = 1, \ldots, n-1$, which yields $\gamma(X_1) \sqsubseteq_{\mathcal{T}} \exists r_1.\gamma(X_2) \sqsubseteq_{\mathcal{T}} \exists r_1.\exists r_2.\gamma(X_3) \sqsubseteq_{\mathcal{T}} \cdots \sqsubseteq_{\mathcal{T}} \exists r_1.\cdots\exists r_{n-1}.\gamma(X_n)$. Since $X_1 = X_n$ and $n \geq 2$, this contradicts our assumption that $\mathcal{T}$ is cycle-restricted. Thus, S^γ must be acyclic. □

Since S^γ is acyclic, it induces a substitution σ_{S^γ}. To simplify the notation, we call this substitution in the following σ^γ. The following lemma implies that σ^γ is a unifier of Γ w.r.t. $\mathcal{T}$, and thus proves Theorem 5.

Lemma 8. *Let $C_1, \ldots, C_n, D \in \mathrm{At}$. Then $\gamma(C_1) \sqcap \ldots \sqcap \gamma(C_n) \sqsubseteq_{\mathcal{T}} \gamma(D)$ implies $\sigma^\gamma(C_1) \sqcap \ldots \sqcap \sigma^\gamma(C_n) \sqsubseteq_{\mathcal{T}} \sigma^\gamma(D)$.*

Proof. We prove the lemma by induction over

$$max\{rd(\sigma^\gamma(E)) \mid E \in \{C_1, \ldots, C_n, D\} \wedge E \text{ not ground}\},$$

where the *role depth* $rd(C)$ of a concept description C is defined as follows: $rd(A) = rd(\top) = 0$ for $A \in N_C$, $rd(C \sqcap D) = \max\{rd(C), rd(D)\}$, $rd(\exists r.C) = 1 + rd(C)$.

First, assume that $D = Y \in N_v$, and let $S^\gamma_Y = \{D_1, \ldots, D_m\}$. By the definition of S^γ, this implies $\gamma(Y) \sqsubseteq_{\mathcal{T}} \gamma(D_1) \sqcap \ldots \sqcap \gamma(D_m)$, and thus

$$\gamma(C_1) \sqcap \ldots \sqcap \gamma(C_n) \sqsubseteq_{\mathcal{T}} \gamma(D_1) \sqcap \ldots \sqcap \gamma(D_m).$$

We apply Lemma 1 to this subsumption. Consider $\gamma(D_j)$ for some $j, 1 \leq j \leq m$. Since D_j is a non-variable atom, $\gamma(D_j)$ is an atom, and thus the first or the second case of the lemma holds.

1. In the first case, there is an $i, 1 \leq i \leq n$, such that one of the following two cases holds:
   (i) *C_i is a non-variable atom and $\gamma(C_i) \sqsubseteq^s_{\mathcal{T}} \gamma(D_j)$.*
   By the definition of $\sqsubseteq^s_{\mathcal{T}}$, there are two possible cases. Either both concept descriptions are the same concept name A, or both are existential restrictions for the same role name r. In the first case, $C_i = A = D_j$, and thus $\sigma^\gamma(C_i) = A = \sigma^\gamma(D_j)$. In the second case, $C_i = \exists r.C'_i$, $D_j = \exists r.D'_j$, and $\gamma(C'_i) \sqsubseteq_{\mathcal{T}} \gamma(D'_j)$. Both C'_i and D'_j are elements of At. The role depth of $\sigma^\gamma(C'_i)$ is obviously smaller than the role depth of $\sigma^\gamma(C_i)$. For the same reason, the role depth of $\sigma^\gamma(D'_j)$ is smaller that the one of $\sigma^\gamma(D_j)$. Since $\sigma^\gamma(D_j)$ is a top-level conjunct in $\sigma^\gamma(D)$, the role depth of $\sigma^\gamma(D'_j)$ is also smaller than the one of $\sigma^\gamma(D)$. Consequently, if C_i or D_j is non-ground, induction yields $\sigma^\gamma(C'_i) \sqsubseteq_{\mathcal{T}} \sigma^\gamma(D'_j)$, and thus also $\sigma^\gamma(C_i) = \exists r.\sigma^\gamma(C'_i) \sqsubseteq_{\mathcal{T}} \exists r.\sigma^\gamma(D'_j) = \sigma^\gamma(D_j)$. If C_i, D_j are both ground, then $\sigma^\gamma(C_i) = C_i = \gamma(C_i) \sqsubseteq_{\mathcal{T}} \gamma(D_j) = D_j = \sigma^\gamma(D_j)$.
   (ii) *$C_i = X$ is a variable and the top-level conjunction of $\gamma(X)$ contains an atom E such that $E \sqsubseteq^s_{\mathcal{T}} \gamma(D_j)$.*
   Then we have $\gamma(X) \sqsubseteq_{\mathcal{T}} E \sqsubseteq_{\mathcal{T}} \gamma(D_j)$, and thus $D_j \in S^\gamma_X$. By the definition of σ^γ, this implies $\sigma^\gamma(C_i) = \sigma^\gamma(X) \sqsubseteq_{\mathcal{T}} \sigma^\gamma(D_j)$.

