Pushing the $\mathcal{EL}$ Envelope

Franz Baader and **Sebastian Brandt** and **Carsten Lutz**
Intitute for Theoretical Computer Science
TU Dresden, Germany
lastname@tcs.inf.tu-dresden.de

Abstract

Recently, it has been shown that the small description logic (DL) $\mathcal{EL}$, which allows for conjunction and *existential restrictions*, has better algorithmic properties than its counterpart $\mathcal{FL}_0$, which allows for conjunction and *value restrictions*. Whereas the subsumption problem in $\mathcal{FL}_0$ becomes already intractable in the presence of acyclic TBoxes, it remains tractable in $\mathcal{EL}$ even with general concept inclusion axioms (GCIs). On the one hand, we extend the positive result for $\mathcal{EL}$ by identifying a set of expressive means that can be added to $\mathcal{EL}$ without sacrificing tractability. On the other hand, we show that basically all other additions of typical DL constructors to $\mathcal{EL}$ with GCIs make subsumption intractable, and in most cases even EXPTIME-complete. In addition, we show that subsumption in $\mathcal{FL}_0$ with GCIs is EXPTIME-complete.

1 Introduction

The quest for tractable (i.e., polynomial-time decidable) description logics (DLs), which started in the 1980s after the first intractability results for DLs were shown [Brachman and Levesque, 1984; Nebel, 1988], was until recently restricted to DLs extending the basic language $\mathcal{FL}_0$, which allows for conjunction ($\sqcap$) and value restrictions ($\forall r.C$). The main reason was that, when clarifying the logical status of property arcs in semantic networks and slots in frames, the decision was taken that arcs/slots should be read as value restrictions rather than existential restrictions ($\exists r.C$).

For subsumption between concept descriptions, the tractability barrier was investigated in detail in the early 1990s [Donini *et al.*, 1991]. However, as soon as terminologies (TBoxes) were taken into consideration, tractability turned out to be unattainable: even with the simplest form of acyclic TBoxes, subsumption in $\mathcal{FL}_0$ (and thus in all languages extending it) is coNP-hard [Nebel, 1990]. Subsumption in $\mathcal{FL}_0$ is PSPACE-complete w.r.t. cyclic TBoxes [Baader, 1996; Kazakov and de Nivelle, 2003], and we show in this paper that it becomes even EXPTIME-complete in the presence of general concept inclusion axioms (GCIs), which are supported by all modern DL systems.

For these reasons, and also because of the need for expressive DLs supporting GCIs in applications, from the mid 1990s on the DL community has mainly given up on the quest of finding tractable DLs. Instead, it investigated more and more expressive DLs, for which reasoning is worst-case intractable. The goal was then to find practical subsumption algorithms, i.e., algorithms that are easy to implement and optimize, and which—though worst-case exponential or even worse—behave well in practice (see, e.g., [Horrocks *et al.*, 2000]). This line of research has resulted in the availability of highly optimized DL systems for expressive DLs [Horrocks, 1998; Haarslev and Möller, 2001], and successful applications: most notably the recommendation by the W3C of the DL-based language OWL [Horrocks *et al.*, 2003] as the ontology language for the Semantic Web.

Recently, the choice of value restrictions as a sine qua non of DLs has been reconsidered. On the one hand, it was shown that the DL $\mathcal{EL}$, which allows for conjunction and existential restrictions, has better algorithmic properties than $\mathcal{FL}_0$. Subsumption in $\mathcal{EL}$ stays tractable w.r.t. both acyclic and cyclic TBoxes [Baader, 2003b], and even in the presence of GCIs [Brandt, 2004]. On the other hand, there are applications where value restrictions are not needed, and where the expressive power of $\mathcal{EL}$ or small extensions thereof appear to be sufficient. In fact, SNOMED, the Systematized Nomenclature of Medicine, employs $\mathcal{EL}$ [Spackman, 2000] with an acyclic TBox. Large parts of the Galen medical knowledge base can also be expressed in $\mathcal{EL}$ with GCIs and transitive roles [Rector and Horrocks, 1997]. Finally, the Gene Ontology [Consortium, 2000] can be seen as an acyclic $\mathcal{EL}$ TBox with one transitive role.

Motivated by the positive complexity results cited above and the use of extensions of $\mathcal{EL}$ in applications, we start with the DL $\mathcal{EL}$ with GCIs, and investigate the effect on the complexity of the subsumption problem that is caused by the addition of standard DL constructors available in ontology languages like OWL. We prove that the subsumption problem remains tractable when adding the bottom concept (and thus disjointness statements), nominals (i.e., singleton concepts), a restricted form of concrete domains (e.g., references to numbers and strings), and a restricted form[1] of role-value maps

[1] Adding arbitrary role-value maps to $\mathcal{EL}$ is known to cause undecidability [Baader, 2003a].

Name	Syntax	Semantics
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
concrete domain	$p(f_1, \ldots, f_k)$ for $p \in \mathcal{P}^{\mathcal{D}_j}$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y_1, \ldots, y_k \in \Delta^{\mathcal{D}_j} : f_i^{\mathcal{I}}(x) = y_i$ for $1 \le i \le k \wedge (y_1, \ldots, y_k) \in p^{\mathcal{D}_j}\}$
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
RI	$r_1 \circ \cdots \circ r_k \sqsubseteq r$	$r_1^{\mathcal{I}} \circ \cdots \circ r_k^{\mathcal{I}} \subseteq r^{\mathcal{I}}$

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

(which can express transitivity and the right-identity rule required in medical applications [Spackman, 2000]). We then prove that, basically, all other additions of standard DL constructors lead to intractability of the subsumption problem, and in most cases even to EXPTIME-hardness. Proofs and further technical details can by found in the accompanying technical report [Baader *et al.*, 2005].

2 The Description Logic $\mathcal{EL}^{++}$

In DLs, *concept descriptions* are inductively defined with the help of a set of *constructors*, starting with a set $\mathsf{N_C}$ of *concept names*, a set $\mathsf{N_R}$ of *role names*, and (possibly) a set $\mathsf{N_I}$ of *individual names*. In this section, we introduce the extension $\mathcal{EL}^{++}$ of $\mathcal{EL}$, whose concept descriptions are formed using the constructors shown in the upper part of Table 1. There and in general, we use a and b to denote individual names, r and s to denote role names, and C, D to denote concept descriptions.

The concrete domain constructor provides an interface to so-called concrete domains, which permits reference to, e.g., strings and integers. Formally, a *concrete domain* $\mathcal{D}$ is a pair $(\Delta^{\mathcal{D}}, \mathcal{P}^{\mathcal{D}})$ with $\Delta^{\mathcal{D}}$ a set and $\mathcal{P}^{\mathcal{D}}$ a set of *predicate names*. Each $p \in \mathcal{P}$ is associated with an arity $n > 0$ and an extension $p^{\mathcal{D}} \subseteq (\Delta^{\mathcal{D}})^n$. To provide a link between the DL and the concrete domain, we introduce a set of *feature names* $\mathsf{N_F}$. In Table 1, p denotes a predicate of some concrete domain $\mathcal{D}$ and $f_1, \ldots, f_k$ are feature names. The DL $\mathcal{EL}^{++}$ may be equipped with a number of concrete domains $\mathcal{D}_1, \ldots, \mathcal{D}_n$ such that $\Delta^{\mathcal{D}_i} \cap \Delta^{\mathcal{D}_j} = \emptyset$ for $1 \le i < j \le n$. If we want to stress the use of particular concrete domains $\mathcal{D}_1, \ldots, \mathcal{D}_n$, we write $\mathcal{EL}^{++}(\mathcal{D}_1, \ldots, \mathcal{D}_n)$ instead of $\mathcal{EL}^{++}$.

The semantics of $\mathcal{EL}^{++}(\mathcal{D}_1, \ldots, \mathcal{D}_n)$-concept descriptions is defined in terms of an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. The *domain* $\Delta^{\mathcal{I}}$ is a non-empty set of individuals and the *interpretation function* $\cdot^{\mathcal{I}}$ maps each concept name $A \in \mathsf{N_C}$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$, each role name $r \in \mathsf{N_R}$ to a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$, each individual name $a \in \mathsf{N_I}$ to an individual $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, and each feature name $f \in \mathsf{N_F}$ to a partial function $f^{\mathcal{I}}$ from $\Delta^{\mathcal{I}}$ to $\bigcup_{1 \le i \le n} \Delta^{\mathcal{D}_i}$. The extension of $\cdot^{\mathcal{I}}$ to arbitrary concept descriptions is inductively defined as shown in the third column of Table 1.

An $\mathcal{EL}^{++}$ *constraint box (CBox)* is a finite set of *general concept inclusions (GCIs)* and *role inclusions (RIs)*, whose syntax can be found in the lower part of Table 1. Note that a finite set of GCIs would commonly be called a *general TBox*. We use the term CBox due to the presence of RIs. An interpretation $\mathcal{I}$ is a *model* of a CBox $\mathcal{C}$ if, for each GCI and RI in $\mathcal{C}$, the conditions given in the third column of Table 1 are satisfied. In the definition of the semantics of RIs, the symbol "$\circ$" denotes composition of binary relations.

The main inference problem considered in this paper is subsumption. Given two $\mathcal{EL}^{++}$-concept descriptions C, D we say that C *is subsumed by* D *w.r.t. the CBox* $\mathcal{C}$ ($C \sqsubseteq_{\mathcal{C}} D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{C}$.

Some remarks regarding the expressivity of $\mathcal{EL}^{++}$ are in order. First, though we restrict the attention to subsumption, $\mathcal{EL}^{++}$ is expressive enough to reduce all other standard reasoning tasks (concept satisfiability, ABox consistency, instance problem) to the subsumption problem and vice versa [Baader *et al.*, 2005]. Second, our RIs generalize three means of expressivity important in ontology applications: *role hierarchies* $r \sqsubseteq s$; *transitive roles*, which can be expressed by writing $r \circ r \sqsubseteq r$; and so-called *right-identity rules* $r \circ s \sqsubseteq s$, which are important in medical applications [Spackman, 2000; Horrocks and Sattler, 2003]. Third, the bottom concept in combination with GCIs can be used to express *disjointness* of complex concept descriptions: $C \sqcap D \sqsubseteq \bot$ says that C, D are disjoint. Finally, the *unique name assumption* for individual names can be enforced by writing $\{a\} \sqcap \{b\} \sqsubseteq \bot$ for all relevant individual names a and b.

3 Tractability of $\mathcal{EL}^{++}$

Before we can describe a polynomial-time subsumption algorithm for $\mathcal{EL}^{++}$, we must introduce an appropriate normal form for CBoxes. Given a CBox $\mathcal{C}$, we use $\mathsf{BC}_{\mathcal{C}}$ to denote the smallest set of concept descriptions that contains the top concept $\top$, all concept names used in $\mathcal{C}$, and all concept descriptions of the form $\{a\}$ or $p(f_1, \ldots, f_k)$ appearing in $\mathcal{C}$. Then, $\mathcal{C}$ is in *normal form* if

1. all GCIs have one of the following forms, where $C_1, C_2 \in \mathsf{BC}_{\mathcal{C}}$ and $D \in \mathsf{BC}_{\mathcal{C}} \cup \{\bot\}$:
$$\begin{array}{rclcrcl} C_1 & \sqsubseteq & D, & \quad & C_1 & \sqsubseteq & \exists r.C_2, \\ C_1 \sqcap C_2 & \sqsubseteq & D, & & \exists r.C_1 & \sqsubseteq & D. \end{array}$$
2. all role inclusions are of the form $r \sqsubseteq s$ or $r_1 \circ r_2 \sqsubseteq s$.

By introducing new concept and role names, any CBox $\mathcal{C}$ can be turned into a normalized CBox $\mathcal{C}'$ that is a *conservative extension* of $\mathcal{C}$, i.e., every model of $\mathcal{C}'$ is also a model of $\mathcal{C}$, and every model of $\mathcal{C}$ can be extended to a model of $\mathcal{C}'$ by appropriately choosing the interpretations of the additional concept and role names. In [Baader *et al.*, 2005] it shown that this transformation can actually be done in linear time, yielding a normalized CBox $\mathcal{C}'$ whose size is *linear* in the size of $\mathcal{C}$.

Lemma 1 *Subsumption w.r.t. CBoxes in* $\mathcal{EL}^{++}$ *can be reduced in linear time to subsumption w.r.t. normalized CBoxes in* $\mathcal{EL}^{++}$.

In the following, all CBoxes are assumed to be normalized. When developing the subsumption algorithm for normalized $\mathcal{EL}^{++}$ CBoxes, we can restrict our attention to subsumption

CR1 If $C' \in S(C)$, $C' \sqsubseteq D \in \mathcal{C}$, and $D \notin S(C)$
then $S(C) := S(C) \cup \{D\}$

CR2 If $C_1, C_2 \in S(C)$, $C_1 \sqcap C_2 \sqsubseteq D \in \mathcal{C}$, and $D \notin S(C)$
then $S(C) := S(C) \cup \{D\}$

CR3 If $C' \in S(C)$, $C' \sqsubseteq \exists r.D \in \mathcal{C}$, and $(C, D) \notin R(r)$
then $R(r) := R(r) \cup \{(C, D)\}$

CR4 If $(C, D) \in R(r)$, $D' \in S(D)$, $\exists r.D' \sqsubseteq E \in \mathcal{C}$,
and $E \notin S(C)$
then $S(C) := S(C) \cup \{E\}$

CR5 If $(C, D) \in R(r)$, $\bot \in S(D)$, and $\bot \notin S(C)$,
then $S(C) := S(C) \cup \{\bot\}$

CR6 If $\{a\} \in S(C) \cap S(D)$, $C \leadsto_R D$, and $S(D) \not\subseteq S(C)$
then $S(C) := S(C) \cup S(D)$

CR7 If $\mathsf{con}_j(S(C))$ is unsatisfiable in $\mathcal{D}_j$ and $\bot \notin S(C)$,
then $S(C) := S(C) \cup \{\bot\}$

CR8 If $\mathsf{con}_j(S(C))$ implies $p(f_1, \ldots, f_k) \in \mathsf{BC}_\mathcal{C}$ in $\mathcal{D}_j$
and $p(f_1, \ldots, f_k) \notin S(C)$,
then $S(C) := S(C) \cup \{p(f_1, \ldots, f_k)\}$

CR9 If $p(f_1, \ldots, f_k), p'(f'_1, \ldots, f'_{k'}) \in S(C)$, $p \in \mathcal{P}^{\mathcal{D}_j}$,
$p' \in \mathcal{P}^{\mathcal{D}_\ell}$, $j \neq \ell$, $f_s = f'_t$ for some s, t, and $\bot \notin S(C)$,
then $S(C) := S(C) \cup \{\bot\}$

CR10 If $(C, D) \in R(r)$, $r \sqsubseteq s \in \mathcal{C}$, and $(C, D) \notin R(s)$
then $R(s) := R(s) \cup \{(C, D)\}$

CR11 If $(C, D) \in R(r_1)$, $(D, E) \in R(r_2)$, $r_1 \circ r_2 \sqsubseteq r_3 \in \mathcal{C}$,
and $(C, E) \notin R(r_3)$
then $R(r_3) := R_i(r_3) \cup \{(C, E)\}$

Table 2: Completion Rules

between concept *names*. In fact, $C \sqsubseteq_\mathcal{C} D$ iff $A \sqsubseteq_{\mathcal{C}'} B$, where $\mathcal{C}' = \mathcal{C} \cup \{A \sqsubseteq C, D \sqsubseteq B\}$ with A and B new concept names. Our subsumption algorithm not only computes subsumption between two given concept names w.r.t. the normalized input CBox $\mathcal{C}$; it rather *classifies* $\mathcal{C}$, i.e., it simultaneously computes the subsumption relationships between *all* pairs of concept names occurring in $\mathcal{C}$.

Now, let $\mathcal{C}$ be a CBox in normal form that is to be classified. We use $\mathsf{R}_\mathcal{C}$ to denote the set of all role names used in $\mathcal{C}$. The algorithm computes

- a mapping S from $\mathsf{BC}_\mathcal{C}$ to a subset of $\mathsf{BC}_\mathcal{C} \cup \{\top, \bot\}$, and
- a mapping R from $\mathsf{R}_\mathcal{C}$ to a binary relation on $\mathsf{BC}_\mathcal{C}$.

The intuition is that these mappings make implicit subsumption relationships explicit in the following sense:

(I1) $D \in S(C)$ implies that $C \sqsubseteq_\mathcal{C} D$,

(I2) $(C, D) \in R(r)$ implies that $C \sqsubseteq_\mathcal{C} \exists r.D$.

In the algorithm, these mappings are initialized as follows:

- $S(C) := \{C, \top\}$ for each $C \in \mathsf{BC}_\mathcal{C}$,
- $R(r) := \emptyset$ for each $r \in \mathsf{R}_\mathcal{C}$.

Then the sets $S(C)$ and $R(r)$ are extended by applying the completion rules shown in Table 2 until no more rule applies.

Some of the rules use abbreviations that still need to be introduced. First, **CR6** uses the relation $\leadsto_R \subseteq \mathsf{BC}_\mathcal{C} \times \mathsf{BC}_\mathcal{C}$, which is defined as follows: $C \leadsto_R D$ iff there are $C_1, \ldots, C_k \in \mathsf{BC}_\mathcal{C}$ such that

- $C_1 = C$ or $C_1 = \{b\}$ for some individual name b,
- $(C_j, C_{j+1}) \in R(r_j)$ for some $r_j \in \mathsf{R}_\mathcal{C}$ ($1 \leq j < k$),
- $C_k = D$.

Second, rules **CR7** and **CR8** use the notion $\mathsf{con}_j(S_i(C))$, and satisfiability and implication in a concrete domain. If p is a predicate of the concrete domain $\mathcal{D}_j$, then the $\mathcal{EL}^{++}$-concept description $p(f_1, \ldots, f_n)$ can be viewed as an atomic first-order formula with variables $f_1, \ldots, f_n$. Thus, it makes sense to consider Boolean combinations of such atomic formulae, and to talk about whether such a formula is satisfiable in (the first-order interpretation) $\mathcal{D}_j$, or whether in $\mathcal{D}_j$ one such formula implies another one. For a set Γ of $\mathcal{EL}^{++}(\mathcal{D}_1, \ldots, \mathcal{D}_n)$-concept descriptions and $1 \leq j \leq n$, we define

$$\mathsf{con}_j(\Gamma) := \bigwedge_{p(f_1,\ldots,f_k) \in \Gamma \text{ with } p \in \mathcal{P}^{\mathcal{D}_j}} p(f_1, \ldots, f_k).$$

For the rules **CR7** and **CR8** to be executable in polynomial time, satisfiability and implication in the concrete domains $\mathcal{D}_1, \ldots, \mathcal{D}_n$ must be decidable in polynominal time. However, for our algorithm to be complete, we must impose an additional condition on the concrete domains. The concrete domain $\mathcal{D}$ is *p-admissible* iff

1. satisfiability and implication in $\mathcal{D}$ are decidable in polynominal time;
2. $\mathcal{D}$ is *convex*: if a conjunction of atoms of the form $p(f_1, \ldots, f_k)$ implies a disjunction of such atoms, then it also implies one of its disjuncts.

Let us now show that the rules of Table 2 indeed yield a polynomial algorithm for subsumption in $\mathcal{EL}^{++}(\mathcal{D}_1, \ldots, \mathcal{D}_n)$ provided that the concrete domains $\mathcal{D}_1, \ldots, \mathcal{D}_n$ are p-admissible.

The following lemma is an easy consequence of the facts that (i) each rule application adds an element to one of the sets $S(C) \subseteq \mathsf{BC}_\mathcal{C} \cup \{\top, \bot\}$ or $R(C, D) \subseteq \mathsf{BC}_\mathcal{C} \times \mathsf{BC}_\mathcal{C}$, (ii) the cardinality of $\mathsf{BC}_\mathcal{C}$ is polynomial in the size of $\mathcal{C}$, and (iii) the relation $\leadsto_R$ can be computed using (polytime) graph reachability, and (iv) the concrete domains are p-admissible.

Lemma 2 *For a normalized CBox $\mathcal{C}$, the rules of Table 2 can only be applied a polynomial number of times, and each rule application is polynomial.*

The next lemma shows how all subsumption relationships between concept names occurring in $\mathcal{C}$ can be determined once the completion algorithm has terminated.

Lemma 3 *Let S be the mapping obtained after the application of the rules of Table 2 for the normalized CBox $\mathcal{C}$ has terminated, and let A, B be concept names occurring in $\mathcal{C}$. Then $A \sqsubseteq_\mathcal{C} B$ iff one of the following two conditions holds:*

- $S(A) \cap \{B, \bot\} \neq \emptyset$,
- *there is an* $\{a\} \in \mathsf{BC}_\mathcal{C}$ *such that* $\bot \in S(\{a\})$.

The if-direction of this lemma (soundness) immediately follows from the fact that (I1) and (I2) are satisfied for the initial definition of S, R, and that application of the rules preserves (I1) and (I2). This is trivial for most of the rules. We consider **CR6** in more detail. If $\{a\} \in S(C) \cap S(D)$, then $C, D \sqsubseteq_{\mathcal{C}} \{a\}$. Now, $C \rightsquigarrow_R D$ implies that $C \sqsubseteq_{\mathcal{C}} \exists r_1.\cdots\exists r_{k-1}.D$ or $\{b\} \sqsubseteq_{\mathcal{C}} \exists r_1.\cdots\exists r_{k-1}.D$ for some individual name b. In the second case, this implies that D cannot be empty in any model of $\mathcal{C}$, and in the first case it implies that D is non-empty in any model of $\mathcal{C}$ for which C is non-empty. Together with $C, D \sqsubseteq_{\mathcal{C}} \{a\}$, this implies that $C \sqsubseteq_{\mathcal{C}} D$, which shows that the rule **CR6** is sound since it preserves (I1).

To show the only-if-direction of the lemma, we assume that the two conditions do not hold, and then use the computed mappings S, R to construct a model $\mathcal{I}$ of $\mathcal{C}$ such that $A^{\mathcal{I}} \not\subseteq B^{\mathcal{I}}$. Basically, the domain $\Delta^{\mathcal{I}}$ of this model consists of all elements $C \in \mathsf{BC}_{\mathcal{C}}$ such that $A \rightsquigarrow_R C$. However, we identify elements C, D of $\mathsf{BC}_{\mathcal{C}}$ if there is an individual name a such that $\{a\} \in S(C) \cap S(D)$. The interpretation of concept and role names is determined by S, R: if $D \in \mathsf{BC}_{\mathcal{C}}$ is a concept name, then $D^{\mathcal{I}} = \{C \in \mathsf{BC}_{\mathcal{C}} \mid D \in S(C)\}$, and if r is a role name, then $(C, D) \in r^{\mathcal{I}}$ iff $(C, D) \in R(r)$. Finally, the interpretation of the feature names is determined by the the assignments satisfying the conjunctions $\mathsf{con}_j(S(C))$ (see [Baader *et al.*, 2005] for a more detailed description of this construction, and a proof that it indeed yields a countermodel to the subsumption relationship $A \sqsubseteq_{\mathcal{C}} B$).

To sum up, we have shown the following tractability result:

Theorem 4 *Let $\mathcal{D}_1, \ldots, \mathcal{D}_n$ be p-admissible concrete domains. Then subsumption in $\mathcal{EL}^{++}(\mathcal{D}_1, \ldots, \mathcal{D}_n)$ w.r.t. CBoxes can be decided in polynomial time.*

P-admissibile and non-admissible concrete domains

In order to obtain concrete DLs of the form $\mathcal{EL}^{++}(\mathcal{D}_1, \ldots, \mathcal{D}_n)$ for $n > 0$ to which Theorem 4 applies, we need concrete domains that are p-admissible. In the following, we introduce two concrete domains that are p-admissible, and show that small extensions of them are no longer p-admissible.

The concrete domain $\mathsf{Q} = (\mathbb{Q}, \mathcal{P}^{\mathsf{Q}})$ has as its domain the set $\mathbb{Q}$ of rational numbers, and its set of predicates $\mathcal{P}^{\mathsf{Q}}$ consists of the following predicates:

- a unary predicate $\top_{\mathsf{Q}}$ with $(\top_{\mathsf{Q}})^{\mathsf{Q}} = \mathbb{Q}$;
- unary predicates $=_q$ and $>_q$ for each $q \in \mathbb{Q}$;
- a binary predicate $=$;
- a binary predicate $+_q$, for each $q \in \mathbb{Q}$, with $(+_q)^{\mathsf{Q}} = \{(q', q'') \in \mathbb{Q}^2 \mid q' + q = q''\}$.

The concrete domain S is defined as $(\Sigma^*, \mathcal{P}^{\mathsf{S}})$, where Σ is the ISO 8859-1 (Latin-1) character set and $\mathcal{P}^{\mathsf{S}}$ consistes of the following predicates:

- a unary predicate $\top_{\mathsf{S}}$ with $(\top_{\mathsf{S}})^{\mathsf{S}} = \Sigma^*$;
- a unary predicate $=_w$, for each $w \in \Sigma^*$;
- a binary predicate $=$;
- a binary predicate conc_w, for each $w \in \Sigma^*$, with $\mathsf{conc}_w^{\mathsf{Q}} = \{(w', w'') \mid w'' = w'w\}$.

Polynomiality of reasoning in Q can be shown by a reduction to linear programming, and polynomiality of reasoning in S has been proved in [Lutz, 2003]. Moreover, we can show [Baader *et al.*, 2005] that both Q and S are convex.

Proposition 5 *Q and S are p-admissible.*

Both Q and S are interesting concrete domains since they allow us to refer to concrete numbers and strings in concepts, and use the properties of the concrete predicates when reasoning. However, the predicates available in these concrete domains are rather restricted. Unfortunately, p-admissibility is a fragile property, i.e., we cannot extend Q and S by other interesting predicates without losing p-admissibility. As an illustration, we consider one extension of Q and one of S.

The concrete domain $\mathsf{Q}^{\leq_q, >_q}$ extends Q by the additional unary predicates $(\leq_q)_{q \in \mathbb{Q}}$. Then the $\mathsf{Q}^{\leq_q, >_q}$-conjunction $>_0(f')$ implies the disjunction $\leq_0(f) \vee >_0(f)$ without implying one of its disjuncts. Thus, $\mathsf{Q}^{\leq_q, >_q}$ is not convex.

Next, consider any concrete domain S^* with domain Σ^* for some finite alphabet Σ and, for every $s \in \Sigma^*$, the unary predicates pref_s and suff_s with the semantics

$$\mathsf{pref}_s^{\mathsf{S}^*} := \{s' \in \Sigma^* \mid s \text{ is a prefix of } s'\}$$
$$\mathsf{suff}_s^{\mathsf{S}^*} := \{s' \in \Sigma^* \mid s \text{ is a suffix of } s'\}.$$

Let $\Sigma = \{a_1, \ldots, a_n\}$. Then the S^*-conjunction $\mathsf{suff}_{a_1}(f)$ implies the disjunction $\mathsf{pref}_{a_1}(f) \vee \ldots \vee \mathsf{pref}_{a_n}(f)$ without implying any of its disjuncts.

4 Intractable extensions of $\mathcal{EL}$ with GCIs

In this section we consider the sublanguage $\mathcal{EL}$ of $\mathcal{EL}^{++}$ and restrict the attention to general TBoxes, i.e., finite sets of GCIs. Recall that $\mathcal{EL}$ is obtained from $\mathcal{EL}^{++}$ by dropping all concept constructors except conjunction, existential restriction, and top. We will show that the extension of $\mathcal{EL}$ with basically any typical DL constructor not present in $\mathcal{EL}^{++}$ results in intractability of subsumption w.r.t. general TBoxes. Syntax and semantics of the additional constructors used in this section can be found in Table 3, where $\#S$ denotes the cardinality of a set S.

Name	Syntax	Semantics
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
value restriction	$\forall r.C$	$\{x \mid \forall y : (x, y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
at-least restriction	$(\geqslant n\ r)$	$\{x \mid \#\{y \in \Delta^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}}\} \geq n\}$
at-most restriction	$(\leqslant n\ r)$	$\{x \mid \#\{y \in \Delta^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}}\} \leq n\}$
inverse roles	$\exists r^-.C$	$\{x \mid \exists y : (y, x) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$

Table 3: The additional constructors.

In addition to the subsumption problem, we will sometimes also consider the *satisfiability problem*: the concept description C is satisfiable w.r.t. the general TBox $\mathcal{T}$ iff there exists a model $\mathcal{I}$ of $\mathcal{T}$ with $C^{\mathcal{I}} \neq \emptyset$. As in the previous section, we can restrict the attention to satisfiability/subsumption of concept *names* w.r.t. general TBoxes.

Atomic negation

Let $\mathcal{EL}^{\neg}$ be the extension of $\mathcal{EL}$ with negation, and let $\mathcal{EL}^{(\neg)}$ be obtained from $\mathcal{EL}^{\neg}$ by restricting the applicability of negation to concept names (*atomic* negation). Since $\mathcal{EL}^{\neg}$ is a notational variant of the DL $\mathcal{ALC}$, ExpTime-completeness of satisfiability and subsumption in $\mathcal{ALC}$ w.r.t. general TBoxes [Schild, 1991] carries over to $\mathcal{EL}^{\neg}$. EXPTIME-completeness even carries over to $\mathcal{EL}^{(\neg)}$ since $\neg C$ with C complex can be replaced with $\neg A$ for a new concept name A if we add the two GCIs $A \sqsubseteq C$ and $C \sqsubseteq A$.

Theorem 6 *In $\mathcal{EL}^{(\neg)}$, satisfiability and subsumption w.r.t. general TBoxes is* EXPTIME*-complete.*

Disjunction

Let $\mathcal{ELU}$ be the extension of $\mathcal{EL}$ with disjunction. Subsumption in $\mathcal{ELU}$ w.r.t. general TBoxes is in EXPTIME since $\mathcal{ELU}$ is a fragment of $\mathcal{ALC}$. To obtain a matching EXPTIME lower bound, we reduce satisfiability in $\mathcal{EL}^{(\neg)}$ w.r.t. general TBoxes to subsumption in $\mathcal{ELU}$ w.r.t. general TBoxes. To this end, let A_0 be a concept name and $\mathcal{T}$ a general $\mathcal{EL}^{(\neg)}$ TBox. For each concept name A occurring in $\mathcal{T}$, we take a *new* concept name A' (i.e., one not occurring in $\mathcal{T}$). Also fix an additional new concept name L. Then the general TBox $\mathcal{T}^*$ is obtained from $\mathcal{T}$ by replacing each subconcept $\neg A$ with A', and then adding the following GCIs:

- $\top \sqsubseteq A \sqcup A'$ and $A \sqcap A' \sqsubseteq L$ for each $A \in \mathsf{N_C}$ in $\mathcal{T}$;
- $\exists r.L \sqsubseteq L$.

Note that $\exists r.L \sqsubseteq L$ is equivalent to $\neg L \sqsubseteq \forall r.\neg L$. It thus ensures that L acts as the bottom concept in connected countermodels of $A_0 \sqsubseteq_{\mathcal{T}^*} L$. Using this observation, it is not hard to verify that A_0 is satisfiable w.r.t. $\mathcal{T}$ iff $A_0 \not\sqsubseteq_{\mathcal{T}^*} L$.

Theorem 7 *In $\mathcal{ELU}$, subsumption w.r.t. general TBoxes is* EXPTIME*-complete.*

At-least restrictions

Let $\mathcal{EL}^{\geq 2}$ be the extension of $\mathcal{EL}$ with at-least restrictions of the form $(\geqslant 2\ r)$. Subsumption in $\mathcal{EL}^{\geq 2}$ w.r.t. general TBoxes is in EXPTIME since $\mathcal{EL}^{\geq 2}$ is a fragment of $\mathcal{ALC}$ extended with number restrictions [De Giacomo and Lenzerini, 1994]. We establish a matching lower bound by reducing subsumption in $\mathcal{ELU}$ w.r.t. general TBoxes. Let A_0 and B_0 be concept names and $\mathcal{T}$ a general $\mathcal{ELU}$ TBox. Without loss of generality, we may assume that all concept inclusions in $\mathcal{T}$ have one of the following forms:

$$C \sqsubseteq D, \quad \exists r.C \sqsubseteq D, \qquad C \sqsubseteq \exists r.D,$$
$$C_1 \sqcap C_2 \sqsubseteq C, \quad C \sqsubseteq C_1 \sqcup C_2,$$

where C, D, C_1, and C_2 are concept names or $\top$. Call the resulting TBox $\mathcal{T}^*$. To convert $\mathcal{T}$ into an $\mathcal{EL}^{\geq 2}$ CBox, we simulate each GCI $C \sqsubseteq C_1 \sqcup C_2$ by introducing two new concept names A and B and a new role name r, and putting

$$C \sqsubseteq \exists r.A \sqcap \exists r.B,$$
$$C \sqcap \exists r.(A \sqcap B) \sqsubseteq C_1, \qquad C \sqcap (\geqslant 2\ r) \sqsubseteq C_2.$$

It is easy to see that $A_0 \sqsubseteq_{\mathcal{T}} B_0$ iff $A_0 \sqsubseteq_{\mathcal{T}^*} B_0$.

Theorem 8 *In $\mathcal{EL}^{\geq 2}$, subsumption w.r.t. general TBoxes is* EXPTIME*-complete.*

The interested reader may note that similar reductions can be used to show EXPTIME-completeness for $\mathcal{EL}$ extended with one of the role constructors negation, union, and transitive closure.

Non-p-admissible concrete domains

P-admissibility of the concrete domains is not only a sufficient condition for polynomiality of reasoning in $\mathcal{EL}^{++}$, but also a necessary one: if $\mathcal{D}$ is a non-convex concrete domain, then subsumption in $\mathcal{EL}(\mathcal{D})$ is EXPTIME-hard, where $\mathcal{EL}(\mathcal{D})$ is the extension of $\mathcal{EL}$ with the concrete domain $\mathcal{D}$. The proof uses a stronger version of Theorem 7: we can show that subsumption of concept names w.r.t. *restricted* $\mathcal{ELU}$ TBoxes is EXPTIME-complete, where a restricted $\mathcal{ELU}$ TBox is a general $\mathcal{EL}$ TBox extended with a *single* GCI of the form $A \sqsubseteq B_1 \sqcup B_2$ [Baader *et al.*, 2005].

The subsumption problem for such restricted $\mathcal{ELU}$ TBoxes can be reduced to subsumption in $\mathcal{EL}^{++}(\mathcal{D})$ as follows. Let A_0 and B_0 be concept names and $\mathcal{T}$ a restricted $\mathcal{ELU}$ TBox. Since $\mathcal{D}$ is not convex, there is a satisfiable conjunction c of atoms of the form $p(f_1, \ldots, f_k)$ that implies a disjunction $a_1 \vee \ldots \vee a_m$ of such atoms, but none of its disjuncts. If we assume that this is a minimal such counterexample (i.e., m is minimal), then we also know that c does not imply $a_2 \vee \ldots \vee a_m$, and that each of the a_i is satifiable. Then we have (i) each assignment of values from $\mathcal{D}$ that satisfies c satisfies a_1 or $a_2 \vee \ldots \vee a_m$; (ii) there is an assignment satisfying c and a_1, but not $a_2 \vee \ldots \vee a_m$; (iii) there is an assignment satisfying c and $a_2 \vee \ldots \vee a_m$ but not a_1. Now, let $\mathcal{T}^*$ be obtained from $\mathcal{T}$ by replacing the single GCI $A \sqsubseteq B \sqcup B'$ by $A \sqsubseteq c$, $a_1 \sqsubseteq B$, and $a_i \sqsubseteq B'$ for $i = 2, \ldots, m$. It is easy to see that $A_0 \sqsubseteq_{\mathcal{T}} B_0$ iff $A_0 \sqsubseteq_{\mathcal{T}^*} B_0$.

Theorem 9 *For any non-convex concrete domain $\mathcal{D}$, subsumption in $\mathcal{EL}(\mathcal{D})$ w.r.t. general TBoxes is* EXPTIME*-hard.*

For example, this theorem applies to the non-convex concrete domains introduced in Section 3.

Inverse roles

Let $\mathcal{ELI}$ be the extension of $\mathcal{EL}$ with inverse roles. We show that subsumption in $\mathcal{ELI}$ w.r.t. general TBoxes is PSPACE-hard by reducing satisfiability in $\mathcal{ALE}$ w.r.t. *primitive TBoxes*: $\mathcal{ALE}$ extends $\mathcal{EL}$ with value restrictions and atomic negation; primitive TBoxes are finite sets of GCIs whose left-hand side is a concept *name*. This satisfiability problem is known to be PSPACE-complete [Calvanese, 1996].

Let A_0 be a concept name and $\mathcal{T}$ a primitive $\mathcal{ALE}$ TBox. We assume without loss of generality that $\mathcal{T}$ contains only concept inclusions of the following forms:

$$A \sqsubseteq B, \quad A \sqsubseteq \neg B, \quad A \sqsubseteq \exists r.B, \quad A \sqsubseteq \forall r.B,$$

where A, B, and B' are concept names. We take a new concept name L and define the general $\mathcal{ELI}$ TBox $\mathcal{T}^*$ to consist of the following GCIs:

- $A \sqsubseteq D$ for all $A \sqsubseteq D \in \mathcal{T}$
  if D is a concept name or of the form $\exists r.B$;
- $\exists r^-.A \sqsubseteq B$ for all $A \sqsubseteq \forall r.B \in \mathcal{T}$;
- $A \sqcap B \sqsubseteq L$ for all $A \sqsubseteq \neg B \in \mathcal{T}$;
- $\exists r.L \sqsubseteq L$.

As in the case of $\mathcal{ELU}$, the concept inclusion $\exists r.L \sqsubseteq L$ is equivalent to $\neg L \sqsubseteq \forall r.\neg L$ and ensures that L acts as the bottom concept in connected countermodels of $A_0 \sqsubseteq_{\mathcal{T}^*} L$. Additionally, $\exists r^-.A \sqsubseteq B$ is clearly equivalent to $A \sqsubseteq \forall r.B$.

Thus, it is not hard to verify that A_0 is satisfiable w.r.t. $\mathcal{T}$ iff $A_0 \not\sqsubseteq_{\mathcal{T}^*} L$.

Theorem 10 *Subsumption in $\mathcal{ELI}$ w.r.t. general TBoxes is* PSPACE*-hard.*

The exact complexity of this problem is still open (the best upper bound we know of is EXPTIME, stemming from results for the DL $\mathcal{ALCI}$ [De Giacomo and Lenzerini, 1994]).

At-most restrictions/functional roles/$\mathcal{FL}_0$

Let $\mathcal{EL}^{\leq 1}$ be the extension of $\mathcal{EL}$ with at-most restrictions of the forn $(\leqslant 1\ r)$. As in the case of $\mathcal{EL}^{\geq 2}$, subsumption in $\mathcal{EL}^{\leq 1}$ w.r.t. general TBoxes is in EXPTIME since $\mathcal{EL}^{\leq 1}$ is a fragment of $\mathcal{ALC}$ with number restrictions. We prove a matching lower bound by reducing subsumption in the DL $\mathcal{FL}_0^{tf}$ w.r.t. general TBoxes. $\mathcal{FL}_0^{tf}$ offers only the concept constructors conjunction and value restriction and requires all roles to be interpreted as total functions. Subsumption in this DL w.r.t. general TBoxes was proved EXPTIME-complete in [Toman and Weddell, 2005].

