TECHNISCHE
UNIVERSITÄT
DRESDEN

Dresden University of Technology
Institute for Theoretical Computer Science
Chair for Automata Theory

LTCS–Report

Pushing the $\mathcal{EL}$ Envelope

Franz Baader   Sebastian Brandt   Carsten Lutz

LTCS-Report 05-01

Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
http://lat.inf.tu-dresden.de

Hans-Grundig-Str. 25
01062 Dresden
Germany

Pushing the $\mathcal{EL}$ Envelope

Franz Baader Sebastian Brandt Carsten Lutz

Institut für Theoretische Informatik
TU Dresden, Germany
lastname@tcs.inf.tu-dresden.de

Abstract

Recently, it has been shown that the small 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 w.r.t. general concept inclusion axioms (GCIs). On the one hand, we will 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 will 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 will 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 [6, 26], 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 [10]. 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 [27]. Subsumption in $\mathcal{FL}_0$ is PSpace-complete w.r.t. cyclic TBoxes [3, 21], 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., [20]). This line of research has resulted in the availability of highly optimized DL systems for expressive DLs [17, 14], and successful applications: most notably the recommendation by the W3C of the DL-based language OWL [18] 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 [4], and even in the presence of GCIs [7]. 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 [9] employs $\mathcal{EL}$ [30, 31] with an acyclic TBox. Large parts of the Galen medical knowledge base can also be expressed in $\mathcal{EL}$ with GCIs and transitive roles [28]. Finally, the Gene Ontology [32] 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 of role-value maps (which can express transitivity and the right-identity rule required in medical applications [30]). 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.

2 Description Logics

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 $f_1, \dots, f_k$ are feature names. The DL $\mathcal{EL}^{++}$ may be equipped with a number of concrete domains $\mathcal{D}_1, \dots, \mathcal{D}_n$ such that

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 $1 \leq i \leq k \ \wedge (y_1, \ldots, y_k) \in p^{\mathcal{D}_j}\}$
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
RI	$r_1 \circ \cdots \circ r_k \sqsubseteq r$	$r_1^{\mathcal{I}} \circ \cdots \circ r_k^{\mathcal{I}} \subseteq r^{\mathcal{I}}$
concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$r(a,b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$

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

$\Delta^{\mathcal{D}_i} \cap \Delta^{\mathcal{D}_j} = \emptyset$ for $1 \leq i < j \leq n$. 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 $\bigcup_{1 \leq i \leq n} \Delta^{\mathcal{D}_i}$. 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 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.

An $\mathcal{EL}^{++}$ *assertional box (ABox)* is a finite set of *concept assertions* and *role assertions* , whose syntax can also be found in Table 1. ABoxes are used to describe a snapshot of the world. An interpretation $\mathcal{I}$ is a *model* of an ABox $\mathcal{A}$ if, for each concept assertion and role assertion in $\mathcal{A}$, the conditions given in the third column of Table 1 are satisfied.

The most relevant inference problems for description logics can be described as follows:

- *Concept satisfiability.* A concept C is *satisfiable w.r.t. a CBox* $\mathcal{C}$ if there exists a model $\mathcal{I}$ of $\mathcal{C}$ such that $C^{\mathcal{I}} \neq \emptyset$.

- *Concept subsumption.* A concept C *subsumes* a concept D w.r.t. a CBox $\mathcal{C}$ (written $C \sqsubseteq_{\mathcal{C}} D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in every model $\mathcal{I}$ of $\mathcal{C}$.
- *ABox consistency.* An ABox $\mathcal{A}$ is *consistent* w.r.t. a CBox $\mathcal{C}$ if $\mathcal{A}$ and $\mathcal{C}$ have a common model.
- *The instance problem.* An individual name a is an *instance* of a concept C in an ABox $\mathcal{A}$ w.r.t. a CBox $\mathcal{C}$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$ for every common model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{C}$.

In the remainder of this paper, we will concentrate on subsumption as the basic reasoning task. This is justified by the facts that, first, all of the above reasoning tasks can be mutually polynomially reduced to one another, and second, subsumption is the most "traditional" reasoning service in description logics. We show mutual reducibility by reducing all (other) reasoning tasks to subsumption, and vice versa:

- Satisfiability to (non-)subsumption: a concept C is satisfiable w.r.t. a CBox $\mathcal{C}$ iff $C \not\sqsubseteq_{\mathcal{C}} \bot$.
- Instance problem to subsumption. We convert an ABox $\mathcal{A}$ into a concept $C_{\mathcal{A}}$ as follows:

  $$C_{\mathcal{A}} := \bigsqcap_{C(a) \in \mathcal{A}} \exists u.(\{a\} \sqcap C) \quad \sqcap \quad \bigsqcap_{r(a,b) \in \mathcal{A}} \exists u.(\{a\} \sqcap \exists r.\{b\})$$

  where u is a new role name not used in $\mathcal{A}$. Then, an individual a is an instance of a concept C in an ABox $\mathcal{A}$ w.r.t. a CBox $\mathcal{C}$ iff $\{a\} \sqcap C_{\mathcal{A}} \sqsubseteq_{\mathcal{C}} C$.
- Consistency to subsumption: $\mathcal{A}$ is consistent w.r.t. $\mathcal{C}$ iff $C_{\mathcal{A}} \not\sqsubseteq \bot$.
- Subsumption to satisfiability: $C \sqsubseteq_{\mathcal{C}} D$ iff $C \sqcap \{a\}$ is unsatisfiable w.r.t. the CBox $\mathcal{C} \cup \{D \sqcap \{a\} \sqsubseteq \bot\}$, where a is an individual name not occurring in C, D, and $\mathcal{C}$.
- Subsumption to the instance problem: $C \sqsubseteq_{\mathcal{C}} D$ if a is an instance of D in the ABox $\{a : C\}$ w.r.t. $\mathcal{C}$.
- Subsumption to consistency: $C \sqsubseteq_{\mathcal{C}} D$ iff the ABox $\{C(a)\}$ is inconsistent w.r.t. the TBox $\mathcal{C} \cup \{D \sqcap \{a\} \sqsubseteq \bot\}$.

Three remarks regarding the expressivity of $\mathcal{EL}^{++}$ are in order. First, 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 [30, 19]. Second, 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 Deciding Subsumption in $\mathcal{EL}^{++}(\mathcal{D}_1, \ldots, \mathcal{D}_k)$

We develop a polynomial time algorithm for subsumption in $\mathcal{EL}^{++}$. To this end, it is convenient to first introduce an appropriate normal form for CBoxes.

3.1 A Normal Form for CBoxes

Given a CBox $\mathcal{C}$, we use $\mathsf{BC}_\mathcal{C}$ to denote the set of *basic concept descriptions for* $\mathcal{C}$, i.e., the smallest set of concept descriptions that contains

- the top concept $\top$;
- all concept names used in $\mathcal{C}$;
- all (sub)concepts of the form $\{a\}$ or $p(f_1, \ldots, f_k)$ appearing in $\mathcal{C}$.

Now, a normal form for CBoxes can be defined as follows.

Definition 1 (Normal Form for CBoxes). An $\mathcal{EL}^{++}$-CBox $\mathcal{C}$ is in *normal form* if

1. all concept inclusions have one of the following forms, where $C_1, C_2 \in \mathsf{BC}_\mathcal{C}$ and $D \in \mathsf{BC}_\mathcal{C} \cup \{\bot\}$:

$$\begin{aligned} C_1 &\sqsubseteq D \\ C_1 \sqcap C_2 &\sqsubseteq D \\ C_1 &\sqsubseteq \exists r.C_2 \\ \exists r.C_1 &\sqsubseteq D \end{aligned}$$

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 the interpretations of the additional concept and role names.

We now show that this transformation can actually be done in linear time, yielding a normalized CBox $\mathcal{C}'$ whose size is *linear* in the size of $\mathcal{C}$, where the size $|\mathcal{C}|$ of a CBox $\mathcal{C}$ is the is the number of symbols needed to write down $\mathcal{C}$.

Lemma 2. *Subsumption w.r.t. CBoxes in $\mathcal{EL}^{++}$ can be reduced in linear time to subsumption w.r.t. normalized CBoxes in $\mathcal{EL}^{++}$.*

Proof. A CBox can be converted into normal form using the translation rules shown in Figure 1 in two phases:

1. exhaustively apply rules **NF1** to **NF4**;
2. exhaustively apply rules **NF5** to **NF7**.

Here "rule application" means that the concept inclusion on the left-hand side is replaced with the set of concept inclusions on the right-hand-side. In Phase 1, the rule **NF2** is applied modulo commutativity of conjunction. It is easily verified that the size of the normalized CBox $\mathcal{C}'$ computed by applying the normalization rules is linear in the size of the original CBox $\mathcal{C}$, and that $\mathcal{C}'$ is computed using at most $|\mathcal{C}|$ rule applications. ❏

$$
\begin{array}{lrcl}
\textsf{NF1} & r_1 \circ \cdots \circ r_k \sqsubseteq s & \longrightarrow & \{r_1 \circ \cdots \circ r_{k-1} \sqsubseteq u,\ u \circ r_k \sqsubseteq s\} \\
\textsf{NF2} & C \sqcap \hat{D} \sqsubseteq E & \longrightarrow & \{\ \hat{D} \sqsubseteq A, C \sqcap A \sqsubseteq E\ \} \\
\textsf{NF3} & \exists r.\hat{C} \sqsubseteq D & \longrightarrow & \{\ \hat{C} \sqsubseteq A, \exists r.A \sqsubseteq D\ \} \\
\textsf{NF4} & \bot \sqsubseteq D & \longrightarrow & \emptyset \\
 & & & \\
\textsf{NF5} & \hat{C} \sqsubseteq \hat{D} & \longrightarrow & \{\ \hat{C} \sqsubseteq A, A \sqsubseteq \hat{D}\ \} \\
\textsf{NF6} & B \sqsubseteq \exists r.\hat{C} & \longrightarrow & \{\ B \sqsubseteq \exists r.A, A \sqsubseteq \hat{C}\ \} \\
\textsf{NF7} & B \sqsubseteq C \sqcap D & \longrightarrow & \{\ B \sqsubseteq C, B \sqsubseteq D\ \}
\end{array}
$$

where $\hat{C}, \hat{D} \notin \mathsf{BC}_\mathcal{C}$, u denotes a new role name, and A a new concept name.

Figure 1: Normalization Rules

Note that the CBox obtained by rule application is of *linear* size only since we apply normalization rules in two phases: if all rules are applied together in one phase, we obtain a quadratic blowup in the worst case due to the duplication of the concept B by Rule **NF7**.

3.2 The Algorithm

We now develop a polynomial-time algorithm for deciding subsumption in $\mathcal{EL}^{++}$ w.r.t. CBoxes in normal form. Here and in the remainder of the paper, we can restrict our attention to subsumption 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 $C \in \mathsf{BC}_\mathcal{C}$,

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 $\{a\} \in S(C) \cap S(D)$, $C \rightsquigarrow D$, 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

- $R(r) := \emptyset$ 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 $\rightsquigarrow \subseteq \mathsf{BC}_{\mathcal{C}} \times \mathsf{BC}_{\mathcal{C}}$, which is defined as follows: $C \rightsquigarrow D$ 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.

Definition 3. The concrete domain $\mathcal{D}$ is *p-admissible* if

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.

We investigate the property of p-admissibility in more detail in Section 4, where we also exhibit some useful concrete domains that are p-admissible.

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 4. *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* $\bot \in S(\{a\})$.

Lemma 4 will be proved in the subsequent section, where it is also shown that the algorithm terminates after polynomially many rule applications. Before going into formal details, let us briefly discuss soundness of the algorithm on an intuitive level. 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 trivially seen for most of the rules. However, it is worthwhile to consider **CR6** in more detail. If $\{a\} \in S(C) \cap S(D)$, then $C, D \sqsubseteq_{\mathcal{C}} \{a\}$. Now, $C \rightsquigarrow D$ 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). When dropping the requirement $C \rightsquigarrow D$ from this rule, (I1) is no longer preserved.

3.3 Soundness, Completeness,and Termination

We start with proving termination after polynomially many rule applications.

Lemma 5. *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.*

Proof. It is readily checked that the cardinality of $\mathsf{BC}_\mathcal{C}$ and $\mathsf{R}_\mathcal{C}$ is linear in the size of $\mathcal{C}$. Each rule application performed by the algorithm adds a new element of $\mathsf{BC}_\mathcal{C} \cup \{\bot\}$ to a set $S(C)$, for some $C \in \mathsf{BC}_\mathcal{C}$, or a new tuple $(C, D) \in \mathsf{BC}_\mathcal{C} \times \mathsf{BC}_\mathcal{C}$ to a relation $R(r)$, for some $r \in \mathsf{R}_\mathcal{C}$. Since no rule removes elements of these sets/relations, the total number of rule applications is polynomial. It is readily checked that each rule application can be performed in polynomial time. In particular, note that the relation $\rightsquigarrow$ can be computed using (polytime) graph reachability, ❏

We now prove Lemma 4. For conveniene, we treat the "if" direction (soundness) and the "only if" direction (completeness) separately. In the proofs, we will use the notion of a *solution* for a conjunction $\mathsf{con}_i(S(C))$. Such a solution is a mapping $\delta : \mathsf{N_F} \to \Delta^\mathcal{D}$ such that $(\delta(f_1), \ldots, \delta(f_k)) \in p^{\mathcal{D}_i}$ (henceforth denoted with $\delta \models p(f_1, \ldots, f_k)$) for each conjunct $p(f_1, \ldots, f_k)$ of c. Clearly, a conjunction $\mathsf{con}_i(S(C))$ is satisfiable iff there exists a solution for it.

Lemma 6 (Soundness). *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 $A \sqsubseteq_\mathcal{C} B$ if one of the following two conditions holds:*

S1 $S(A) \cap \{B, \bot\} \neq \emptyset$,

S2 there is an $\{a\} \in \mathsf{BC}_\mathcal{C}$ *such that* $\bot \in S(\{a\})$.

Proof. Assume that the algorithm is applied to a normalized CBox $\mathcal{C}$ yielding the sequences of mappings $S_0, \ldots, S_m$ and $R_0, \ldots, R_m$. Let A_0 and B_0 be two concept names such that (at least) one of the Conditions $S1$ and $S2$ is satisfied. To show that $A_0 \sqsubseteq_\mathcal{C} B_0$, we prove the following claim.

Claim. For all $n \in \mathbb{N}$, models $\mathcal{I}$ of $\mathcal{C}$, $r \in \mathsf{R}_\mathcal{C}$, and $x \in C^\mathcal{I}$, the following holds:

(a) if $D \in S_n(C)$ then $x \in D^\mathcal{I}$; and

(b) if $(C, D) \in R_n(r)$ then there is a $y \in \Delta^\mathcal{I}$ with $(x, y) \in r^\mathcal{I}$ and $y \in D^\mathcal{I}$.

The claim is proved by induction on n. Let $\mathcal{I}$ be a model of $\mathcal{C}$ and $x \in C^\mathcal{I}$. First for the induction start. For (a), $n = 0$ implies $S_n(C) = \{C, \top\}$. Thus, $x \in C^\mathcal{I}$ implies $x \in D^\mathcal{I}$ for all $D \in S_n(C)$. Point (b) is immediate since $R_0(r) = \emptyset$ for all $r \in \mathsf{R}_\mathcal{C}$. Now for the induction step. For (a), we assume that $D \in S_n(C) \setminus S_{n-1}(C)$ (for otherwise we are done by the induction hypothesis). We make a case distinction according to the rule that was used to add the concept D to S_n:

CR1 Then there exists a $C' \in S_{n-1}(C)$ and a concept inclusion $I = C' \sqsubseteq D \in \mathcal{C}$. By Point (a) of the induction hypothesis (IH), we have $x \in C'^\mathcal{I}$ implying by I that also $x \in D^\mathcal{I}$.

CR2 Then there exist $C_1, C_2 \in S_{n-1}(C)$ and a concept inclusion $I = C_1 \sqcap C_2 \sqsubseteq D \in \mathcal{C}$. By Point (a) of IH, $C_1, C_2 \in S_{n-1}(A)$ yields $x \in C_1^{\mathcal{I}}$ and $x \in C_2^{\mathcal{I}}$, implying by I that $x \in D^{\mathcal{I}}$.

CR4 Then there exist $E, E' \in \mathsf{BC}_{\mathcal{C}}$, a role name $r \in \mathsf{R}_{\mathcal{C}}$, and a concept inclusion $I = \exists r.E' \sqsubseteq D \in \mathcal{C}$ such that $(C, E) \in R_{n-1}(r)$ and $E' \in S_{n-1}(E)$. By Point (b) of IH, there is a $y \in \Delta^{\mathcal{I}}$ such that $(x, y) \in r^{\mathcal{I}}$ and $y \in E^{\mathcal{I}}$. By Point (a) of IH, we have $y \in E'^{\mathcal{I}}$. Thus I yields $x \in D^{\mathcal{I}}$.

CR5 If this rule is used, then we have $D = \bot$ and there is an $E \in \mathsf{BC}_{\mathcal{C}}$ such that $(C, E) \in \mathsf{R}(r)$ for some $r \in \mathsf{R}_{\mathcal{C}}$ and $\bot \in S_{n-1}(E)$. By Point (b) of IH, there is a $y \in \Delta^{\mathcal{I}}$ such that $(x, y) \in r^{\mathcal{I}}$ and $y \in E^{\mathcal{I}}$. By Point (a) of IH, we have $y \in \bot^{\mathcal{I}}$. As this is impossible, we conclude that there are no models $\mathcal{I}$ of $\mathcal{C}$ with $C^{\mathcal{I}} \neq \emptyset$. Thus, adding $\bot$ to $S(C)$ (trivially) preserves Point (a).

CR6 Then there exists an $E \in \mathsf{BC}_{\mathcal{C}}$ and an individual name a such that $\{a\} \in S_{n-1}(C) \cap S_{n-1}(E)$, $D \in S_{n-1}(E)$, and there are $C_1, \ldots, C_k \in \mathsf{BC}_{\mathcal{C}}$ such that

(i) $C_1 = C$ or $C_1 = \{b\}$ for some individual name b;

(ii) $C_k = E$;

(iii) $(C_i, C_{i+1}) \in R_{n-1}(r_j)$ for some $r_j \in \mathsf{R}_{\mathcal{C}}$ $(1 \leq i < k)$.

By Point (b) of IH and (iii), there are $y_1, \ldots, y_k \in \Delta^{\mathcal{I}}$ s.t. $y_1 \in \{x\} \cup \{b^{\mathcal{I}} | b \in \mathsf{N_I}\}$, $y_k \in C_k^{\mathcal{I}} = E^{\mathcal{I}}$, and $(y_i, y_{i+1}) \in r_j^{\mathcal{I}}$ for some $r_j \in \mathsf{R}_{\mathcal{C}}$ $(1 \leq i < k)$. By Point (a) of IH, $x \in C^{\mathcal{I}}$ and $\{a\} \in S_{n-1}(C) \cap S_{n-1}(E)$ implies $x = a^{\mathcal{I}} = y_k$. Also by Point (a), $D \in S_{n-1}(E)$ implies $y_k \in D^{\mathcal{I}}$. Thus, $x \in D^{\mathcal{I}}$ as required.

CR7 If this rule is used, then we have $D = \bot$ and $\mathsf{con}_i(S_{n-1}(C))$ is unsatisfiable for some i. Define a function $\delta : \mathsf{N_F} \to \Delta^{\mathcal{D}_i}$ by setting $\delta(f) := f^{\mathcal{I}}(x)$. Using Part (a) of IH, we get that $x \in p(f_1, \ldots, f_k)^{\mathcal{I}}$ for every conjunct $p(f_1, \ldots, f_k)$ of $\mathsf{con}_i(S_{n-1}(C))$. Thus, δ is a solution for $\mathsf{con}_i(S_{n-1}(C))$, contradicting its unsatisfiability. Thus, there can be no model $\mathcal{I}$ of $\mathcal{C}$ with $C^{\mathcal{I}} \neq \emptyset$. Thus, adding $\bot$ to $S(C)$ (trivially) preserves Point (a).

CR8 Then D is of the form $p(f_1, \ldots, f_k)$ with $p \in \mathcal{P}^{\mathcal{D}_i}$ for some i, and $\mathsf{con}_i(S_{n-1}(C))$ implies D. As in the previous case, we have $x \in p(f_1, \ldots, f_k)^{\mathcal{I}}$ for every conjunct $p(f_1, \ldots, f_k)$ of $\mathsf{con}_i(S_{n-1}(C))$ by Part (a) of IH. Since $\mathsf{con}_i(S_{n-1}(C))$ implies D, we thus have $x \in D^{\mathcal{I}}$ as required.

CR9 If this rule is used, then we have $D = \bot$ and there are $p(f_1, \ldots, f_k) \in S_{n-1}(C)$ and $p'(f'_1, \ldots, f'_{k'}) \in S_{n-1}(C)$ such that $p \in \mathcal{P}^{\mathcal{D}_i}$ and $p' \in \mathcal{P}^{\mathcal{D}_j}$ with $i \neq j$. By Point (a) of IH, we have $x \in p(f_1, \ldots, f_k)^{\mathcal{I}} \cap p'(f'_1, \ldots, f'_{k'})^{\mathcal{I}}$. Thus $f_i^{\mathcal{I}} \in \Delta^{\mathcal{D}_i} \cap \Delta^{\mathcal{D}_j}$, contradicting the disjointness of $\Delta^{\mathcal{D}_i}$ and $\Delta^{\mathcal{D}_j}$. Again, Point(a) is trivially preserved.

For (b), we assume $(C, D) \in R_n(r) \setminus R_{n-1}(r)$ and make a case distinction according to the rule that was used to add (C, D) to $R_n(r)$:

CR3 Then there is a $C' \in \mathsf{BC}_{\mathcal{C}}$ with $C' \in S_{n-1}(C)$ and a concept inclusion $I = C' \sqsubseteq \exists r.D \in \mathcal{C}$. By Point (a) of IH, $x \in C^{\mathcal{I}}$ implies $x \in C'^{\mathcal{I}}$. By I, there is a y such that $(x, y) \in r^{\mathcal{I}}$ and $y \in D^{\mathcal{I}}$ as required.

CR10 Then $(C, D) \in R_{n-1}(s)$ for some s with $s \sqsubseteq r \in \mathcal{C}$. By Point (b) of IH, there is a $y \in \Delta^{\mathcal{I}}$ such that $(x, y) \in s^{\mathcal{I}}$ and $y \in D^{\mathcal{I}}$. Since $s \sqsubseteq r \in \mathcal{C}$, we have $(x, y) \in r^{\mathcal{I}}$ and are done.

CR11 Then there is an $E \in \mathsf{BC}_{\mathcal{C}}$ such that $(C, E) \in R_{n-1}(r_1)$ and $(E, D) \in R_{n-1}(r_2)$ for some r_1, r_2 with $r_1 \circ r_2 \sqsubseteq r \in \mathcal{C}$. By Point (b) of IH, there is a $y \in \Delta^{\mathcal{I}}$ such that $(x, y) \in r_1^{\mathcal{I}}$ and $y \in E^{\mathcal{I}}$. Another application of Point (b) yields the existence of a $z \in \Delta^{\mathcal{I}}$ such that $(y, z) \in r_2^{\mathcal{I}}$ and $z \in D^{\mathcal{I}}$. Since $r_1 \circ r_2 \sqsubseteq r \in \mathcal{C}$, we have $(x, z) \in r^{\mathcal{I}}$ and are done.

This finishes the proof of Claim 1.

Using the claim, it is now easy to prove that $A_0 \sqsubseteq_{\mathcal{C}} B_0$. We make a case distinction according to whether condition $S1$ or $S2$ is satisfied.

$S1$ Let $B_0 \in S_m(A_0)$. By Point (a) of Claim 1, we have $x \in B_0^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{C}$ and all $x \in A_0^{\mathcal{I}}$. In other words, $A_0 \sqsubseteq_{\mathcal{C}} B_0$. Now let $\bot \in S_m(A_0)$. By Point (a) of Claim 1, we have $x \in \bot^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{C}$ and all $x \in A_0^{\mathcal{I}}$. In other words, there are no models $\mathcal{I}$ of $\mathcal{C}$ with $A_0^{\mathcal{I}} \neq \emptyset$. Thus $A_0 \sqsubseteq_{\mathcal{C}} B_0$.

$S2$ Let $\bot \in S_m(\{a\})$ for some individual name a. By Point (a) of Claim 1, we have $a^{\mathcal{I}} \in \bot^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{C}$. In other words, there are no models of $\mathcal{C}$. Thus $A_0 \sqsubseteq_{\mathcal{C}} B_0$.

❑

Lemma 7 (Completeness). *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 $A \sqsubseteq_{\mathcal{C}} B$ implies that one of the following two conditions holds:*

S1 $S(A) \cap \{B, \bot\} \neq \emptyset$,

S2 there is an $\{a\} \in \mathsf{BC}_{\mathcal{C}}$ *such that* $\bot \in S(\{a\})$.

Proof. We show the contrapositive. Thus assume that the algorithm does not satisfy $S1$ and $S2$ after termination. We show that this implies $A_0 \not\sqsubseteq_{\mathcal{C}} B_0$ by constructing a model $\mathcal{I}$ of $\mathcal{C}$ such that $a \in A_0^{\mathcal{I}} \setminus B_0^{\mathcal{I}}$ for some $a \in \Delta^{\mathcal{I}}$.

Assume that the algorithm computed the sequences of mappings $S_0, \ldots, S_m$ and $R_0, \ldots, R_m$. For convenience, denote S_m with S and R_m with R. Set $\mathsf{BC}_{\mathcal{C}}^- := \{C \in \mathsf{BC}_{\mathcal{C}} \mid A_0 \rightsquigarrow C\}$. Then define a relation $\sim$ on $\mathsf{BC}_{\mathcal{C}}^-$ as follows:

$$C \sim D \text{ iff } C = D \text{ or } \{a\} \in S(C) \cap S(D) \text{ for some individual name } a.$$

Using Rule **CR6**, it is readily checked that "$\sim$" is an equivalence relation. We use $[C]$ to denote the equivalence class of $C \in \mathsf{BC}_{\mathcal{C}}^-$ w.r.t. "$\sim$". The equivalence classes of "$\sim$" will be used to define the domain elements of the model to be constructed. Before actually defining this model, we prove two claims:

Claim 1. For all $C, C' \in \mathsf{BC}_{\mathcal{C}}^{-}$ with $C \sim C'$ and all $r \in \mathsf{R}_{\mathcal{C}}$, we have

1. $S(C) = S(C')$;
2. $(C, D) \in R(r)$ implies $(C', D) \in R(r)$.

Proof: Point 1 is an immediate consequence of non-applicability of **CR6**. The proof of Point 2 is by induction on the smallest i such that $(C, D) \in R_i(r)$. As $R_0(r) = \emptyset$ for all role names r, the induction start is trivial. Now for the induction step. Let $(C, D) \in R_i(r) \setminus R_{i-1}(r)$ with $i > 0$. We make a case distinction according to the rule applied:

CR3 Then there is an $E \in S_{i-1}(C)$ and a concept inclusion $I = E \sqsubseteq \exists r.D \in \mathcal{C}$. Since $C \sim C'$, **CR6** ensures that $E \in S_j(C')$ for some $j \geq 0$. Thus **CR3** ensures that $(D, E) \in R(r)$.

CR10 Then we have $(C, D) \in R_{i-1}(s)$ for some role name s with $s \sqsubseteq r \in \mathcal{C}$. By IH, this implies $(C', D) \in R_j(s)$ for some $j \geq 0$. Thus, **CR12** ensures that $(C', D) \in R(r)$.

CR11 Then there is an $E \in \mathsf{BC}_{\mathcal{C}}$ such that $(C, E) \in R_{i-1}(r_1)$ and $(E, D) \in R_{i-1}(r_2)$ for some role names r_1, r_2 with $r_1 \circ r_2 \sqsubseteq r \in \mathcal{C}$. By definition of "$\rightsquigarrow$", $C \in \mathsf{BC}_{\mathcal{C}}^{-}$ implies $D \in \mathsf{BC}_{\mathcal{C}}^{-}$. Thus, the IH yields $(C, E) \in R_{i-1}(r_1)$, which implies $(C', E) \in R_j(r_1)$ for some $j \geq 0$. **CR13** will eventually be applied to $(C', E) \in R_\ell(r_1)$ and $(E, D) \in R_\ell(r_2)$ for some $\ell \geq 0$, yielding $(C', D) \in R_{\ell+1}(r) \subseteq R(r)$.

This finishes the proof of Claim 1. Point (1) allows us to unambigously identify a given equivalence class $[C]$ of "$\sim$" with a set of concepts $S(C)$. This will be used implicitly in what followows.

Claim 2. For each $C \in \mathsf{BC}_{\mathcal{C}}^{-}$ and each $i \in \{1, \ldots, n\}$, we can find a solution $\delta([C], i)$ for $\mathsf{con}_i(S(C))$ such that, for all concepts $D \in \mathsf{BC}_{\mathcal{C}}$ of the form $p(f_1, \ldots, f_k)$ with $p \in \mathcal{P}^{\mathcal{D}_i}$, we have $\delta([C], t) \models D$ iff $D \in S(C)$.

Proof: By Conditions $S1$ and $S2$, we have $\bot \notin S(A_0)$ and $\bot \notin S(\{a\})$ for all $\{a\} \in \mathsf{BC}_{\mathcal{C}}$. Due to Rule **CR5** and by definition of $\mathsf{BC}_{\mathcal{C}}^{-}$, it follows that $\bot \notin S(C)$. Thus, by Rule **CR7** there exists a solution for $\mathsf{con}_i(S(C))$. It remains to be shown that this solution can be chosen such that it does not satisfy any concept $p(f_1, \ldots, f_k) \in \mathsf{BC}_{\mathcal{C}} \setminus S(C)$. Let Γ be the set of all solutions for $\mathsf{con}_i(S(C))$. Moreover, assume to the contrary of what is to be shown that there exists a set $\Psi \subseteq \mathsf{BC}_{\mathcal{C}} \setminus S(C)$ of concepts of the form $p(f_1, \ldots, f_k)$ with $p \in \mathcal{P}^{\mathcal{D}_i}$ such that each solution from Γ satisfies a concept from Ψ, i.e., $\mathsf{con}_i(S(C))$ implies the disjunction of all concepts in Ψ. By Property 2 of p-admissibility, $\mathsf{con}_i(S(C))$ implies a single concept X from Ψ. By rule **CR8**, this implies $X \in S(C)$ in contradiction to $X \in \Psi$.

This finishes the proof of Claim 2. For each $C \in \mathsf{BC}_{\mathcal{C}}^{-}$ and each $i \in \{1, \ldots, n\}$, fix a solution $\delta([C], i)$ for $\mathsf{con}_i(S(C))$ as in Claim 2. We now define an interpretation $\mathcal{I}$ as follows:

$$\begin{array}{rcl} \Delta^{\mathcal{I}} & := & \{[C] \mid C \in \mathsf{BC}_{\mathcal{C}}^{-}\}; \\ A^{\mathcal{I}} & := & \{[C] \in \Delta^{\mathcal{I}} \mid A \in S(C)\} \text{ for all } A \in \mathsf{N_C} \cap \mathsf{BC}_{\mathcal{C}}; \\ a^{\mathcal{I}} & := & [\{a\}] \text{ for all } \{a\} \in \mathsf{BC}_{\mathcal{C}}; \\ r^{\mathcal{I}} & := & \{([C],[D]) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid \exists D' \in [D] : (C,D) \in R(r)\} \text{ for all } r \in \mathsf{R}_{\mathcal{C}}; \\ f^{\mathcal{I}}([C]) & := & \delta([C],i) \text{ if there exists a } p(f_1,\dots,f_m) \in S(C) \text{ with } p \in \mathcal{P}^{\Delta_i} \text{ and} \\ & & f_j = f \text{ for some } j \in \{1,\dots,m\}, \text{ for all } f \in \mathsf{N_F} \text{ and } [C] \in \Delta^{\mathcal{I}}. \end{array}$$

All concept names not in $\mathsf{BC}_{\mathcal{C}}$ and all role names not in $R_{\mathcal{C}}$ are mapped to the empty set. Each individual name a with $\{a\} \notin \mathsf{BC}_{\mathcal{C}}$ is interpreted as $a^{\mathcal{I}} := [A_0]$ (this choice is arbitrary). Note the following:

- the use of the equivalence relation "$\sim$" ensures that, for each individual name a, $a^{\mathcal{I}}$ is well-defined;
- the interpretation of roles is well-defined due to Point 2 of Claim 1.
- the interpretation of features is well-defined since $\bot \notin S(C)$ for all $C \in \mathsf{BC}_{\mathcal{C}}^{-}$ and due to Rule **CR9**.

We now erstablish an additional, central claim.

Claim 3. For all $[C] \in \Delta^{\mathcal{I}}$ and $D \in \mathsf{BC}_{\mathcal{C}} \cup \{\bot\}$, we have $[C] \in D^{\mathcal{I}}$ iff $D \in S(C)$.

The proof makes a case distinction according to the form of D:

- $D = \top$. Easy since $\top \in S(C)$ for all $C \in \mathsf{BC}_{\mathcal{C}}^{-}$.
- $D = \bot$. Easy since, in the proof of Claim 2, we already argued that $\bot \notin S(C)$ for all $C \in \mathsf{BC}_{\mathcal{C}}^{-}$.
- D is a concept name. Then $[C] \in D^{\mathcal{I}}$ iff $D \in S(C)$ is immediate by definition of $\mathcal{I}$.
- $D = \{a\}$. Then $[C] \in \{a\}^{\mathcal{I}}$ implies $a^{\mathcal{I}} = [C]$ and thus $[C] = [\{a\}]$ by definition of $\{a\}^{\mathcal{I}}$. This yields $\{a\} \in S(C)$ since $\{a\} \in S_0(\{a\})$. Conversely, $\{a\} \in S(C)$ implies $[C] = [\{a\}]$ by definition of "$\sim$" and thus $a^{\mathcal{I}} = [C]$ implying $[C] \in \{a\}^{\mathcal{I}}$ by the semantics.
- $D = p(f_1,\dots,f_k)$ with $p \in \mathcal{P}^{\mathcal{D}_i}$ for some i. Then $[C] \in D^{\mathcal{I}}$ iff $\delta([C],i) \models D$ iff $D \in S(C)$. The first "iff" is by definition of $\mathcal{I}$ and the semantics and the latter by choice of $\delta([C],i)$.

This finishes the proof of Claim 3. We now show that $\mathcal{I}$ is a model of $\mathcal{C}$ with $x \in (A_0^{\mathcal{I}} \setminus B_0^{\mathcal{I}})$ for some $x \in \Delta^{\mathcal{I}}$. Since $A_0 \in \mathsf{BC}_{\mathcal{C}}^{-}$ by definition of $\mathsf{BC}_{\mathcal{C}}^{-}$, we have $[A_0] \in \Delta^{\mathcal{I}}$. By $S1$, we have $B_0 \notin S(A_0)$. By definition of S_0, we have $A_0 \in S(A_0)$. Thus, Claim 3 yields $[A_0] \in (A_0^{\mathcal{I}} \setminus B_0^{\mathcal{I}})$. It remains to be shown that $\mathcal{I}$ is a model of $\mathcal{C}$. We make a case distinction according to the form of concept and role inclusions:

- $C \sqsubseteq D$. Let $[C'] \in C^{\mathcal{I}}$. By Claim 3, we have $C \in S(C')$. Due to Rule **CR1**, this implies $D \in S(C')$ and thus $[C'] \in D^{\mathcal{I}}$ by Claim 3.
- $C \sqcap D \sqsubseteq E$. Similar to the previous case using Rule **CR2**.
- $C \sqsubseteq \exists r.D$. Let $[C'] \in C^{\mathcal{I}}$. Then $C \in S(C')$ by Claim 3. By Rule **CR3**, we thus have $(C', D) \in R(r)$. By definition of $r^{\mathcal{I}}$, this implies $([C'], [D]) \in r^{\mathcal{I}}$. Moreover, $D \in S_0(D)$ implies $D \in S(D)$. Thus, Claim 3 yields $[D] \in D^{\mathcal{I}}$. Together, this yields $[C'] \in (\exists r.D)^{\mathcal{I}}$ as required.
- $\exists r.C \sqsubseteq D$. Let $[E] \in (\exists r.C)^{\mathcal{I}}$. Hence there is an $[F] \in \Delta^{\mathcal{I}}$ such that $([E], [F]) \in r^{\mathcal{I}}$ and $[F] \in C^{\mathcal{I}}$. By definition of $\mathcal{I}$, this means that there is $F' \in [F]$ such that $(E, F') \in R(r)$. Moreover, $[F'] = [F] \in C^{\mathcal{I}}$ implies $C \in S(F')$ by Claim 3. By Rule **CR4**, we thus have $D \in S(E)$. Thus $[E] \in D^{\mathcal{I}}$ by Claim 3 as required.
- $r \sqsubseteq s$. Let $([C], [D]) \in r^{\mathcal{I}}$. Then there is a $D' \in [D]$ such that $(C, D') \in R(r)$. By **CR10**, we obtain $(C, D') \in R(s)$. By definition of $\mathcal{I}$, we thus have $([C], [D']) = ([C], [D]) \in s^{\mathcal{I}}$ as required.
- $r_1 \circ r_2 \sqsubseteq s$. Let $([C], [D]) \in r_1^{\mathcal{I}}$ and $([D], [E]) \in r_2^{\mathcal{I}}$. Then there are $D' \in [D]$ and $E' \in [E]$ such that $(C, D') \in R(r)$ and $(D, E') \in R(r)$. By Point 2 of Claim 1, the latter yields $(D', E') \in R(r)$. By **CR10**, we thus obtain $(C, E') \in R(s)$. By definition of $\mathcal{I}$, we thus have $([C], [E']) = ([C], [E]) \in s^{\mathcal{I}}$ as required.

❑

We obtain the following result as a consequence of Lemmas 2, 5, and 4, and the reduction of satisfiability, consistency, and the instance problem to subsumption given in Section 2.

Theorem 8. *Satisfiability, subsumption, ABox consistency, and the instance problem in $\mathcal{EL}^{++}$ can be decided in polynomial time.*

It is not hard to see that, taken together, the proofs of Lemma 6 and 7 yield a small model property for $\mathcal{EL}^{++}$. To formulate it, let the *size* $|C|$ of a concept C and the site $|\mathcal{A}|$ of an ABox $\mathcal{A}$ be defined analogously to the size of CBoxes: it is simply the number of symbols needed to write down C and $\mathcal{A}$, respectively. Via the reductions of satisfiability and ABox consistency to subsumption, we obtain the following.

Theorem 9. *Let C and D be concepts, $\mathcal{A}$ an ABox, and $\mathcal{C}$ a CBox. Then the following holds:*

1. *If C is satisfiable w.r.t. $\mathcal{C}$, then C and $\mathcal{C}$ have a common model of size linear in $|C| + |\mathcal{C}|$;*
2. *if C is not subsumed by D w.r.t. $\mathcal{C}$, then there exists a model $\mathcal{I}$ of $\mathcal{C}$ of size linear in $|C| + |D| + |\mathcal{C}|$ such that $a \in C^{\mathcal{I}} \setminus D^{\mathcal{I}}$ for some $a \in \Delta^{\mathcal{I}}$;*
3. *If $\mathcal{A}$ is consistent w.r.t. $\mathcal{C}$, then $\mathcal{A}$ and $\mathcal{C}$ have a common model of size linear in $|\mathcal{A}| + |\mathcal{C}|$;*
4. *if an individual a is not an instance of C in $\mathcal{A}$ w.r.t. $\mathcal{C}$, then there exists a model $\mathcal{I}$ of $\mathcal{A}$ and $\mathcal{C}$ of size linear in $|C| + |\mathcal{A}| + |\mathcal{C}|$ such that $a^{\mathcal{I}} \notin C^{\mathcal{I}}$.*

4 P-admissible 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 8 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. To simplify notation, we call every finite conjunction of atomic formulae $p(f_1, \ldots, f_k)$ from a concrete domain $\mathcal{D}$ a *$\mathcal{D}$-conjunction.*

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 $\mathsf{conc}_w^{\mathsf{Q}} = \{(w', w'') \mid w'' = w'w\}$.

We now show that both Q and S are p-admissible.

Proposition 10. *The concrete domain* Q *is p-admissible.*

Proof. First for Point 1 of p-admissibility. Assume that, in Q-conjunctions, we admit the following additional predicates:

- a unary predicate $<_q$ for each $q \in \mathbb{Q}$ with $(P_<)^{\mathsf{Q}} = \{q' \in \mathbb{Q} \mid q' < q\}$;
- a binary predicate $<$ with the obvious extension.

If this extended set of predicates is available, we can reduce Q-implication to Q-satisfiability: assume that we want to decide whether a Q-conjunction c implies a formula $p(f_1, \ldots, f_k)$ with $p \in \mathcal{P}^{\mathsf{Q}}$. We make a case distinction according to p:

$=_q$ the implication holds if neither $c \wedge <_q(f_1)$ nor $c \wedge >_q(f_1)$ is satisfiable;

$>_q$ the implication holds if neither $c \wedge <_q(f_1)$ nor $c \wedge =_q(f_1)$ is satisfiable;

$=$ the implication holds if neither $c \wedge <(f_1, f_2)$ nor $c \wedge <(f_2, f_1)$ is satisfiable;

$+_q$ the implication holds if neither $c \wedge +_q(f_1, f) \wedge <(f, f_2)$ nor $c \wedge +_q(f_1, f) \wedge <(f_2, f)$ is satisfiable, where f is a feature name not appearing in c.

Using a straightforward reduction to linear programming, it is shown in [24] that satisfiability of Q-conjunctions using the extended set of predicates is decidable in polynomial time.

Now for Point 2 of p-admissibility. First, let c be a Q-conjunction, and let Γ be a finite set of formulae of the form $p(f_1, \dots, f_k)$ such that c implies no formula from Γ. Obviously, c must be satisfiable. Assume that c implies the disjunction over all formulae in Γ. W.l.o.g, we may assume that c does not contain conjuncts of the form $\top_{\mathsf{Q}}(f)$ since the conjunction that is obtained from c by dropping such conjuncts is equivalent to c. Moreover, the fact that c does not imply any formula from Γ means that Γ also contains no concepts of the form $\top_{\mathsf{Q}}(f)$. Our aim is to construct a solution δ for c such that $\delta \not\models C$ for all $C \in \Gamma$, in contradiction to our assumption. The construction is done in two steps: first, we define a solution for c that does not satisfy any formula $>_q(f) \in \Gamma$, and then we tweak this solution such that no other formulae from Γ are satisfied. For the first step, we start with defining a relation $\sim$ on the set of features $\mathsf{N_F}$ as follows:

$$f \sim f' \text{ iff } f = f' \text{ or } f \text{ and } f' \text{ occur jointly in a conjunct of } c.$$

Clearly, the transitive closure $\sim^*$ of $\sim$ is an equivalence relation. We now define, for each equivalence class Δ of $\sim^*$, a *distance function* d_Δ that takes each pair of features $f, f' \in \Delta$ to a rational number as follows:

- $d_\Delta(f, f) = 0$;
- $d_\Delta(f, f') = 0$ if $=(f, f') \in c$ or $=(f', f) \in c$;
- $d_\Delta(f, f') = q$ if $+_q(f, f') \in c$;
- $d_\Delta(f, f') = -q$ if $+_q(f', f) \in c$;
- $d_\Delta(f, f') = d_\Delta(f, f'') + d_\Delta(f'', f')$.

Note that d_Δ is total on Δ due to the definition of $\sim$ and well-defined since c is satisfiable. We say that a feature f is *fixed by c* if there exists a feature f' with $f \sim^* f'$ and $=_q(f') \in c$ for some $q \in \mathbb{Q}$. Observe that, for a given $\sim^*$-equivalence class, either all the features in the class are fixed or all are not fixed. Thus we can also talk about equivalence classes to be fixed.

Let $\Delta_1, \dots, \Delta_k$ be the equivalence classes of $\sim^*$. We define a solution δ_0 for c. This is done separately for each Δ_i, $1 \leq i \leq k$:

1. If Δ_i is fixed, then take a feature $f \in \Delta_i$ with $=_q(f) \in c$ and set $\delta_0(f) = q$. For all other features $f' \in \Delta_i$, set $\delta_0(f') = \delta_0(f) + d_{\Delta_i}(f, f')$.
2. If Δ_i is not fixed, then choose a feature $f \in \Delta_i$. Next, choose a value $\delta_0(f) \in \mathbb{Q}$ such that the following conditions are satisfied:
   - $\delta_0(f) + d_{\Delta_i}(f, f') > q$ for all $f' \in \Delta_i$ and all q with $>_q(f') \in c$.

- $\delta_0(f) + d_{\Delta_i}(f, f') \leq q$ for all $f' \in \Delta_i$ and all q with $>_q(f') \in \Gamma$.

For all other $f' \in \Delta_i$, set $\delta_0(f') = \delta_0(f) + d_{\Delta_i}(f, f')$.

To verify that a rational number $\delta_0(f)$ as required above indeed exists, let us assume the opposite. Then there is a $>_q(f') \in c$ and a $>_{q'}(f'') \in \Gamma$ such that $d_{\Delta_i}(f, f') - d_{\Delta_i}(f, f'') = d_{\Delta_i}(f', f'') \geq q - q'$. By definition of d_{Δ_i}, there is thus no solution δ_0 for c that does not satisfy $>_{q'}(f'') \in \Gamma$. This contradicts the fact that c does not imply any element of Γ.

Using the definition of d_Δ and of δ_0, it is readily checked that δ_0 is a solution for c satisfying none of the formulae $>_q(f)$ in Γ. The latter is obvious if $\delta(f)$ has been defined in Point 2 above. If it has been defined in Point 1, then $\delta(f) > q$ clearly yields that c implies $>_q(f)$, and thus $>_q(f) \notin \Gamma$.

Now for the second step, which deals with formulae $=_q(f)$, $=(f, f')$, and $+_q(f, f')$ in Γ that may be "accidentally" satisfied by δ_0. We destroy such satisfactions by carefully "shifting down" values of δ_0. To this end, choose a $b \in \mathbb{Q}$ such that the following conditions are satisfied:

1. $b > 0$;
2. for all conjuncts $>_q(f)$ of c, $b < \delta_0(f) - q$;
3. for all $=_q(f) \in \Gamma$ with $\delta_0(f) \neq q$, $b < |\delta_0(f) - q|$;
4. for all $=(f, f') \in \Gamma$ with $\delta_0(f) \neq \delta_0(f')$, $b < |\delta_0(f) - \delta_0(f')|$;
5. for all $+_q(f, f') \in \Gamma$ with $\delta_0(f') \neq \delta_0(f) + q$, $b < |\delta_0(f') - (\delta_0(f) + q)|$;

We define a new solution δ of c as follows:

$$\delta(f) := \begin{cases} \delta_0(f) - b & \text{if } f \text{ is not fixed by } c \\ \delta_0(f) & \text{otherwise} \end{cases}$$

It is not hard to show that δ is indeed a solution of c: conjuncts $>_q(f)$ are satisfied by choice of b (Point 2); conjuncts $=_q(f)$ are satisfied since they are satisfied by δ_0 and their presence implies that f is fixed by c; and conjuncts $=_((f, f')$ and $+_q(f, f')$ are satisfied since they are satisfied by δ_0 and their presence implies that f is fixed by c iff f' is fixed by c.

Moreover, the new solution δ does not satisfy any formula in Γ: formulae $>_q(f)$ have not been satisfied by δ_0, and we only shifted *down* when moving to δ; formulae of the other form are not satisfied by definition of δ and choice of b. ❏

Proposition 11. *The concrete domain* S *is p-admissible.*

Proof. First for Point 1 of p-admissibility. Consider the concrete domain $\mathsf{S}' = (\Sigma^*, \mathcal{P}^{\mathsf{S}'})$, with $\mathcal{P}^{\mathsf{S}'}$ containing the following properties:

1. a unary predicate $\top_{\mathsf{S}}$ as in S;
2. a unary predicate $=_\varepsilon$ as in S (but only for the empty word), and it's negation $\neq_\varepsilon$ with the obvious extension;
3. binary predicates $=$ and $\neq$ with the obvious extension;
4. binary predicate conc_w and $\overline{\mathsf{conc}_w}$ for each $w \in \Sigma^*$, where the extension of conc_w is as in S, and the extension of S' is complementary.

We claim that satisfiability and implication in S can be polynomially reduced to satisfiability in S':

- To check satisfiability of an S-conjunction c, first extend c with the conjunct $=_\varepsilon (e)$, where e is a feature name not occurring in c, and then replace each conjunct $=_w(f)$ in c with $w \neq \varepsilon$ by the conjunct $\mathsf{conc}_w(e, f)$. Finally, check satisfiability of the resulting conjunction c' in S'.
- To check whether an S-conjunction c implies a formula $p(f_1, \dots, f_n)$, first transform c into c' as in the satisfiability case above. If p is of the form $=_\varepsilon$, $=$, or conc_w, then simply check whether c' extended with the conjunct $\overline{p}(f_1, \dots, f_n)$ is unsatisfiable. If p is of the form $=_w$ with $w \neq \varepsilon$, then check whether c' extended with the conjunct $\mathsf{conc}_w(e, f_1)$ is unsatisfiable.

Since it is shown in [22] that satisfiability in S' are decidable in polynomial time, we thus obtain the same result for satisfiability and implication in S.

Now for Point 2 of p-admissibility. First, let c be an S-conjunction, and let Γ be a finite set of formulae of the form $p(f_1, \dots, f_k)$ such that c implies no formula from Γ. Again, in this case c is satisfiable. Assume that c implies the disjunction over all formulae in Γ. As in the case of the concrete domain Q, we may assume that the predicate $\top_{\mathsf{S}}(f)$ does not occur in c and Γ. Our aim is to construct a solution δ for c such that $\delta \not\models C$ for all $C \in \Gamma$, in contradiction to the assumption.

To this end, let δ_0 be an arbitrary solution for c. Let us tweak this solution such that no formula from Γ is satisfied. We start with defining a relation $\sim$ on the set of features $\mathsf{N_F}$ as follows:

$$f \sim f' \text{ iff } f = f' \text{ or } f \text{ and } f' \text{ occur jointly in a conjunct of } c.$$

The transitive closure $\sim^*$ of $\sim$ is an equivalence relation. We say that a feature f is *fixed by* c if there exists a feature f' with $f \sim^* f'$ and $=_w(f') \in c$ for some $w \in \Sigma^*$. Observe that, for a given $\sim^*$-equivalence class, either all the features in the class are fixed or all are not fixed. Thus we can also talk about equivalence classes to be (non-)fixed. Let $\alpha_1, \dots, \alpha_n$ denote the non-fixed equivalence classes of $\sim^*$. Then fix words $w_1, \dots, w_n \in \Sigma^*$ such that the following conditions are satisfied:

1. w_i is not a prefix of w, for $1 \leq i \leq n$ and $=_w$ a predicate occurring in Γ;

2. w_i is not a prefix of w_j, for $1 \leq i, j \leq n$ and $i \neq j$;

3. w_i is not a prefix of $\delta(f)$ for $1 \leq i \leq n$ and each $f \in \mathsf{N_F}$ occurring in c.

Now define a new solution δ of c as follows:

$$\delta(f) := \begin{cases} w_i \cdot \delta_0(f) & \text{if } f \in \alpha_i \\ \delta_0(f) & \text{if there is no such } \alpha_i \ . \end{cases}$$

It remains to show that $\delta \models c$ and $\delta \not\models C$ for all $C \in \Gamma$. For the former, we argue as follows:

- For all predicates in c of the form $=_w(f)$, f is, by definition, non-fixed. Thus $\delta(f) = \delta_0(f) = w$.
- All predicates in c of the form $=(f, g)$ remain satisfied because their existence implies that f and g are in the same equivalence class and thus $\delta(f) = w \cdot \delta_0(f)$ and $\delta(g) = w \cdot \delta_0(g)$ for some $w \in \Sigma^*$.
- All predicates $\mathsf{conc}_w(f, g)$ remain satisfied for the same reason.

Now for the latter.

- Consider some $=_w(f) \in \Gamma$. If f is is fixed and $\delta_0 \models =_w(f)$, then c implies $=_w(f)$ in contradiction to the assumption that c does not imply any element of Γ. Thus, either f is not fixed or $\delta_0 \not\models =_w(f)$. If f is not fixed, then $\delta \not\models =_w(f)$ because of Property 1 of the words $w_1, \ldots, w_n$. If f is fixed and $\delta_0 \not\models =_w(f)$, then we clearly also have $\delta \not\models =_w(f)$.
- Now consider $=(f, g) \in \Gamma$. If f and g are in the same equivalence class and $\delta_0 \models =(f, g)$ then c implies $=(f, g)$ contradicting our assumption. Thus either f and g are not in the same equivalence class or $\delta_0 \not\models =(f, g)$. In f and g are not in the same equivalence class, we have $\delta \not\models =(f, g)$ due to Properties 2 and 3 of the words $w_1, \ldots, w_n$. If they are in the same equivalence class and $\delta_0 \not\models =(f, g)$, then we clearly also have $\delta \not\models =(f, g)$.
- The case $\mathsf{conc}_w(f, g) \in \Gamma$ is analogous.

❏

Note that p-admissibility of concrete domains is easily broken. Consider e.g. the following examples:

- The concrete domain $\mathsf{Q}^{<_q,>_q}$ with domain $\mathbb{Q}$ that has the predicates $(>_q)_{q \in \mathbb{Q}}$ from Q and, additionally, unary predicates $(<_q)_{q \in \mathbb{Q}}$ with

  $$(<_q)^{\mathsf{Q}^{<_q,>_q}} := \{q' \in \mathbb{Q} \mid q' < q\}.$$

  Then the $\mathsf{Q}^{<_q,>_q}$-conjunction $c := >_0(f')$ does not imply any concept from $\Gamma := \{<_0(f), =_0(f), >_0(f)\}$, but every solution of c satisfies some concept of Γ.

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	$\forall r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \forall y \in \Delta^{\mathcal{I}} : (x, y) \in r^{\mathcal{I}} \to y \in C^{\mathcal{I}}\}$
at-least restriction	$(\geqslant n\ r)$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}}\} \geq n\}$
at-most restriction	$(\leqslant n\ r)$	$\{x \in \Delta^{\mathcal{I}} \mid \#\{y \in \Delta^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}}\} \leq n\}$
inverse roles	$\exists r^{-}.C$	$\{x \mid \exists y \in \Delta^{\mathcal{I}} : (y, x) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
role negation	$\exists \neg r.C$	$\{x \mid \exists y \in \Delta^{\mathcal{I}} : (y, x) \notin r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
role union	$\exists r \cup s.C$	$\{x \mid \exists y \in \Delta^{\mathcal{I}} : (y, x) \in r^{\mathcal{I}} \cup s^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
transitive closure	$\exists r^{*}.C$	$\{x \mid \exists y \in \Delta^{\mathcal{I}} : (y, x) \in (r^{\mathcal{I}})^{+} \wedge y \in C^{\mathcal{I}}\}$

Table 3: The additional constructors.

- Any concrete domain S^* with domain Σ^* for some finite alphabet Σ and the unary predicates pref_s and suff_s for every $s \in \Sigma^*$ with

  $$\begin{aligned} \mathsf{pref}_s^{\mathsf{S}^*} &:= \{s' \mid s \text{ is a prefix of } s'\} \\ \mathsf{suff}_s^{\mathsf{S}^*} &:= \{s' \mid s \text{ is a suffix of } s'\} \end{aligned}$$

  Assume $a \in \Sigma$. Then the S^*-conjunction $c := \mathsf{suff}_a(f)$ implies no formula from $\Gamma := \{\mathsf{pref}_\sigma(f) \mid \sigma \in \Sigma\}$, but every solution of c satisfies some formula from Γ.

- Any concrete domain S^* with domain Σ^* for some finite alphabet Σ, the unary predicates $\top_{\mathsf{S}^*}$ and $=_\varepsilon$ with the obvious semantics, and the unary predicates pref_s, $s \in \Sigma^*$, as in the previous example. Then the S^*-conjunction $c := \top_{\mathsf{S}^*}(f)$ implies no concept from $\Gamma := \{=_\varepsilon(f)\} \cup \{\mathsf{pref}_\sigma(f) \mid \sigma \in \Sigma\}$, but every solution of c satisfies some concept from Γ.

5 Lower Bounds

The purpose of this section is to justify our choice of constructors in the language $\mathcal{EL}^{++}$. To this end, 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 and $(r^{\mathcal{I}})^+$ denotes the transitive closure of the relation $r^{\mathcal{I}}$. As in the previous section, we can restrict the attention to satisfiability/subsumption of concept *names* w.r.t. general TBoxes.

Before considering concept and role constructors, we briefly discuss a natural extension of CBoxes: role inclusions can be strengthened to so-called *role-value-maps (RVMs)*, i.e., to inclusions $r_1 \circ \cdots \circ r_k \sqsubseteq s_1 \circ \cdots \circ s_\ell$ whose right-hand side is a *composition* of role names. The semantics of RVMs is defined in analogy with the semantics of $\mathcal{EL}^{++}$'s role inclusions. By a result of Baader [4], subsumption in $\mathcal{EL}$ is undecidable already if only RVMs, but no concept inclusions are admitted in CBoxes.

Theorem 12 (Baader). *Subsumption of $\mathcal{EL}$-concepts w.r.t. RVMs is undecidable.*

In the following, we walk through the constructors listed in Table 3 and, for each of them, prove that subsumption w.r.t. general TBoxes is not tractable.

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 [29] 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 13. *In $\mathcal{EL}^{(\neg)}$, satisfiability and subsumption w.r.t. general TBoxes is* ExpTime-*complete.*

For many other extensions of $\mathcal{EL}$ presented in this section, satisfiability is trivial in the sense that every concept is satisfiable w.r.t. every TBox. In the following, we will only explicitly mention satisfiability if it is *not* trivial.

Disjunction

Let $\mathcal{ELU}$ be the extension of $\mathcal{EL}$ with disjunction. Our aim is to show that subsumption in $\mathcal{ELU}$ w.r.t. general TBoxes is ExpTime-complete. The upper bound follows from $\mathcal{ELU}$ being a fragment of $\mathcal{ALC}$. For the lower bound, we reduce *satisfiability* of $\mathcal{EL}^{(\neg)}$-concepts w.r.t. general TBoxes to subsumption of $\mathcal{ELU}$-concepts. The former is ExpTime-hard by Theorem 13.

Let A_0 be an $\mathcal{EL}^{(\neg)}$ concept name and $\mathcal{T}$ a general $\mathcal{EL}^{(\neg)}$ TBox. To decide satisfiability of A_0 w.r.t. $\mathcal{T}$, take a new (i.e. distinct from A_0 and not occurring in $\mathcal{T}$) concept name A' for each concept name A occurring in $\mathcal{T}$. Also fix an additional new concept name L. Then the TBox $\mathcal{T}^*$ is obtained from $\mathcal{T}$ by first 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 concept name A occurring in $\mathcal{T}$;
- $\exists r.L \sqsubseteq L$.

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

Theorem 14. *In $\mathcal{ELU}$, subsumption w.r.t. general TBoxes is* ExpTime-*complete.*

This theorem improves upon the result of Brandt that subsumption of $\mathcal{ELU}$ concepts w.r.t. general TBoxes is NP-hard [7], and it improves upon the result of Hladik and Sattler that satisfiability of $\mathcal{ELU}$ concepts extended with functional roles and the bottom concept w.r.t. general TBoxes is ExpTime-hard [15].

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 [12]. 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. We assume that all concept inclusions in $\mathcal{T}$ have one of the following forms:

$$\begin{aligned} C &\sqsubseteq D \\ C_1 \sqcap C_2 &\sqsubseteq C \\ C &\sqsubseteq C_1 \sqcup C_2 \\ C &\sqsubseteq \exists r.D \\ \exists r.C &\sqsubseteq D \end{aligned}$$

where C, D, C_1, and C_2 are concept names or $\top$. It is easily verified that this assumption can be made without loss of generality since every general TBox can be converted into normal form using normalization rules similar to the one presented in Figure 1. Note in particular that $C_1 \sqcup C_2 \sqsubseteq C$ can be replaced by the two rules $C_1 \sqsubseteq C$ and $C_2 \sqsubseteq C$. To convert $\mathcal{T}$ into an $\mathcal{EL}^{\geq 2}$ CBox, we only need to rephrase concept implications of the form $C \sqsubseteq C_1 \sqcup C_2$. This is done as follows: introduce two new concept names A and B and a new role name r, and replace the mentioned implication with

$$\begin{array}{rcl} C & \sqsubseteq & \exists r.A \sqcap \exists r.B \\ C \sqcap \exists r.(A \sqcap B) & \sqsubseteq & C_1 \\ C \sqcap (\geqslant 2\ r) & \sqsubseteq & C_2 \end{array}$$

Call the resulting TBox $\mathcal{T}^*$. It is easily seen that $A_0 \sqsubseteq_{\mathcal{T}} B_0$ iff $A_0 \sqsubseteq_{\mathcal{T}^*} B_0$.

Theorem 15. *In $\mathcal{EL}^{\geq 2}$, subsumption w.r.t. general TBoxes is* ExpTime*-complete.*

Role Constructors $\neg$, $\cup$, $\cdot^*$

We consider the extension $\mathcal{EL}^{R\neg}$ of $\mathcal{EL}$ with role negation, $\mathcal{EL}^{\cup}$ with role union, and $\mathcal{EL}^*$ with transitive closure. For these three DLs, subsumption w.r.t. general TBoxes can be proved ExpTime-hard using a technique similar to the one employed for at-least restrictions in the previous section: the lower bounds are established by reducing subsumption in $\mathcal{ELU}$ w.r.t. general TBoxes. Thus, let A_0 and B_0 be concept names and $\mathcal{T}$ a general $\mathcal{ELU}$ TBox. As in the proof of Theorem 15, we assume that $\mathcal{T}$ is in normal form. We convert $\mathcal{T}$ into a new IBox $\mathcal{T}^*$ by replacing each concept inclusion $C \sqsubseteq C_1 \sqcup C_2$ as follows:

- In $\mathcal{EL}^{R\neg}$, we introduce a new concept name A and two new role names r and s. Then we replace the above inclusion by the following:

$$\begin{array}{rcl} C & \sqsubseteq & \exists r.A \\ C \sqcap \exists s.A & \sqsubseteq & C_1 \\ C \sqcap \exists \neg s.A & \sqsubseteq & C_2 \end{array}$$

- In $\mathcal{EL}^{\cup}$, we introduce a new concept name A and two new role names r and s. Then we replace the above inclusion by the following:

$$\begin{array}{rcl} C & \sqsubseteq & \exists r \cup s.A \\ C \sqcap \exists r.A & \sqsubseteq & C_1 \\ C \sqcap \exists s.A & \sqsubseteq & C_2 \end{array}$$

- In $\mathcal{EL}^{*}$, we introduce a new concept name A and a new role name r. Then we replace the above inclusion by the following:

$$\begin{array}{rcl} C & \sqsubseteq & \exists r^{+}.A \\ C \sqcap \exists r.A & \sqsubseteq & C_1 \\ C \sqcap \exists r.\exists r^{+}.A & \sqsubseteq & C_2 \end{array}$$

The ExpTime upper bound is obtained from the fact that in $\mathcal{ALC}$ extended with the Boolean operators on roles, subsumption w.r.t. general TBoxes is in ExpTime [25], and the same holds for the description logic $\mathcal{ALC}_{\mathsf{reg}}$ [11, 29, 2].

Theorem 16. *In $\mathcal{EL}R\neg$, $\mathcal{EL}^{\cup}$, and $\mathcal{EL}^{*}$, subsumption w.r.t. general TBoxes is* ExpTime*-complete.*

Non-p-admissible Concrete Domains

We now show that 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 $\mathcal{D}$, i.e., with features f that are mapped to partial functions from $\Delta^{\mathcal{I}}$ to $\Delta^{\mathcal{D}}$, and with a concept constructor $p(f_1, \dots, f_k)$ for each k-ary predicate $p \in \mathcal{P}^{\mathcal{D}}$.

To prove ExpTime-hardness, we first strengthen Theorem 14 as follows. Let a *single-disjunction TBox (sd-TBox)* be a general $\mathcal{EL}$ TBox that, additionally, contains zero or one concept implication of the form $A \sqsubseteq B_1 \sqcup B_2$ with A, B_1, and B_2 concept names. We show that subsumption of $\mathcal{EL}$-concepts w.r.t. sd-TBoxes is ExpTime-complete. The lower bound is proved by reduction of subsumption in $\mathcal{ELU}$ w.r.t. general TBoxes, which is ExpTime-hard by Theorem 14. Thus, let A_0 and B_0 be concept names and $\mathcal{T}$ a general $\mathcal{ELU}$ TBox. We again assume that $\mathcal{T}$ is in the usual normal form introduced above. For the reduction, introduce new concept names U and U' and a new role name $r_{A,B,B'}$ for each concept implication $A \sqsubseteq B \sqcup B' \in \mathcal{T}$. We convert $\mathcal{T}$ into a sd-TBox $\mathcal{T}^{*}$

by adding the concept implication $\top \sqsubseteq U \sqcup U'$. and replacing each concept implication $A \sqsubseteq B \sqcup B'$ with

$$\begin{array}{rcl} \top & \sqsubseteq & \exists r_{A,B,B'}.\top \\ A \sqcap \exists r_{A,B,B'}.U & \sqsubseteq & B \\ A \sqcap \exists r_{A,B,B'}.U' & \sqsubseteq & B' \end{array}$$

It is easy to check that $A_0 \sqsubseteq_{\mathcal{T}} B_0$ iff $A_0 \sqsubseteq_{\mathcal{T}^*} B_0$. Together with the upper bound from Theorem 13, we thus obtain the following:

Theorem 17. *Subsumption of $\mathcal{EL}$-concepts w.r.t. sd-TBoxes is* ExpTime*-complete.*

Now back to the ExpTime-hardness of subsumption in $\mathcal{EL}(\mathcal{D})$, where $\mathcal{D}$ is a non-convex concrete domain. We reduce subsumption in $\mathcal{EL}$ w.r.t. sd-TBoxes. Let A_0 and B_0 be concept names and $\mathcal{T}$ an sd-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 18. *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 concrete domains introduced at the end of Section 4. We obtain the following corollary.

Corollary 19. *For the following concrete domains $\mathcal{D}$, subsumption in $\mathcal{EL}(\mathcal{D})$ w.r.t. general TBoxes is* ExpTime*-hard:*

- *the concrete domain $\mathsf{Q}^{<_q,>_q}$;*
- *any concrete domain S^* with domain Σ^* for some finite alphabet Σ and the unary predicates pref_s and suff_s for every $s \in \Sigma^*$;*
- *any concrete domain S^* with domain Σ^* for some finite alphabet Σ, the unary predicates $\top_{\mathsf{S}^*}$ and $=_\varepsilon$, and the unary predicates pref_s, for each $s \in \Sigma^*$.*

Using results from [23], a matching upper bound can be obtainted for the case where $\mathcal{D}$-satisfiability is in ExpTime. This is e.g. the case for the first item of Corollary 19 [24].

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 the description logic $\mathcal{ALE}$ w.r.t. so-called primitive TBoxes:

- $\mathcal{ALE}$ is obtained by extending $\mathcal{EL}^\forall$ with atomic negation;
- *primitive TBoxes* are general TBoxes whose concept inclusions have the form $A \sqsubseteq C$, with A a concept name.

It has been shown by Calvanese that satisfiability in $\mathcal{ALE}$ w.r.t. primitive TBoxes is PSpace-complete [8].

Let A_0 be a concept name, and $\mathcal{T}$ a primitive $\mathcal{ALE}$ TBox. We assume that $\mathcal{T}$ is in normal form: every concept inclusion is of one of the following forms:

$$\begin{aligned} &A \sqsubseteq B \\ &A \sqsubseteq \neg B \\ &A \sqsubseteq B \sqcap B' \\ &A \sqsubseteq \exists r.B \\ &A \sqsubseteq \forall r.B \end{aligned}$$

where A, B, and B' are concept names. It is easily verified that this assumption can be made without loss of generality since every primitive TBox can be converted into normal form using normalization rules similar to the one presented in Figure 1.

For the reduction, we take a new concept name L and define a general $\mathcal{ELI}$ TBox $\mathcal{T}^*$ containing the following concept inclusions:

- $A \sqsubseteq D$ 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 the subsumption $A_0 \sqsubseteq_{\mathcal{T}^*} L$. 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. $\mathcal{T}$ if, and only if, $A_0 \not\sqsubseteq_{\mathcal{C}} L$.

Theorem 20. *in $\mathcal{ELI}$, subsumption 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 $\mathcal{ALCI}$ [12]).

At-Most Restrictions

Let $\mathcal{EL}^{\leq 1}$ be the extension of $\mathcal{EL}$ with at-most restrictions of the form $(\leqslant 1\ r)$. 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 by Toman and Wedell: as noted below Corollary 12 of [33], this is an immediate consequence of the proof of Theorem 11 in the same paper. Note that $\mathcal{FL}_0$ is often assumed to additionally offer the $\top$-concept. For our purposes, it is simpler to exlude it. This is justified by the fact that ExpTime-hardness of subsumption in $\mathcal{FL}_0^{tf}$ also does not presuppose the presence of the $\top$-concept as well.

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 $\forall r.C$ appearing on the left-hand side of a GCI with $\exists r.C$, and each subconcept $\forall r.C$ appearing on the right-hand side of a GCI with $(\leqslant 1\ r) \sqcap \exists r.C$. Then the following holds:

Lemma 21. *$A_0 \sqsubseteq_{\mathcal{T}} B_0$ iff $A_0 \sqsubseteq_{\mathcal{T}^*} B_0$.*

Proof. We show the contrapositives of both directions. First assume that $A_0 \not\sqsubseteq_{\mathcal{T}^*} B_0$, i.e., there is an $\mathcal{EL}^{\leq 1}$ model $\mathcal{I}$ of $\mathcal{T}^*$ and an $x_0 \in A_0^{\mathcal{I}} \setminus B_0^{\mathcal{I}}$. First modify $\mathcal{I}$ to a model $\mathcal{I}'$ as follows: for each $x \in \Delta^{\mathcal{I}}$ and each role name r such that $|\{y \in \Delta^{\mathcal{I}} \mid (x,y) \in r^{\mathcal{I}}\}| > 1$, delete all out-going r-edges starting at x. To show that $\mathcal{I}'$ is still a model of $\mathcal{T}^*$, it suffices to prove that $C^{\mathcal{I}'} \subseteq C^{\mathcal{I}}$ for concepts C appearing on the left-hand side of GCIs, and $C^{\mathcal{I}'} = C^{\mathcal{I}}$ for concepts C' appearing on the right-hand side of GCIs. The former is easy since left-hand sides of GCIs are $\mathcal{EL}$ concepts, and the latter is easy since existential restrictions $\exists r.C$ on right-hand sides of GCIs are *always* conjoined with $(\leqslant 1\ r)$. Clearly, all role names are interpreted as partial functions in $\mathcal{I}'$. We now extend these interpretations to total functions: Let $\Delta^{\mathcal{I}''}$ be defined as follows:

$$\begin{array}{rcll}
\Delta^{\mathcal{I}''} & := & \Delta^{\mathcal{I}'} \uplus \{x_\perp\} & \\
A^{\mathcal{I}''} & := & A^{\mathcal{I}'} & \text{for all concept names } A \\
r^{\mathcal{I}''} & := & r^{\mathcal{I}'} \cup \{(x,x_\perp) \mid (x,y) \notin r^{\mathcal{I}'} \text{ for all } y \in \Delta^{\mathcal{I}'}\} & \text{for all role names } r.
\end{array}$$

Observe that the new domain element $x_\perp$ is in the extension of no concept name. It is readily checked that $x \in C^{\mathcal{I}'}$ iff $x \in C^{\mathcal{I}''}$ for all $x \in \Delta^{\mathcal{I}}$ and $\mathcal{EL}$ concepts C not using the $\top$ concept. Thus, $\mathcal{I}''$ is still a model of $\mathcal{T}^*$. Since role names are interpreted as total functions, it is easy to show by structural induction that

$$C^{\mathcal{I}''} = (C^\dagger)^{\mathcal{I}''} = (C^\ddagger)^{\mathcal{I}''} \qquad (*)$$

for every $\mathcal{FL}_0^{tf}$ concept C, where $C^\dagger$ denotes the result of replacing each subconcept $\forall r.C$ with $\exists r.C$, and $C^\ddagger$ denotes the result of replacing each subconcept $\forall r.C$ with

$(\leqslant 1\ r) \sqcap \exists r.C$. Since $\mathcal{I}''$ is a model of $\mathcal{T}^*$, this clearly yields that $\mathcal{I}''$ is a model of $\mathcal{T}$ as well. Since $x_0 \in A_0^{\mathcal{I}''} \setminus B_0^{\mathcal{I}''}$, we have $A_0 \not\sqsubseteq_{\mathcal{T}} B_0$ as required.

Now for the other direction. Assume that $A_0 \not\sqsubseteq_{\mathcal{T}} B_0$, i.e., there is an $\mathcal{FL}_0^{tf}$ model $\mathcal{I}$ of $\mathcal{T}$ and an $x_0 \in A_0^{\mathcal{I}} \setminus B_0^{\mathcal{I}}$. As in the direction considered first, the fact that all relations are interpreted as total functions implies that we have $x \in (\exists r.C)^{\mathcal{I}}$ iff $x \in (\forall r.C)^{\mathcal{I}}$ for all $x \in \Delta^{\mathcal{I}}$, $\mathcal{EL}$ concepts C, and role names r. Since $\mathcal{I}$ is a model of $\mathcal{T}$, it is thus also a model of $\mathcal{T}^*$ yielding $A_0 \not\sqsubseteq_{\mathcal{T}^*} B_0$ as required. ❏

We thus obtain the following theorem:

Theorem 22. *Subsumption in $\mathcal{EL}^{\leq 1}$ w.r.t. general TBoxes is* ExpTime*-complete.*

Ana analogous result can be proved if the concept constructor $(\leqslant 1\ r)$ is replaced by CBox assertions $\mathsf{Funct}(r)$, which are satisfied by an interpretation $\mathcal{I}$ if $r^{\mathcal{I}}$ is a partial function.

6 Comparison with $\mathcal{FL}_0$

The purpose of this paper is to investigate the complexity of reasoning w.r.t. general TBoxes in extensions of the basic description logic $\mathcal{EL}$. To fully appreciate the result from [7] that subsumption in $\mathcal{EL}$ w.r.t. general TBoxes is polynomial and the results from this paper that polynomiality is even preserved for several extensions of $\mathcal{EL}$, it is worthwhile to compare the computational complexity of $\mathcal{EL}$ with that of it's sibling DL $\mathcal{FL}_0$ providing only for the concept constructors $\sqcap$ and $\forall r.C$. Such a comparison is performed in this section yielding the result that, although looking just as harmless as $\mathcal{EL}$, $\mathcal{FL}_0$ is much less robust than $\mathcal{EL}$ w.r.t. the addition of TBox formalisms: we prove that subsumption in $\mathcal{FL}_0$ with general TBoxes is ExpTime-complete.

Summing up the work on $\mathcal{EL}$ carried out in this and previous papers, we obtain the following picture:

- Subsumption of $\mathcal{EL}$ concepts without reference to a TBox is of polynomial complexity [5].

- Subsumption in $\mathcal{EL}$ w.r.t. *standard TBoxes* is still of polynomial complexity, where a standard TBox is a finite set of concept definitions $A \doteq C$ with A a concept name. Such a concept definition is satisfied by an interpretation $\mathcal{I}$ if $A^{\mathcal{I}} = C^{\mathcal{I}}$ [4].

- Subsumption in $\mathcal{EL}$ w.r.t. general TBoxes is still of polynomial complexity [7].

- Even for several extensions of $\mathcal{EL}$, subsumption w.r.t. general TBoxes remains polynomial (current paper).

When we summarize the work on $\mathcal{FL}_0$, we obtain a dramatically different picture: in this case, adding a more powerful TBox formalism usually results in an increase of the complexity of reasoning:

- Subsumption of $\mathcal{FL}_0$ concepts without reference to a TBox is of polynomial complexity [6].

- Subsumption in $\mathcal{FL}_0$ w.r.t. acyclic TBoxes is co-NP-complete, where an acyclic TBox is a standard TBox that does not contain concept definitions

$$A_0 \doteq C_0, \ldots, A_{k_1} \doteq C_{k-1}$$

  such that $A_{i+1 \mod k}$ is used in C_i for all $i < k$ [27].

- Subsumption in $\mathcal{FL}_0$ w.r.t. (possibly cyclic) standard TBoxes is PSpace-complete [1, 3, 21].

To complete the picture for $\mathcal{FL}_0$ and to illustrate that the robust computational behavior of $\mathcal{EL}$ is rather surprising, in the following we prove that subsumption in $\mathcal{FL}_0$ w.r.t. general TBoxes is ExpTime-complete. As containment in ExpTime follows from the fact that subsumption in $\mathcal{ALC}$ w.r.t. general TBoxes is in ExpTime, it remains to establish the lower bound.

The proof is by a reduction of subsumption in $\mathcal{FL}_0^{tf}$ w.r.t. general TBoxes. As has already been mentioned, this problem is ExpTime-complete by results from [33]. Let A_0 and B_0 be concept names and $\mathcal{T}$ a general $\mathcal{FL}_0^{tf}$ TBox. For simplicity, we assume that $\mathcal{T}$ is in normal form, i.e., it only contains concept definitions of the following forms:

$$\begin{array}{rcl} A & \sqsubseteq & B \\ A & \sqsubseteq & \forall r.B \\ A \sqcap A' & \sqsubseteq & B \\ \forall r.A & \sqsubseteq & B \end{array}$$

where A, A', B, and B' are concept names. It is not hard to verify that every general $\mathcal{FL}_0^{tf}$ TBox can be converted into normal form in polynomial time by using normalization rules similar to the one presented in Figure 1. Then we have the following:

Lemma 23. *$A_0 \sqsubseteq_\mathcal{T} B_0$ in $\mathcal{FL}_0$ iff $A_0 \sqsubseteq_\mathcal{T} B_0$ in $\mathcal{FL}_0^{tf}$.*

Proof. The "only if" direction is trivial: consider the contrapositive, i.e., $A_0 \not\sqsubseteq_\mathcal{T} B_0$ in $\mathcal{FL}_0^{tf}$ implies $A_0 \not\sqsubseteq_\mathcal{T} B_0$ in $\mathcal{FL}_0$. As every interpretation witnessing the former is also a witness for the latter, there is nothing to be done.

Now for the (contrapositive of the) "if" direction. Assume that $A_0 \not\sqsubseteq_\mathcal{T} B_0$, i.e., there is an $\mathcal{FL}_0$ model $\mathcal{I}$ of $\mathcal{T}$ and an $x_0 \in A_0^\mathcal{I} \setminus B_0^\mathcal{I}$. We show how to convert $\mathcal{I}$ into an $\mathcal{FL}_0^{tf}$ interpretation witnessing $A_0 \not\sqsubseteq_\mathcal{T} B_0$ in $\mathcal{FL}_0^{tf}$. Let $\mathsf{R}_\mathcal{T}$ denote the set of role names occurring in $\mathcal{T}$, and $\mathcal{S}$ the set of all sequences of role names from $\mathsf{R}_\mathcal{T}$, including the empty sequence ε. For each $S \in \mathcal{S}$ and $x \in \Delta^\mathcal{I}$, we use $S^\mathcal{I}(x)$ to denote the set $\{y \in \Delta^\mathcal{I} \mid (x, y) \in S^\mathcal{I}\}$, where $S^\mathcal{I}$ is defined in the obvious way using composition of relations (and $\varepsilon^\mathcal{I}(x) = \{x\}$). Now we construct a $\mathcal{FL}_0^{tf}$ interpretation $\mathcal{J}$ as follows:

- $\Delta^\mathcal{J} = \mathcal{S}$.
- $A^\mathcal{J} := \{S \mid S^\mathcal{I}(x_0) \subseteq A^\mathcal{I}\}$;
- $r^\mathcal{J} := \{(S, S') \mid S' = Sr\}$ for all $r \in \mathsf{R}_\mathcal{T}$;

By definition of $\mathcal{J}$, we have that $\varepsilon \in A_0^{\mathcal{J}} \setminus B_0^{\mathcal{J}}$. It is readily checked that all role names are interpreted as total functions. To show that $A_0 \not\sqsubseteq_{\mathcal{T}} B_0$, it thus remains to show that $\mathcal{J}$ satisfies all concept inclusions in $\mathcal{T}$:

1. $A \sqsubseteq B$. Let $S \in A^{\mathcal{J}}$. Then $S^{\mathcal{I}}(x_0) \subseteq A^{\mathcal{I}}$. Since $\mathcal{I}$ satisfies $A \doteq B$, this yields $S^{\mathcal{I}}(x_0) \subseteq B^{\mathcal{I}}$ and thus $S \in B^{\mathcal{J}}$ as required.

2. $A \sqcap A' \sqsubseteq B$. Similar to the previous case.

3. $\forall r.A \sqsubseteq B$. Let $S \in (\forall r.A)^{\mathcal{J}}$. Then $Sr \in A^{\mathcal{J}}$. Thus, $Sr^{\mathcal{I}}(x_0) \subseteq A^{\mathcal{I}}$ implying $S^{\mathcal{I}}(x_0) \subseteq (\forall r.A)^{\mathcal{I}}$. Hence, $S^{\mathcal{I}}(x_0) \subseteq B^{\mathcal{I}}$ since $\mathcal{I}$ satisfies $\forall r.A \sqsubseteq B$. It follows that $S \in B^{\mathcal{J}}$ as required.

4. $A \sqsubseteq \forall r.B$. Let $S \in A^{\mathcal{J}}$. Then $S^{\mathcal{I}}(x_0) \subseteq A^{\mathcal{I}}$, and, since $\mathcal{I}$ satisfies $A \sqsubseteq \forall r.B$, we have $Sr^{\mathcal{I}}(x_0) \subseteq B^{\mathcal{I}}$. It follows that $Sr \in B^{\mathcal{J}}$ implying $S \in (\forall r.B)^{\mathcal{J}}$ since Sr is the only r-successor of S in $\mathcal{J}$.

❑

We thus obtain the following theorem, improving a result from [13] which states that subsumption in $\mathcal{EL}$ extended with value restrictions is ExpTime-complete.

Theorem 24. *Subsumption in $\mathcal{FL}_0$ w.r.t. general TBoxes is* ExpTime*-complete.*

Thus, subsumption w.r.t. general TBoxes is polynomial in the fragment $\mathcal{EL}$ of $\mathcal{ALC}$, but it is ExpTime-complete in the equally harmless looking fragment $\mathcal{FL}_0$—and thus just as hard as subsumption in full $\mathcal{ALC}$. In parallel to our work, Theorem 24 was independently proved by Martin Hofmann using a reduction of (the existence of winning stratiegies in) pushdown games [16].

7 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.

References

[1] F. Baader. Terminological cycles in KL-ONE-based knowledge representation languages. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 621–626, Boston, MA, USA, 1990.

[2] F. Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 446–451, Sydney, Australia, 1991.

[3] F. Baader. Using automata theory for characterizing the semantics of terminological cycles. *Annals of Mathematics and Artificial Intelligence*, 18(2–4):175–219, 1996.

[4] F. Baader. Terminological cycles in a description logic with existential restrictions. In G. Gottlob and T. Walsh, editors, *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 325–330. Morgan Kaufmann, 2003.

[5] F. Baader, R. Küsters, and R. Molitor. Computing least common subsumers in description logics with existential restrictions. In T. Dean, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI'99)*, pages 96–101. Morgan Kaufmann, 1999.

[6] R. J. Brachman and H. J. Levesque. The tractability of subsumption in frame-based description languages. In *Proc. of the 4th Nat. Conf. on Artificial Intelligence (AAAI'84)*, pages 34–37, 1984.

[7] S. Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In R. L. de Mantáras and L. Saitta, editors, *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-2004)*, pages 298–302. IOS Press, 2004.

[8] D. Calvanese. Reasoning with inclusion axioms in description logics: Algorithms and complexity. In *Proceedings of the Twelfth European Conference on Artificial Intelligence (ECAI-96)*, pages 303–307, 1996.

[9] R. Cote, D. Rothwell, J. Palotay, R. Beckett, and L. Brochu. The systematized nomenclature of human and veterinary medicine. Technical report, SNOMED International, Northfield, IL: College of American Pathologists, 1993.

[10] F. M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The complexity of concept languages. In J. Allen, R. Fikes, and E. Sandewall, editors, *Proc. of the 2nd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR'91)*, pages 151–162. Morgan Kaufman, 1991.

[11] M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *J. Comput. Syst. Sci.*, 18:194–211, 1979.

[12] G. D. Giacomo and M. Lenzerini. Boosting the correspondence between description logics and propositional dynamic logics. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI'94). Volume 1*, pages 205–212. AAAI Press, 1994.

[13] R. Givan, D. Kozen, D. McAllester, and C. Witty. Tarskian set constraints. Manuscript, 1996. Abriged version published in: Proceedings of the 11$^{\text{th}}$ Annual IEEE Symposium on Logic in Computer Science, 1996.

[14] V. Haarslev and R. Möller. RACER system description. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proceedings of the First International Joint Conference on Automated Reasoning (IJCAR'01)*, number 2083 in Lecture Notes in Artifical Intelligence, pages 701–705. Springer-Verlag, 2001.

[15] J. Hladik and U. Sattler. A translation of looping alternating automata to description logics. In *Proc. of the 19th Conference on Automated Deduction (CADE-19)*, volume 2741 of *Lecture Notes in Artificial Intelligence*. Springer Verlag, 2003.

[16] M. Hofmann. Personal communication.

[17] I. Horrocks. Using an expressive description logic: Fact or fiction? In *Proceedings of the Sixth International Conference on the Principles of Knowledge Representation and Reasoning (KR98)*, pages 636–647, 1998.

[18] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[19] I. Horrocks and U. Sattler. Decidability of shiq with complex role inclusion axioms. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI-2003)*, pages 343–348. Morgan-Kaufmann Publishers, 2003.

[20] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3):239–264, 2000.

[21] 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 E. F. Diego Calvanese, Giuseppe De Giacomo, editor, *Proceedings of the International Workshop in Description Logics 2003 (DL2003)*, number 81 in CEUR-WS (http://ceur-ws.org/), 2003.

[22] C. Lutz. NExpTime-complete description logics with concrete domains. LTCS-Report 00-01, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 2000. See http://www-lti.informatik.rwth-aachen.de/Forschung/Reports.html.

[23] C. Lutz. *The Complexity of Reasoning with Concrete Domains.* PhD thesis, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 2002.

[24] C. Lutz. Description logics with concrete domains—a survey. In F. W. Philippe Balbiani, Nobu-Yuki Suzuki and M. Zakharyaschev, editors, *Advances in Modal Logics Volume 4*, pages 265–296. King's College Publications, 2003.

[25] C. Lutz and U. Sattler. Mary likes all cats. In F. Baader and U. Sattler, editors, *Proceedings of the 2000 International Workshop in Description Logics (DL2000)*, number 33 in CEUR-WS (http://ceur-ws.org/), pages 213–226, 2000.

[26] B. Nebel. Computational complexity of terminological reasoning in BACK. *Artificial Intelligence*, 34(3):371–383, 1988.

[27] B. Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43:235–249, 1990.

[28] A. Rector and I. Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proceedings of the Workshop on Ontological Engineering, AAAI Spring Symposium (AAAI'97)*, Stanford, CA, 1997. AAAI Press.

[29] K. D. Schild. A correspondence theory for terminological logics: Preliminary report. In J. Mylopoulos and R. Reiter, editors, *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 466–471. Morgan Kaufmann, 1991.

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

[31] K. Spackman, K. Campbell, and R. Cote. SNOMED RT: A reference terminology for health care. pages 640–644, 1997. Fall Symposium Supplement.

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

[33] D. Toman and G. Weddell. On reasoning about structural equality in xml: A description logic approach. *Theoretical Computer Science*. To appear, available from http://db.uwaterloo.ca/∼david/papers.html.