Terminological Cycles in a Description Logic with Existential Restrictions*

Franz Baader
Theoretical Computer Science
TU Dresden
D-01062 Dresden, Germany
baader@inf.tu-dresden.de

Abstract

Cyclic definitions in description logics have until now been investigated only for description logics allowing for value restrictions. Even for the most basic language $\mathcal{FL}_0$, which allows for conjunction and value restrictions only, deciding subsumption in the presence of terminological cycles is a PSPACE-complete problem. This paper investigates subsumption in the presence of terminological cycles for the language $\mathcal{EL}$, which allows for conjunction, existential restrictions, and the top-concept. In contrast to the results for $\mathcal{FL}_0$, subsumption in $\mathcal{EL}$ remains polynomial, independent of whether we use least fixpoint semantics, greatest fixpoint semantics, or descriptive semantics.

1 Introduction

Early description logic (DL) systems allowed the use of value restrictions ($\forall r.C$), but not of existential restrictions ($\exists r.C$). Thus, one could express that all children are male using the value restriction $\forall$child.Male, but not that someone has a son using the existential restriction $\exists$child.Male. 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 (see, e.g., [Nebel, 1990]). Once one considers more expressive DLs allowing for full negation, existential restrictions come in as the dual of value restrictions [Schmidt-Schauß and Smolka, 1991]. Thus, for historical reasons, DLs that allow for existential, but not for value restrictions, are largely unexplored. In the present paper, we investigate terminological cycles in the DL $\mathcal{EL}$, which allows for conjunction, existential restrictions, and the top-concept. In contrast to (even very inexpressive) DLs with value restrictions, subsumption in $\mathcal{EL}$ remains polynomial in the presence of terminological cycles. It should be noted that there are indeed applications where the small DL $\mathcal{EL}$ appears to be sufficient. In fact, SNOMED, the Systematized Nomenclature of Medicine [Cote *et al.*, 1993] employs $\mathcal{EL}$ [Spackman, 2001; Spackman *et al.*, 1997]. Even though SNOMED does not appear to use cyclic definitions, this may be due to a lack of technology rather than need. In fact, the Galen medical knowledge base contains many cyclic dependencies [Rector and Horrocks, 1997]. Also, even in the case of acyclic terminologies, our polynomial subsumption algorithm improves on the usual approach that first unfolds the TBox (a potentially exponential step) and then applies the polynomial subsumption algorithm for $\mathcal{EL}$-concept descriptions [Baader *et al.*, 1999].

The first thorough investigation of cyclic terminologies in description logics (DL) is due to Nebel [1991], who introduced three different semantics for such terminologies: least fixpoint (lfp) semantics, which considers only the models that interpret the defined concepts as small as possible; greatest fixpoint (gfp) semantics, which considers only the models that interpret the defined concepts as large as possible; and descriptive semantics, which considers all models.

In [Baader, 1990; 1996], subsumption w.r.t. cyclic terminologies in the small DL $\mathcal{FL}_0$, which allows for conjunction and value restrictions only, was characterized with the help of finite automata. This characterization provided PSPACE decision procedures for subsumption in $\mathcal{FL}_0$ with cyclic terminologies for the three types of semantics introduced by Nebel. In addition, it was shown that subsumption is PSPACE-hard. The results for cyclic $\mathcal{FL}_0$-terminologies were extended by Küsters [1998] to $\mathcal{ALN}$, which extends $\mathcal{FL}_0$ by atomic negation and number restrictions.

The fact that the DL $\mathcal{ALC}$ (which extends $\mathcal{FL}_0$ by full negation) is a syntactic variant of the multi-modal logic K opens a way for treating cyclic terminologies and more general recursive definitions in more expressive languages like $\mathcal{ALC}$ and extensions thereof by a reduction to the modal mu-calculus [Schild, 1994; De Giacomo and Lenzerini, 1994]. In this setting, one can use a mix of the three types of semantics introduced by Nebel. However, the complexity of the subsumption problem is EXPTIME-complete.

In spite of these very general results for cyclic definitions in expressive languages, there are still good reasons to look at cyclic terminologies in less expressive (in particular sub-Boolean) description logics. One reason is, of course, that one can hope for a lower complexity of the subsumption problem. For DLs with value restrictions, this hope is not fulfilled, though. Even in the inexpressive DL $\mathcal{FL}_0$, subsumption becomes PSPACE-complete if one allows for cyclic definitions. This is still better than the EXPTIME-completeness that one

*Partially supported by the DFG under grant BA 1122/4-3.

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

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

has in $\mathcal{ALC}$ with cyclic definitions, but from the practical point of view it still means that the subsumption algorithm may need exponential time.

In contrast, the subsumption problem in $\mathcal{EL}$ can be decided in polynomial time w.r.t. the three types of semantics introduced by Nebel. The main tool used to show these results is a characterization of subsumption through the existence of so-called simulation relations.

In the next section we will introduce the DL $\mathcal{EL}$ as well as cyclic terminologies and the three types of semantics for these terminologies. Then we will show in Section 3 how such terminologies can be translated into description graphs. In this section, we will also define the notion of a simulation between nodes of a description graph, and mention some useful properties of simulations. The next three sections are then devoted to the characterization of subsumption in $\mathcal{EL}$ w.r.t. gfp, lfp, and descriptive semantics, respectively.

2 Cyclic terminologies in the DL $\mathcal{EL}$

Concept descriptions are inductively defined with the help of a set of *constructors*, starting with a set N_C of *concept names* and a set N_R of *role names*. The constructors determine the expressive power of the DL. We restrict the attention to the DL $\mathcal{EL}$, whose concept descriptions are formed using the constructors top-concept ($\top$), conjunction ($C \sqcap D$), and existential restriction ($\exists r.C$). The semantics of $\mathcal{EL}$-concept descriptions is defined in terms of an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. The domain $\Delta^{\mathcal{I}}$ of $\mathcal{I}$ is a non-empty set of individuals and the interpretation function $\cdot^{\mathcal{I}}$ maps each concept name $A \in N_C$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and each role $r \in N_R$ to a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$. The extension of $\cdot^{\mathcal{I}}$ to arbitrary concept descriptions is inductively defined, as shown in the third column of Table 1.

A *terminology* (or *TBox* for short) is a finite set of concept definitions of the form $A \equiv D$, where A is a concept name and D a concept description. In addition, we require that TBoxes do not contain *multiple definitions*, i.e., there cannot be two distinct concept descriptions D_1 and D_2 such that both $A \equiv D_1$ and $A \equiv D_2$ belongs to the TBox. Concept names occurring on the left-hand side of a definition are called *defined concepts*. All other concept names occurring in the TBox are called *primitive concepts*. Note that we allow for cyclic dependencies between the defined concepts, i.e., the definition of A may refer (directly or indirectly) to A itself. An interpretation $\mathcal{I}$ is a model of the TBox $\mathcal{T}$ iff it satisfies all its concept definitions, i.e., $A^{\mathcal{I}} = D^{\mathcal{I}}$ for all definitions $A \equiv D$ in $\mathcal{T}$.

The semantics of (possibly cyclic) $\mathcal{EL}$-TBoxes we have just defined is called *descriptive semantic* by Nebel [1991]. For some applications, it is more appropriate to interpret cyclic concept definitions with the help of an appropriate fixpoint semantics. