Complexity of Subsumption in the $\mathcal{EL}$ Family of Description Logics: Acyclic and Cyclic TBoxes

Christoph Haase[1] and **Carsten Lutz**[2]

Abstract. We perform an exhaustive study of the complexity of subsumption in the $\mathcal{EL}$ family of lightweight description logics w.r.t. acyclic and cyclic TBoxes. It turns out that there are interesting members of this family for which subsumption w.r.t. cyclic TBoxes is tractable, whereas it is EXPTIME-complete w.r.t. general TBoxes. For other extensions that are intractable w.r.t. general TBoxes, we establish intractability already for acyclic and cyclic TBoxes.

1 MOTIVATION

Description logics (DLs) are a popular family of KR languages that can be used for the formulation of and reasoning about ontologies [5]. Traditionally, the DL research community has strived for identifying more and more expressive DLs for which reasoning is still decidable. In recent years, however, there have been two lines of development that have led to significant popularity also of DLs with limited expressive power. First, a number of novel and useful lightweight DLs with tractable reasoning problems has been identified, see e.g. [3, 8]. And second, many large-scale ontologies that are formulated in such lightweight DLs have emerged from practical applications. Prominent examples include the Systematized Nomenclature of Medicine, Clinical Terms (SNOMED CT), which underlies the systematized medical terminology used in the health systems of the US, the UK, and other countries [19]; and the gene ontology (GO), which aims at consistent descriptions of gene products in different databases [20].

In this paper, we are concerned with the $\mathcal{EL}$ family of lightweight DLs, which consists of the basic DL $\mathcal{EL}$ and its extensions. Members of this family underly many large-scale ontologies including SNOMED CT and GO. The DL counterpart of an ontology is called a TBox, and the most important reasoning task in DLs is subsumption. In particular, computing subsumption allows to classify the concepts defined in the TBox/ontology according to their generality [5]. In the DL literature, different kinds of TBoxes have been considered. In decreasing order of expressive power, the most common ones are general TBoxes, (potentially) cyclic TBoxes, and acyclic TBoxes. For the $\mathcal{EL}$ family, the complexity of subsumption w.r.t. general TBoxes has exhaustively been analyzed in [3] and its recent successor [4]. In all of the considered cases, subsumption is either tractable or EXPTIME-complete. However, the study of general TBoxes does not reflect common practice of ontology design, as most ontologies from practical applications correspond to cyclic or acyclic TBoxes. For example, SNOMED CT and GO both correspond to so-called *acyclic* TBoxes. Since cyclic and acyclic TBoxes are often preferable in terms of computational complexity [7, 14], the question arises whether there are useful extensions of $\mathcal{EL}$ for which reasoning w.r.t. such TBoxes is computationally cheaper than reasoning w.r.t. general TBoxes.

The goal of the current paper is *to analyse the computational complexity of subsumption in the $\mathcal{EL}$ family of description logics w.r.t. acyclic TBoxes and cyclic TBoxes, with a special emphasis on the border of tractability.* In our analysis, we omit extensions of $\mathcal{EL}$ for which tractability w.r.t. general TBoxes has already been established. Our results exhibit a more varied complexity landscape than in the case of general TBoxes: we identify cases in which reasoning is tractable, co-NP-complete, PSPACE-complete, and EXPTIME-complete. Notably, we identify two maximal extensions of $\mathcal{EL}$ for which subsumption w.r.t. cyclic TBoxes is tractable, whereas it is EXPTIME-complete w.r.t. general TBoxes. In particular, these extensions include primitive negation and at-least restrictions. They also include concrete domains, but fortunately do not require the strong convexity condition that was needed in the case of general TBoxes to guarantee tractability [3]. For other extensions of $\mathcal{EL}$ such as inverse roles and functional roles, we show intractability results already w.r.t. acyclic TBoxes. Compared to the case of general TBoxes, it is often necessary to develop new approaches to lower bound proofs. We also show that the union of the two identified tractable fragments is not tractable. Detailed proofs are provided in [10].

2 DESCRIPTION LOGICS

The two types of expressions in a DL are *concepts* and *roles*, which are built inductively starting from infinite sets $\mathsf{N_C}$ and $\mathsf{N_R}$ of *concept names* and *role names*, and applying *concept constructors* and *role constructors*. The basic description logic $\mathcal{EL}$ provides the concept constructors top ($\top$), conjunction ($C \sqcap D$) and existential restriction ($\exists r.C$), and no role constructors. Here and in what follows, we denote the elements of $\mathsf{N_C}$ with A and B, the elements of $\mathsf{N_R}$ with r and s, and concepts with C and D. The semantics of concepts and roles is given in terms of an *interpretation* $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, with $\Delta^\mathcal{I}$ a non-empty set called the *domain* and $\cdot^\mathcal{I}$ the *interpretation function*, which maps every $A \in \mathsf{N_C}$ to a subset $A^\mathcal{I}$ of $\Delta^\mathcal{I}$ and every role name r to binary relation $r^\mathcal{I}$ of over $\Delta^\mathcal{I}$.

Extensions of $\mathcal{EL}$ are characterized by the additional concept and role constructors that they offer. Figure 1 lists all relevant constructors, concept constructors in the upper part and role constructors in the lower part. The left column gives the syntax, and the right column shows how to inductively extend interpretations to composite concepts and roles. In the presence of role constructors, composite roles can be used inside existential restrictions. In *atleast restrictions* ($\geq n\, r$) and *atmost restrictions* ($\leq n\, r$) , we use n to denote a non-negative integer. The concrete domain constructor $p(f_1, \ldots, f_k)$ de-

[1] University of Oxford, UK, christoph.haase@comlab.ox.ac.uk
[2] TU Dresden, Germany, lutz@tcs.inf.tu-dresden.de

Syntax	Semantics
$\top$	$\Delta^{\mathcal{I}}$
$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$(\leq n\, r)$	$\{x \mid \#\{y \mid (x,y) \in r^{\mathcal{I}}\} \leq n\}$
$(\geq n\, r)$	$\{x \mid \#\{y \mid (x,y) \in r^{\mathcal{I}}\} \geq n\}$
$\exists r.C$	$\{x \mid \exists y : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
$\forall r.C$	$\{x \mid \forall y : (x,y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
$p(f_1, \ldots, f_k)$	$\{x \mid \exists d_1, \ldots, d_k : f_1^{\mathcal{I}}(x) = d_1 \wedge \ldots \wedge f_k^{\mathcal{I}}(x) = d_k \wedge (d_1, \ldots, d_k) \in p^{\mathcal{D}}\}$
$r \cap s$	$r^{\mathcal{I}} \cap s^{\mathcal{I}}$
$r \cup s$	$r^{\mathcal{I}} \cup s^{\mathcal{I}}$
r^-	$\{(x,y) \mid (y,x) \in r^{\mathcal{I}}\}$
r^+	$\bigcup_{i>0} (r^{\mathcal{I}})^i$

Figure 1. Syntax and semantics of concept and role constructors.

serves further explanation, to be given below. To denote extensions of $\mathcal{EL}$, we use the symbol of the added constructors in superscript. For example, $\mathcal{EL}^{\sqcup,\cup,-}$ denotes the extension of $\mathcal{EL}$ with *concept disjunction* ($C \sqcup D$), *role disjunction* ($r \cup s$), and *inverse roles* (r^-).

The concrete domain constructor permits reference to concrete data objects such as strings and integers. It provides the interface to a *concrete domain* $\mathcal{D} = (\Delta_{\mathcal{D}}, \Phi_{\mathcal{D}})$, which consists of a domain $\Delta_{\mathcal{D}}$ and a set of predicates $\Phi_{\mathcal{D}}$ [13]. Each $p \in \Phi_{\mathcal{D}}$ is associated with a fixed arity n and a fixed extension $p^{\mathcal{D}} \subseteq \Delta_{\mathcal{D}}^n$. In the presence of a concrete domain $\mathcal{D}$, we assume that there is an infinite set $\mathsf{N_F}$ of *feature names* disjoint from $\mathsf{N_R}$ and $\mathsf{N_C}$. In Figure 1 and in general, $f_1, \ldots, f_k$ are from $\mathsf{N_F}$ and $p \in \Phi_{\mathcal{D}}$. An interpretation $\mathcal{I}$ maps every $f \in \mathsf{N_F}$ to a partial function $f^{\mathcal{I}}$ from $\Delta^{\mathcal{I}}$ to $\Delta_{\mathcal{D}}$. We use $\mathcal{EL}(\mathcal{D})$ to denote the extension of $\mathcal{EL}$ with the concrete domain $\mathcal{D}$.

In this paper, a *TBox* $\mathcal{T}$ is a finite set of *concept definitions* $A \equiv C$, where $A \in \mathsf{N_C}$ and C is a concept. We require that the left-hand side of all concept definitions in a TBox are unique. A concept name $A \in \mathsf{N_C}$ is *defined* if it occurs on the left-hand side of a concept definition in $\mathcal{T}$, and *primitive* otherwise. A TBox $\mathcal{T}$ is *acyclic* if there are no concept definitions $A_1 \equiv C_1, \ldots, A_k \equiv C_k \in \mathcal{T}$ such that A_{i+1} occurs in C_i for $1 \leq i \leq k$, where $A_{k+1} := A_1$. An interpretation $\mathcal{I}$ is a *model* of $\mathcal{T}$ iff $A^{\mathcal{I}} = C^{\mathcal{I}}$ for all $A \equiv C \in \mathcal{T}$.

The main reasoning task considered in this paper is subsumption. A concept C is *subsumed* by a concept D w.r.t. a TBox $\mathcal{T}$, written $\mathcal{T} \models C \sqsubseteq D$, if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$. If $\mathcal{T}$ is empty or missing, we simply write $C \sqsubseteq D$. Sometimes, we also consider satisfiability of concepts. A concept C is *satisfiable* w.r.t. a TBox $\mathcal{T}$ if there is a model of $\mathcal{T}$ such that $C^{\mathcal{I}} \neq \emptyset$. For many extensions of $\mathcal{EL}$, satisfiability is trivial because there are no unsatisfiable concepts.

3 TRACTABLE EXTENSIONS

We identify two extensions of $\mathcal{EL}$ for which subsumption w.r.t. TBoxes is tractable: $\mathcal{EL}^{\cup,(\neg)}(\mathcal{D})$ and $\mathcal{EL}^{\geq,\cup}$. This should be contrasted with the results in [3] which imply that subsumption w.r.t. general TBoxes is ExpTime-complete in both extensions. In Section 4.1, we show that taking the union of the two extensions results in intractability already w.r.t. acyclic TBoxes.

(C1)	$L_{\mathcal{T}}(B) \subseteq L_{\mathcal{T}}(A)$
(C2)	For each $\exists r_B.B' \in E_{\mathcal{T}}(B)$ there is $\exists r_A.A' \in E_{\mathcal{T}}(A)$ such that $r_A \subseteq r_B$ and $(A', B') \in S$
(C3)	$\mathsf{Con}_{\mathcal{D}}(A)$ implies $\mathsf{Con}_{\mathcal{D}}(B)$

Figure 2. $\mathcal{EL}^{\cup,(\neg)}(\mathcal{D})$: Conditions for adding (A, B) to S.

3.1 Role Disjunction, Primitive Negation, and Concrete Domains

We show that subsumption in $\mathcal{EL}^{\cup,(\neg)}(\mathcal{D})$ w.r.t. (acyclic and cyclic) TBoxes is tractable. The superscript $\cdot^{(\neg)}$ indicates *primitive* negation, i.e., negation can only be applied to concept names. The following is an example of an $\mathcal{EL}^{\cup,(\neg)}(\mathcal{D})$-TBox, where $\mathsf{has\_age}$ is a feature, and $\geq_{13}$ and $\leq_{19}$ are unary predicates of the concrete domain $\mathcal{D}$:

$$\begin{aligned} \mathsf{Parent} &\equiv \mathsf{Human} \sqcap \exists(\mathsf{has\_child} \cup \mathsf{has\_adopted}).\top \\ \mathsf{Mother} &\equiv \mathsf{Parent} \sqcap \mathsf{Female} \sqcap \neg\mathsf{Male} \\ \mathsf{Teenager} &\equiv \mathsf{Human} \sqcap {\geq_{13}}(\mathsf{has\_age}) \sqcap {\leq_{19}}(\mathsf{has\_age}) \end{aligned}$$

To guarantee tractability, we require the concrete domain $\mathcal{D}$ to satisfy a standard condition. Namely, we require $\mathcal{D}$ to be *p-admissibile*, i.e., satisfiability of and implication between *concrete domain expressions* of the form $p_1(v_1^1, \ldots, v_{n_1}^1) \wedge \cdots \wedge p_m(v_1^m, \ldots, v_{n_m}^m)$ are decidable in polynomial time, where the v_j^i are variables that range over $\Delta_{\mathcal{D}}$. In [3], it is shown that a much stronger condition is required to achieve tractability in $\mathcal{EL}(\mathcal{D})$ with general TBoxes. This condition is *convexity*, which requires that if a concrete domain atom $p(v_1, \ldots, v_n)$ implies a disjunction of such atoms, then it implies one of the disjuncts. For our result, there is no need to impose convexity.

When deciding subsumption, we only consider concept names instead of composite concepts. This is sufficient since $\mathcal{T} \models C \sqsubseteq D$ iff $\mathcal{T}' \models A \sqsubseteq B$, where $\mathcal{T}' := \mathcal{T} \cup \{A \equiv C, B \equiv D\}$ and A and B do not occur in $\mathcal{T}$.

The subsumption algorithm requires the input TBox $\mathcal{T}$ to be in the following *normal form*. In each $A \equiv C \in \mathcal{T}$, C is of the form

$$\bigsqcap_{1 \leq i \leq k} L_i \sqcap \bigsqcap_{1 \leq i \leq \ell} \exists r_i.B_i \sqcap \bigsqcap_{1 \leq i \leq m} p_i(f_1^i, \ldots, f_{n_i}^i)$$

where the L_i are *primitive literals*, i.e., possibly negated primitive concept names; the r_i are of the form $r_1 \cup \ldots \cup r_n$; and the B_i are defined concept names. In the following, we refer to the set of literals occurring in C with $L_{\mathcal{T}}(A)$, to the set of existential restrictions as $E_{\mathcal{T}}(A)$, and define the following concrete domain expression, which for simplicity uses features as variables:

$$\mathsf{Con}_{\mathcal{D}}(A) := p_1(f_1^1, \ldots, f_{n_1}^1) \wedge \cdots \wedge p_m(f_1^m, \ldots, f_{n_m}^m).$$

To ease notation, we confuse a role $r_i = r_1 \cup \ldots \cup r_n$ with the set $\{r_1, \ldots, r_n\}$.

It is easy to see how to adapt the algorithm given in [2] to convert an $\mathcal{EL}^{\cup,(\neg)}(\mathcal{D})$-TBox into normal form in quadratic time. During the normalization, we check for unsatisfiable concepts. This is easy since a defined concept name A with $A \equiv C \in \mathcal{T}$ is unsatisfiable w.r.t. $\mathcal{T}$ iff one of the following three conditions holds: (i) there is a primitive concept P with $\{P, \neg P\} \in L_{\mathcal{T}}(A)$; (ii) $\mathsf{Con}_{\mathcal{D}}(A)$ is unsatisfiable; or (iii) there is an $\exists r.B \in E_{\mathcal{T}}(A)$ with B unsatisfiable.

Suppose we want to decide whether A is subsumed by B w.r.t. a TBox $\mathcal{T}$ in normal form. If A is unsatisfiable, the algorithm answers

(C1) $P_\mathcal{T}(B) \subseteq P_\mathcal{T}(A)$

(C2) For each $\exists r_B.B' \in E_\mathcal{T}(B)$ there is $\exists r_A.A' \in E_\mathcal{T}(A)$ such that $r_A \subseteq r_B$ and $(A', B') \in S$

(C3) For each $(\geq m\, r) \in N_\mathcal{T}(B)$, there is $(\geq n\, r) \in N_\mathcal{T}(A)$ such that $n \geq m$.

Figure 3. $\mathcal{EL}^{\geq,\cup}$: Conditions for adding (A, B) to S.

"yes". Otherwise and if B is unsatisfiable, it answers "no". If A and B are both satisfiable, it computes a binary relation S on the defined concept names of $\mathcal{T}$. The relation S is initialized with the identity relation and then completed by exhaustively adding pairs (A, B) for which the conditions in Figure 2 are satisfied.

It is easily seen that the algorithm runs in time polynomial w.r.t. the size of the input TBox. Let $S_0, \ldots, S_n$ be the sequence of relations that it produces. To show soundness, it suffices to prove that if $(A, B) \in S_i$, $i \leq n$, then $\mathcal{T} \models A \sqsubseteq B$. This is straightforward by induction on i. To prove completeness, we have to exhibit a model $\mathcal{I}$ of $\mathcal{T}$ with $A^\mathcal{I} \setminus B^\mathcal{I} \neq \emptyset$. Such a model is constructed in a two-step process. First, we start with an instance of A, and then "apply" the concept definitions in the TBox as implications from left to right, constructing a potentially infinite, tree-shaped interpretation. In the second step, we apply the concept definitions from right to left, filling up the interpretation of defined concepts. Both steps involve some careful bookkeeping which ensures that the constructed instance of A is not an instance of B.

Theorem 1 *Subsumption in $\mathcal{EL}^{\cup,(\neg)}(\mathcal{D})$ w.r.t. TBoxes is in* PTIME.

This result still holds if we additionally allow role conjunction ($r \cap s$) and require that composite roles are in disjunctive normal form (without DNF, subsumption becomes co-NP-hard).It is worth mentioning that, in the presence of general TBoxes, extending $\mathcal{EL}$ with each single one of (i) primitive negation, (ii) role disjunction, and (iii) *any* non-convex concrete domain results in EXPTIME-hardness [3]. Note that convexity of a concrete domain is a rather strong restriction, and it is pleasant that we do not need it to achieve tractability. We point out that it should be possible to enhance the expressive power of $\mathcal{EL}^{\cup,(\neg)}(\mathcal{D})$ by enriching it with additional constructors of the DL $\mathcal{EL}^{++}$ [3]. Examples include nominals and transitive roles.

3.2 Role Disjunction and At-Least Restrictions

In $\mathcal{EL}^{\geq,\cup}$, we allow role disjunction only in existential restrictions, but not in number restrictions. To show that subsumption w.r.t. TBoxes is tractable, we use a variation of the algorithm in the previous section. In the following, we only list the differences. A TBox is in *normal form* if, in each $A \equiv C \in \mathcal{T}$, C is of the form

$$\bigsqcap_{1\leq i\leq k} P_i \sqcap \bigsqcap_{1\leq i\leq \ell} \exists r_i.B_i \sqcap \bigsqcap_{1\leq i\leq m} (\geq n_i\, s_i)$$

where the P_i are primitive concept names, the r_i are of the form $r_1 \cup \ldots \cup r_n$, the B_i are defined concept names, and the s_i are role names. We use $P_\mathcal{T}(A)$ to refer to the set of primitive concept names occurring in C, $E_\mathcal{T}(A)$ is as in the previous section, and $N_\mathcal{T}(A)$ is the set of number restrictions in C. The conditions for adding a pair (A, B) to the relation S are given in Figure 3.

Theorem 2 *Subsumption in $\mathcal{EL}^{\geq,\cup}$ w.r.t. TBoxes is in* PTIME.

In the extension of $\mathcal{EL}$ with only at-least restrictions $(\geq n\, r)$, subsumption w.r.t. general TBoxes is EXPTIME-complete [3]. As we will show in Section 4.3, $\mathcal{EL}$ extended with at-most restrictions $(\leq n\, r)$ is intractable already w.r.t. acyclic TBoxes.

4 INTRACTABLE EXTENSIONS

We identify extensions of $\mathcal{EL}$ for which subsumption is intractable w.r.t. acyclic and cyclic TBoxes.

4.1 Primitive Negation and At-Least Restrictions

We show that taking the union of the DLs $\mathcal{EL}^{\cup,(\neg)}(\mathcal{D})$ and $\mathcal{EL}^{\geq,\cup}$ from Sections 3.1 and 3.2 results in intractability. To this end, we consider $\mathcal{EL}^{\geq,(\neg)}$ and show that subsumption w.r.t. the empty TBox is CO-NP-complete. It is easy to establish the lower bound also for $\mathcal{EL}^{\geq}(\mathcal{D})$ as long as there are two concepts $p(f_1, \ldots, f_n)$ and $p'(f'_1, \ldots, f'_m)$ that are mutually exclusive. This is the case for most practically useful concrete domains $\mathcal{D}$.

For the lower bound, we reduce 3-colorability of graphs to non-subsumption. Given an undirected graph $G = (V, E)$, reserve one concept name P_v for each node $v \in V$, and a single role name r. Then, G is 3-colorable iff $C_G \not\sqsubseteq (\geq 4\, r)$, where

$$C_G := \bigsqcap_{v\in V} \exists r. \left(P_v \sqcap \bigsqcap_{\{v,w\}\in E} \neg P_w \right)$$

Intuitively, if $d \in C_G^\mathcal{I} \setminus (\geq 4\, r)^\mathcal{I}$, then d has at most three r-successors, each describing one of the three colors. The use of primitive negation in C_G ensures that no two adjacent nodes have the same color.

A matching upper bound can be derived from the CO-NP-upper bound for subsumption in $\mathcal{ALUN}$, which has the concept constructors top, bottom ($\bot$), value restriction ($\forall r.C$), conjunction, disjunction, primitive negation, number restrictions, and unqualified existential restriction [11]. Given two $\mathcal{EL}^{\geq,(\neg)}$-concepts C, D, we have $C \sqsubseteq D$ iff $\neg D \sqsubseteq \neg C$. It remains to observe that bringing $\neg C$ and $\neg D$ into negation normal form yields two $\mathcal{ALUN}$-concepts.

Theorem 3 *Subsumption in $\mathcal{EL}^{\geq,(\neg)}$ is* CO-NP-*complete.*

4.2 Inverse Roles

In [1], it is shown that subsumption w.r.t. the empty TBox is tractable in (an extension of) $\mathcal{EL}^-$. We prove that, w.r.t. acyclic TBoxes, subsumption in $\mathcal{EL}^-$ is PSPACE-complete. Since the upper bound follows from PSPACE-completeness of subsumption in $\mathcal{ALCI}$ [5], we concentrate on the lower bound.

We reduce validity of quantified Boolean formulas (QBFs). Let $\varphi = Q_1 v_1 \cdots Q_k v_k.\psi$ be a QBF, where $Q_i \in \{\forall, \exists\}$ for $1 \leq i \leq k$. W.l.o.g., we may assume that $\psi = c_1 \wedge \cdots \wedge c_n$ is in conjunctive normal form. We construct an acyclic TBox $\mathcal{T}_\varphi$ and select two concept names L_0 and E_0 such that φ is valid iff $\mathcal{T}_\varphi \models L_0 \sqsubseteq E_0$. Intuitively, a model of L_0 and $\mathcal{T}_\varphi$ is a binary tree of depth k that is used to evaluate φ. In the tree, a transition from a node at level i to its left successor corresponds to setting v_{i+1} to false, and a transition to the right successor corresponds to setting v_{i+1} to true. Thus, each node on level i corresponds to a truth assignment to the variables $v_1, \ldots, v_i$. In $\mathcal{T}_\varphi$, we use a single role name r and the following concept names:

- $L_0, \ldots, L_k$ represent the level of nodes in the tree model;

- $C_{i,j}$, $1 \le i \le n$ and $1 \le j \le k$, represents truth of the clause c_i on level j of the tree model;
- $E_0, \ldots, E_k$ are used for evaluating ψ, and the index again refers to the level.

For $1 \le i \le k$, we use P_j to denote the conjunction of all concept names $C_{i,j}$, $1 \le i \le n$, such that v_j occurs positively in c_i; similarly, N_j denotes the conjunction of all concept names $C_{i,j}$, $1 \le i \le n$, such that v_j occurs negatively in c_i. Now, the TBox $\mathcal{T}_\varphi$ is as follows:

$$\begin{array}{rcl} L_0 & \equiv & \exists r.(L_1 \sqcap P_1) \sqcap \exists r.(L_1 \sqcap N_1) \\ & \cdots & \\ L_{k-1} & \equiv & \exists r.(L_k \sqcap P_k) \sqcap \exists r.(L_k \sqcap N_k) \\ C_{i,j} & \equiv & \exists r^-.C_{i,j-1} \text{ for } 1 \le i \le n \text{ and } 1 < j \le k \\ E_k & \equiv & C_{1,k} \sqcap \cdots \sqcap C_{n,k} \\ E_i & \equiv & \exists r.E_{i+1} \text{ for } 0 \le i < k \text{ where } Q_{i+1} = \exists \\ E_i & \equiv & \exists r.(P_{i+1} \sqcap E_{i+1}) \sqcap \exists r.(N_{i+1} \sqcap E_{i+1}) \\ & & \text{for } 0 \le i < k \text{ where } Q_{i+1} = \forall \end{array}$$

The definitions for $L_0, \ldots, L_{k-1}$ build up the tree. The use of P_1 and N_1 in these definitions together with the definition of $C_{i,j}$ sets the truth value of the clause c_i according to a partial truth assignment of length j. Finally, the definitions of $E_0, \ldots, E_k$ evaluate φ according to its matrix formula ψ and quantifier prefix. It can be checked that φ is valid iff $\mathcal{T}_\varphi \models L_0 \sqsubseteq E_0$.

Theorem 4 *Subsumption in $\mathcal{EL}^-$ w.r.t. acyclic TBoxes is* PSPACE-*complete.*

We leave the case of cyclic TBoxes as an open problem. In this case, the lower bound from Theorem 4 is complemented only by the EXPTIME upper bound for subsumption in $\mathcal{EL}^-$ w.r.t. general TBoxes from [3].

4.3 Functional Roles

Let $\mathcal{EL}^f$ be $\mathcal{EL}$ extended with functional roles, i.e., there is a countably infinite subset $\mathsf{N_F} \subseteq \mathsf{N_R}$ such that all elements of $\mathsf{N_F}$ are interpreted as partial functions. It is shown in [3] that subsumption in $\mathcal{EL}^f$ w.r.t. general TBoxes is EXPTIME-complete. We show that it is co-NP-complete w.r.t. acyclic TBoxes and PSPACE-complete w.r.t. cyclic ones.

We use $\mathcal{EL}^F$ to denote the variation of $\mathcal{EL}^f$ in which *all* role names are interpreted as partial functions. It has been observed in [3] that there is a close connection between $\mathcal{EL}^F$ and $\mathcal{FL}_0$, which provides the concept constructors conjunction and value restriction. It is easy to exploit this connection to transfer the known co-NP-hardness (PSPACE-hardness) from subsumption in $\mathcal{FL}_0$ w.r.t. acyclic (cyclic) TBoxes as proved in [16, 12] to $\mathcal{EL}^F$. We omit details for brevity. Since the described approach is not very illuminating regarding the source of intractability, however, we give a dedicated proof of co-NP-hardness of subsumption in $\mathcal{EL}^F$ w.r.t. acyclic TBoxes using a reduction from 3-SAT to *non*-subsumption.

Let $\varphi = c_1 \wedge \ldots \wedge c_k$ be a 3-formula in the propositional variables $p_1, \ldots, p_n$ and with $c_j = \ell_1^j \vee \ell_2^j \vee \ell_3^j$ for $1 \le j \le k$. We construct a TBox $\mathcal{T}_\varphi$ and select concept names A_φ and B_1 such that φ is satisfiable iff $\mathcal{T}_\varphi \not\models A_\varphi \sqsubseteq B_1$. In the reduction, we use two role names r_0 and r_1 to represent falsity and truth of variables. More precisely, a path $r_{v_1} \cdots r_{v_n}$ with $r_{v_i} \in \{r_0, r_1\}$ corresponds to the valuation $p_i \mapsto v_i, 1 \le i \le n$. Additionally, we use a number of auxiliary concept names. The TBox $\mathcal{T}_\varphi$ is as follows:

$$\begin{array}{rcl} A_i^j & \equiv & \begin{cases} \exists r_0.A_{i+1}^j & \text{if } p_i \in \{\ell_1^j, \ell_2^j, \ell_3^j\} \\ \exists r_1.A_{i+1}^j & \text{if } \neg p_i \in \{\ell_1^j, \ell_2^j, \ell_3^j\} \\ \exists r_0.A_{i+1}^j \sqcap \exists r_1.A_{i+1}^j & \text{otherwise} \end{cases} \\ A_{n+1}^j & \equiv & \top \\ A_\varphi & \equiv & \bigsqcap_{1 \le j \le k} A_1^j \\ B_i & \equiv & \exists r_0.B_{i+1} \sqcap \exists r_1.B_{i+1} \qquad B_{n+1} \equiv \top \end{array}$$

If $\mathcal{I}$ is a model of $\mathcal{T}_\varphi$ and $d \in (A_1^j)^\mathcal{I}$, $1 \le j \le k$, then d is the root of a tree in $\mathcal{I}$ whose edges are labelled with r_0 and r_1 and whose paths are the valuations that make the clause c_j false. Due to functionality of r_0 and r_1, each $d \in A_\varphi^\mathcal{I}$ is thus the root of a (single) tree whose paths are precisely the valuations that make *any* clause in φ false. Finally, $d \in B_1^\mathcal{I}$ means that d is the root of a full binary tree of depth n whose paths describe *all* valuations. It follows that φ is satisfiable iff $\mathcal{T}_\varphi \not\models A_\varphi \sqsubseteq B_1$.

To prove matching upper bounds for $\mathcal{EL}^f$, we exploit the fact that, due to the $\mathcal{FL}_0$-connection, subsumption in $\mathcal{EL}^F$ is easily shown to be in CO-NP w.r.t. acyclic TBoxes and in PSPACE w.r.t. cyclic ones. We give an algorithm for subsumption in $\mathcal{EL}^f$ that uses subsumption in $\mathcal{EL}^F$ as a subprocedure. Like the algorithms in Section 3, it computes a binary relation S on the set of defined concept names by repeatedly adding pairs (A, B) such that the input TBox entails $A \sqsubseteq B$. The algorithm works for both acyclic and cyclic TBoxes, giving us the desired upper bound in both cases.

We assume the input TBox $\mathcal{T}$ to be in the same normal form as described in Section 3.2, but without concepts of the form $(\geq n\, r)$. Let S be a binary relation on the defined concept names in $\mathcal{T}$. For every concept $\exists r.A$ occurring in $\mathcal{T}$ with $r \notin \mathsf{N_F}$, introduce a fresh concept name $X_{r,A}$ such that $X_{r,A} = X_{r',A'}$ iff $r = r'$, $(A, A') \in S$, and $(A', A) \in S$. Now let the $\mathcal{EL}^F$-TBox $\mathcal{T}_S$ be obtained from $\mathcal{T}$ by (i) replacing every concept $\exists r.A$ where $r \notin \mathsf{N_F}$ with $X_{r,A}$, and (ii) for each $\exists r.A$ in $\mathcal{T}$ with $r \notin \mathsf{N_F}$, adding the concept definition

$$X_{r.A} \equiv X_{r,B_1} \sqcap \cdots \sqcap X_{r,B_n} \sqcap Z_{r,A}$$

where $B_1, \ldots, B_n$ are all concept names with $(A, B_i) \in S$ and $(B_i, A) \notin S$; and $Z_{r,A}$ is a fresh concept name. The algorithm starts with S as the identity relation and then exhaustively performs the following step: add (A, B) to S if $\mathcal{T}_S \models A \sqsubseteq B$. It returns "yes" if the input concepts form a pair in S, and "no" otherwise. Additionally, we can show that subsumption in $\mathcal{EL}^f$ without TBoxes is in PTIME by a reduction to subsumption in $\mathcal{EL}$.

Theorem 5 *Subsumption in $\mathcal{EL}^f$ is in* PTIME*,* CO-NP-*complete w.r.t. acyclic TBoxes and* PSPACE-*complete w.r.t. cyclic TBoxes.*

It is not hard to see that the lower bounds carry over to $\mathcal{EL}^\le$.

4.4 Booleans

We consider extensions of $\mathcal{EL}$ with Boolean constructors, starting with negation. Since $\mathcal{EL}^\neg$ is a notational variant of $\mathcal{ALC}$, we obtain the following from the results in [17, 18].

Theorem 6 *Satisfiability and subsumption in $\mathcal{EL}^\neg$ is* PSPACE-*complete without TBoxes and w.r.t. acyclic TBoxes, and* EXPTIME-*complete w.r.t. cyclic TBoxes.*

Now for disjunction. It has been shown in [6] that subsumption in $\mathcal{EL}^\sqcup$ is CO-NP-complete without TBoxes. In order to establish lower bounds for subsumption w.r.t. TBoxes, we reduce *satisfiability* in $\mathcal{EL}^\neg$ to non-subsumption in $\mathcal{EL}^\sqcup$. An $\mathcal{EL}^\neg$-TBox $\mathcal{T}$ is in *normal form* if for each $A \equiv C \in \mathcal{T}$, C is of the form $\top$, P, $\neg B$, $\exists r.B$, or $B_1 \sqcap B_2$ with P primitive and B, B_1, B_2 defined. It is straightforward to show that any $\mathcal{EL}^\neg$-TBox $\mathcal{T}$ can be transformed into normal form in linear time such that all (non-)subsumptions are preserved. Thus, let $\mathcal{T} = \{A_1 \equiv C_1, \ldots, A_n \equiv C_n\}$ be an $\mathcal{EL}^\neg$-TBox in normal form. Since the proofs underlying Theorem 6 use only a single role name, we may assume w.l.o.g. that $\mathcal{T}$ contains only a single role name r. We convert $\mathcal{T}$ into an $\mathcal{EL}^\sqcup$-TBox $\mathcal{T}'$ by introducing fresh concept names $\overline{A}_1, \ldots, \overline{A}_n$ representing the negations of $A_1, \ldots, A_n$ and replacing every $A \equiv \neg A_j \in \mathcal{T}$ with $A \equiv \overline{A}_j$ and every $A_i \equiv \exists r.A_j \in \mathcal{T}$ with

$$A_i \equiv \exists r.(A_j \sqcap \bigsqcap_{1 \leq k \leq n} (A_k \sqcup \overline{A}_k)).$$

The additional conjunct ensures that A_i and $\overline{A}_i$ cover the domain. To additionally ensure that they are disjoint, we add to $\mathcal{T}'$ the concept definition

$$M \equiv \bigsqcup_{0 \leq i < n} \bigsqcup_{1 \leq j \leq n} \underbrace{\exists r. \ldots \exists r.}_{i \text{ times}}(\overline{A}_j \sqcap A_j) \quad \text{if } \mathcal{T} \text{ is acyclic}$$

$$M \equiv \exists r.M \sqcup \bigsqcup_{1 \leq i \leq n} (\overline{A}_i \sqcap A_i) \quad \text{if } \mathcal{T} \text{ is cyclic.}$$

In both cases, M is a fresh concept name. Then a defined concept name A is satisfiable w.r.t. $\mathcal{T}$ iff $\mathcal{T}' \not\models A \sqcap \bigsqcap_{1 \leq i \leq n} (A_i \sqcup \overline{A}_i) \sqsubseteq M$. We obtain the following result.

Theorem 7 *Subsumption in* $\mathcal{EL}^\sqcup$ *is* PSPACE*-complete w.r.t. acyclic TBoxes and* EXPTIME*-complete w.r.t. cyclic TBoxes.*

4.5 Transitive Closure

We consider $\mathcal{EL}^+$, the extension of $\mathcal{EL}$ with transitive closure of roles. Using a result by Miklau and Suciu on query containment in a fragment of XPath [15], it is easy to show that subsumption in $\mathcal{EL}^+$ is co-NP-complete. By reusing the techniques from Miklau and Suciu's lower bound proof, we can establish PSPACE-hardness (EXPTIME-hardness) of subsumption in $\mathcal{EL}^+$ w.r.t. acyclic (cyclic) TBoxes. More precisely, this is achieved by a reduction of satisfiability in $\mathcal{EL}^\neg$ to non-subsumption in $\mathcal{EL}^+$, similar to the one described in Section 4.4. A corresponding EXPTIME upper bound for the case of cyclic TBoxes is obtained by a straightforward reduction to satisfiability in propositional dynamic logic (PDL). For acyclic TBoxes, we obtain a PSPACE upper bound by a less straightforward reduction to subsumption in $\mathcal{EL}^\sqcup$ w.r.t. acyclic TBoxes, c.f. Theorem 7.

Theorem 8 *Subsumption in* $\mathcal{EL}^+$ *is* CO-NP*-complete,* PSPACE*-complete w.r.t. acyclic TBoxes and* EXPTIME*-complete w.r.t. cyclic ones.*

5 CONCLUSION

The complexity landscape for acyclic/cyclic TBoxes is much less uniform than for general TBoxes. For the case of general TBoxes, non-existence of a unique minimal model of a TBox (in the sense that it can be homomorphically embedded into any other model) was a sufficient (but not necessary) condition for intractability. This is not the case here: in $\mathcal{EL}^{\sqcup,(\neg)}(\mathcal{D})$ and $\mathcal{EL}^{\geq,\sqcup}$, such models do not exist. It is also interesting to note that we did not find a single case in which subsumption is tractable w.r.t. acyclic TBoxes, but intractable w.r.t. cyclic ones.

$\mathcal{EL}$ with	no TBox	acyclic	cyclic
$\neg C$	PSPACE	PSPACE	EXPTIME
$\neg A$	PTIME	PTIME	PTIME
$C \sqcup D$	CO-NP	PSPACE	EXPTIME
functionality	PTIME	CO-NP	PSPACE
$(\geq n\, r)$	PTIME	PTIME	PTIME
$p(f_1, \ldots, f_k)$	PTIME	PTIME	PTIME
$r \cap s$	PTIME	PTIME	PTIME
$r \cup s$	PTIME	PTIME	PTIME
r^-	PTIME	PSPACE	EXPTIME
r^+	CO-NP	PSPACE	EXPTIME

Figure 4. Complexity of subsumption in extensions of $\mathcal{EL}$. Light gray cell background indicates membership in class, dark gray completeness for class.

REFERENCES

[1] F. Baader, R. Molitor, and S. Tobies, 'Tractable and decidable fragments of conceptual graphs', in *Proc. of ICCS-99*, number 1640 in LNCS, pp. 480–493. Springer, (1999).

[2] F. Baader, 'Terminological cycles in a description logic with existential restrictions', in *Proc. of IJCAI-03*, pp. 325–330. Morgan Kaufmann, (2003).

[3] F. Baader, S. Brandt, and C. Lutz, 'Pushing the $\mathcal{EL}$ envelope', in *Proc. of IJCAI-05*, (2005). Morgan-Kaufmann Publishers.

[4] F. Baader, S. Brandt, and C. Lutz, 'Pushing the $\mathcal{EL}$ envelope further', in *Proc. of OWLED08DC*, (2008).

[5] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, eds., 'The Description Logic Handbook: Theory, Implementation, and Applications'. Cambridge University Press, 2003.

[6] S. Brandt, 'Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else?', in *Proc. of ECAI-2004*, pp. 298–302. IOS Press, (2004).

[7] D. Calvanese, 'Reasoning with inclusion axioms in description logics: Algorithms and complexity', in *Proc. of ECAI-96*, pp. 303–307, (1996).

[8] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati, 'Data complexity of query answering in description logics', in *Proc. of KR-2006*, pp. 260–270, (2006).

[9] F.M. Donini, B. Hollunder, M. Lenzerini, A.M. Spaccamela, D. Nardi, and W. Nutt, 'The complexity of existential quantification in concept languages', *Artificial Intelligence*, **53**(2–3), 309–327, (1992).

[10] C. Haase, 'Complexity of Subsumption in Extensions of $\mathcal{EL}$'. Master thesis, TU Dresden, Germany, 2007.

[11] D. Nardi, F.M. Donini, M. Lenzerini and W. Nutt, 'The complexity of concept languages', *Inf. and Comp.* **134**(1), 1–58, (1997).

[12] 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. of DL2003*, number 81 in CEUR-WS, (2003)

[13] C. Lutz, 'Description logics with concrete domains—a survey', in *Advances in Modal Logics Volume 4*, pp. 265–296. King's College Publications, (2003).

[14] C. Lutz, 'Complexity of terminological reasoning revisited', in *Proc. of LPAR-99*, number 1705 in LNAI, pp. 181–200. Springer, (1999).

[15] G. Miklau and D. Suciu, 'Containment and equivalence for an xpath fragment', in Proc. of PODS-02, pp. 65–76, (2002). ACM Press.

[16] B. Nebel, 'Terminological reasoning is inherently intractable', *Artificial Intelligence*, **43**, 235–249, (1990).

[17] K. Schild, 'Terminological cycles and propositional μ-calculus', in *Proc. of KR-94*, pp. 509–520, Morgan Kaufmann, (1994).

[18] M. Schmidt-Schauß and G. Smolka, 'Attributive concept descriptions with complements', *Artificial Intelligence*, **48**(1), 1–26, (1991).

[19] K.A. Spackman, 'Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT', *Journal of the American Medical Informatics Association*, (2000).

[20] The Gene Ontology Consortium, 'Gene Ontology: Tool for the unification of biology', *Nature Genetics*, **25**, 25–29, (2000).