Let A_0 and B_0 be concept names and $\mathcal{T}$ a general $\mathcal{FL}_0^{tf}$ TBox. We convert $\mathcal{T}$ into a general $\mathcal{EL}^{\leq 1}$ TBox $\mathcal{T}^*$ by replacing each subconcept $\forall r.C$ with $\exists r.C$ in GCI left-hand sides, and with $(\leqslant 1\ r) \sqcap \exists r.C$ in GCI right-hand sides. Then $A_0 \sqsubseteq_{\mathcal{T}} B_0$ in $\mathcal{FL}_0^{tf}$ iff $A_0 \sqsubseteq_{\mathcal{T}^*} B_0$ in $\mathcal{EL}^{\leq 1}$.

Theorem 11 *Subsumption in $\mathcal{EL}^{\leq 1}$ w.r.t. general TBoxes is* EXPTIME*-complete.*

Interestingly, the result of Toman and Weddell can also be used to prove EXPTIME-completeness of subsumption in $\mathcal{FL}_0$ w.r.t. general TBoxes. Recall that $\mathcal{FL}_0$ is $\mathcal{FL}_0^{tf}$ without the restriction that roles be total functions. In [Baader *et al.*, 2005] we show that $A_0 \sqsubseteq_{\mathcal{T}} B_0$ in $\mathcal{FL}_0$ iff $A_0 \sqsubseteq_{\mathcal{T}} B_0$ in $\mathcal{FL}_0^{tf}$, i.e., subsumption in $\mathcal{FL}_0$ and $\mathcal{FL}_0^{tf}$ coincides.

Theorem 12 *Subsumption in $\mathcal{FL}_0$ w.r.t. general TBoxes is* EXPTIME*-complete.*

5 Conclusion

We believe that the results of this paper show that—in contrast to the negative conclusions drawn from early complexity results in the area—the quest for tractable DLs that are expressive enough to be useful in practice can be successful. Our DL $\mathcal{EL}^{++}$ is tractable even w.r.t. GCIs, and it offers many constructors that are important in ontology applications.

This is in strong contrast to its counterpart with value restrictions: $\mathcal{FL}_0$ is tractable without TBoxes [Brachman and Levesque, 1984], co-NP-complete for acyclic TBoxes [Nebel, 1990], PSPACE-complete for cyclic TBoxes [Baader, 1996; Kazakov and de Nivelle, 2003], and EXPTIME-complete for general TBoxes (as shown above, and, independently, in [Hofmann, 2005]).

References

[Baader *et al.*, 2005] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ envelope. LTCS-Report LTCS-05-01, Inst. for Theoretical Computer Science, TU Dresden, 2005. See http://lat.inf.tu-dresden.de/research/reports.html.

[Baader, 1996] F. Baader. Using automata theory for characterizing the semantics of terminological cycles. *AMAI*, 18, 1996.

[Baader, 2003a] F. Baader. Restricted role-value-maps in a description logic with existential restrictions and terminological cycles. In *Proc. DL'03*, http://CEUR-WS.org/Vol-81/, 2003.

[Baader, 2003b] F. Baader. Terminological cycles in a description logic with existential restrictions. In *Proc. IJCAI'03*, 2003.

[Brachman and Levesque, 1984] R.J. Brachman and H.J. Levesque. The tractability of subsumption in frame-based description languages. In *Proc. AAAI'84*, 1984.

[Brandt, 2004] S. Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In *Proc. ECAI 2004*, 2004.

[Calvanese, 1996] D. Calvanese. Reasoning with inclusion axioms in description logics: Algorithms and complexity. In *Proc. ECAI'96*, 1996.

[Consortium, 2000] The GO Consortium. Gene Ontology: Tool for the unification of biology. *Nature Genetics*, 25, 2000.

[De Giacomo and Lenzerini, 1994] G. De Giacomo and M. Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics. In *Proc. AAAI'94*, 1994.

[Donini *et al.*, 1991] F. Donini, M. Lenzerini, D. Nardi, and W. Nutt. Complexity of concept languages. In *Proc. KR'91*, 1991.

[Haarslev and Möller, 2001] V. Haarslev and R. Möller. RACER system description. In *Proc. IJCAR 01*, 2001.

[Hofmann, 2005] M. Hofmann. Proof-theoretic approach to description logic. In *Proc. of LICS-05*. 2005

[Horrocks and Sattler, 2003] I. Horrocks and U. Sattler. Decidability of SHIQ with complex role inclusion axioms. In *Proc. IJCAI-03*, 2003.

[Horrocks *et al.*, 2000] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *J. of the IGPL*, 8, 2000.

[Horrocks *et al.*, 2003] I. Horrocks, P.F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *J. of Web Semantics*, 1, 2003.

[Horrocks, 1998] I. Horrocks. Using an expressive description logic: FaCT or fiction? In *Proc. KR'98*, 1998.

[Kazakov and de Nivelle, 2003] 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*. http://CEUR-WS.org/Vol-81/, 2003.

[Lutz, 2003] C. Lutz. NExpTime-complete description logics with concrete domains. *ACM Transactions on Computational Logic*, 5(4):669-705, 2004.

[Nebel, 1988] B. Nebel. Computational complexity of terminological reasoning in BACK. *Artificial Intelligence*, 34, 1988.

[Nebel, 1990] B. Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43, 1990.

[Rector and Horrocks, 1997] A. Rector and I. Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proc. Workshop on Ontological Engineering*, 1997.

[Schild, 1991] K. Schild. A correspondence theory for terminological logics: Preliminary report. In *Proc. IJCAI'91*, 1991.

[Spackman, 2000] K.A. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. *J. of the American Medical Informatics Association*, 2000. Fall Symposium Special Issue.

[Toman and Weddell, 2005] D. Toman and G. Weddell. On reasoning about structural equality in XML: A description logic approach. *TCS*, 2005. To appear.