   Both (i) and (ii) yield $\sigma^\gamma(C_1) \sqcap \ldots \sqcap \sigma^\gamma(C_n) \sqsubseteq_{\mathcal{T}} \sigma^\gamma(D_j)$.
2. In the second case, there are atoms $A_1, \ldots, A_k, B$ of $\mathcal{T}$ such that
   a) $A_1 \sqcap \ldots \sqcap A_k \sqsubseteq_{\mathcal{T}} B$,
   b) for every $\eta, 1 \leq \eta \leq k$, there is $i, 1 \leq i \leq n$, such that one of the following two cases holds:
      (i) C_i is a non-variable atom and $\gamma(C_i) \sqsubseteq^s_{\mathcal{T}} A_\eta$,
      (ii) $C_i = X$ is a variable and the top-level conjunction of $\gamma(X)$ contains an atom E such that $E \sqsubseteq^s_{\mathcal{T}} A_\eta$;
   c) $B \sqsubseteq^s_{\mathcal{T}} \gamma(D_j)$.

   In case (i) we have that either $C_i = A = A_\eta$ is a concept name, or both concept descriptions are existential restrictions $C_i = \exists r.C'_i$ and $A_\eta = \exists r.A'_\eta$ with $\gamma(C'_i) \sqsubseteq_{\mathcal{T}} A'_\eta$. In the first case, we have $\sigma^\gamma(C_i) = A = A_\eta$. In the second case, the case where C_i is ground is again trivial. Otherwise, we can apply induction since $C'_i, A'_\eta \in \mathrm{At}$, the role depth of $\sigma^\gamma(C'_i)$ is smaller than the one of $\sigma^\gamma(C_i)$, and the role depth of A'_η is not counted since it is ground. Thus, we have $\sigma^\gamma(C'_i) \sqsubseteq_{\mathcal{T}} A'_\eta$, which yields $\sigma^\gamma(C_i) \sqsubseteq_{\mathcal{T}} A_\eta$.

   In case (ii), we again have $\gamma(X) \sqsubseteq_{\mathcal{T}} E \sqsubseteq_{\mathcal{T}} A_\eta$, and thus $A_\eta \in S^\gamma_X$. This yields $\sigma^\gamma(C_i) = \sigma^\gamma(X) \sqsubseteq_{\mathcal{T}} A_\eta$.

   For similar reasons as before, we can again show that $B \sqsubseteq^s_{\mathcal{T}} \gamma(D_j)$ implies $B \sqsubseteq_{\mathcal{T}} \sigma^\gamma(D_j)$.

   To sum up, we thus have also in this case $\sigma^\gamma(C_1) \sqcap \ldots \sqcap \sigma^\gamma(C_n) \sqsubseteq_{\mathcal{T}} A_1 \sqcap \ldots \sqcap A_k \sqsubseteq_{\mathcal{T}} B \sqsubseteq_{\mathcal{T}} \sigma^\gamma(D_j)$.

Hence, we have shown that, for all $j, 1 \leq j \leq m$, we have $\sigma^\gamma(C_1) \sqcap \ldots \sqcap \sigma^\gamma(C_n) \sqsubseteq_{\mathcal{T}} \sigma^\gamma(D_j)$, which yields $\sigma^\gamma(C_1) \sqcap \ldots \sqcap \sigma^\gamma(C_n) \sqsubseteq_{\mathcal{T}} \sigma^\gamma(D_1) \sqcap \ldots \sqcap \sigma^\gamma(D_m) = \sigma(D)$. The last identity holds since $D = Y$ and $S^\gamma_Y = \{D_1, \ldots, D_m\}$.

It remains to consider the case where D is a non-variable atom. But then we have

$$\gamma(C_1) \sqcap \ldots \sqcap \gamma(C_n) \sqsubseteq_{\mathcal{T}} \gamma(D),$$

and $\gamma(D)$ is an atom. As for $\gamma(D_j)$ above, we can use Lemma 1 to show that this implies $\sigma^\gamma(C_1) \sqcap \ldots \sqcap \sigma^\gamma(C_n) \sqsubseteq_\mathcal{T} \sigma^\gamma(D)$. □

Example 9 (Cycle-restrictedness is needed)**.** We show that Theorem 5 does not hold for arbitrary general TBoxes. To this purpose, consider the general TBox $\mathcal{T} = \{B \sqsubseteq \exists s.D,\ D \sqsubseteq B\}$, which is not cycle-restricted, and the unification problem

$$\Gamma = \{A_1 \sqcap B \equiv^? Y_1,\ A_2 \sqcap B \equiv^? Y_2,\ \exists s.Y_1 \sqsubseteq^? X,\ \exists s.Y_2 \sqsubseteq^? X,\ X \sqsubseteq^? \exists s.X\}.$$

This problem has the unifier $\gamma := \{Y_1 \mapsto A_1 \sqcap B, Y_2 \mapsto A_2 \sqcap B, X \mapsto \exists s.B\}$. However, the induced assignment S^γ is cyclic since $\gamma(X) = \exists s.B \sqsubseteq_\mathcal{T} \exists s.\exists s.B = \gamma(\exists s.X)$ yields $\exists s.X \in S^\gamma_X$. Thus, γ does not induce a local unifier.

We claim that Γ actually does not have any local unifier w.r.t. $\mathcal{T}$. Assume to the contrary that σ is a local unifier of Γ w.r.t. $\mathcal{T}$. Then $\sigma(X)$ cannot be $\top$ since $\top \not\sqsubseteq_\mathcal{T} \exists s.\top$. Thus, $\sigma(X)$ must contain a top-level atom of the form $\sigma(E)$ for $E \in \mathrm{At}_{\mathsf{nv}}$. This atom cannot be $\sigma(\exists s.Y_i) \equiv_\mathcal{T} \exists s.(A_i \sqcap B)$ for $i \in \{1,2\}$ since then $\sigma(\exists s.Y_j) \sqsubseteq_\mathcal{T} \sigma(E)$ for $j \in \{1,2\} \setminus \{i\}$ would not hold, contradicting the assumption that σ solves $\exists s.Y_j \sqsubseteq^? X$ w.r.t. $\mathcal{T}$. Since local unifiers are induced by acyclic assignments, E cannot be $\sigma(\exists s.X)$, and thus E must be an atom of $\mathcal{T}$. However, none of the atoms $B, D, \exists s.D$ subsume $\exists s.(A_j \sqcap B)$ w.r.t. $\mathcal{T}$, again contradicting the assumption that σ solves $\exists s.Y_j \sqsubseteq^? X$ w.r.t. $\mathcal{T}$.

4 Conclusions

We have shown that unification in $\mathcal{EL}$ stays in NP in the presence of a cycle-restricted general TBox, by giving a brute-force NP-algorithm that tries to guess a local unifier. This algorithm is interesting since it provides a quite simple, self-contained proof for the complexity upper-bound. Indeed, it is much simpler than the original proof (Baader and Morawska 2009; 2010b) of the NP-upper bound for $\mathcal{EL}$ without TBoxes.

In (Baader, Borgwardt, and Morawska 2011), we also introduce a goal-oriented algorithm for unification in $\mathcal{EL}$ in the presence of a cycle-restricted TBox, in which nondeterministic decisions are only made if they are triggered by "unsolved parts" of the unification problem. Another advantage of the goal-oriented algorithm is that it only generates substitutions that are unifiers, whereas the brute-force algorithm generates all local substitutions, and requires a subsequent test of whether this substitution is a unifier. Nevertheless, this algorithms still requires a considerable amount of additional optimization work to be useful in practice.

On the theoretical side, the main topic for future research is to consider unification w.r.t. unrestricted general TBoxes. In order to generalize the brute-force algorithm in this direction, we need to find a more general notion of locality. Starting with the goal-oriented algorithm (Baader, Borgwardt, and Morawska 2011), the idea would be not to fail when a cyclic assignment is generated, but rather to add rules that can break such cycles, similar to what is done in procedures for general E-unification (Morawska 2007).

References

Baader, F., and Morawska, B. 2009. Unification in the description logic $\mathcal{EL}$. In Treinen, R., ed., *Proc. of the 20th Int. Conf. on Rewriting Techniques and Applications (RTA 2009)*, volume 5595 of *Lecture Notes in Computer Science*, 350–364. Springer-Verlag.

Baader, F., and Morawska, B. 2010a. SAT encoding of unification in $\mathcal{EL}$. In Fermüller, C. G., and Voronkov, A., eds., *Proc. of the 17th Int. Conf. on Logic for Programming, Artifical Intelligence, and Reasoning (LPAR-17)*, volume 6397 of *Lecture Notes in Computer Science*, 97–111. Springer-Verlag.

Baader, F., and Morawska, B. 2010b. Unification in the description logic $\mathcal{EL}$. *Logical Methods in Computer Science* 6(3). Special Issue: 20th Int. Conf. on Rewriting Techniques and Applications (RTA'09).

Baader, F., and Narendran, P. 2001. Unification of concept terms in description logics. *J. of Symbolic Computation* 31(3):277–305.

Baader, F.; Borgwardt, S.; and Morawska, B. 2011. Unification in the description logic $\mathcal{EL}$ w.r.t. cycle-restricted TBoxes. LTCS-Report 11-05, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany. See http://lat.inf.tu-dresden.de/research/reports.html.

Baader, F.; Brandt, S.; and Lutz, C. 2005. Pushing the $\mathcal{EL}$ envelope. In Kaelbling, L. P., and Saffiotti, A., eds., *Proc. of the 19th Int. Joint Conf. on Artificial Intelligence (IJCAI 2005)*, 364–369. Morgan-Kaufmann Publishers.

Brandt, S. 2004. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In de Mántaras, R. L., and Saitta, L., eds., *Proc. of the 16th Eur. Conf. on Artificial Intelligence (ECAI 2004)*, 298–302. IOS Press.

Campbell, J. R.; Lopez Osornio, A.; de Quiros, F.; Luna, D.; and Reynoso, G. 2007. Semantic interoperability and SNOMED CT: A case study in clinical problem lists. In Kuhn, K.; Warren, J.; and Leong, T.-Y., eds., *Proc. of the 12th World Congress on Health (Medical) Informatics (MEDINFO 2007)*, 2401–2402. IOS Press.

Hofmann, M. 2005. Proof-theoretic approach to description-logic. In *Proc. of the 20th IEEE Symp. on Logic in Computer Science (LICS 2005)*, 229–237.

Morawska, B. 2007. General E-unification with eager variable elimination and a nice cycle rule. *J. of Automated Reasoning* 39(1):77–106.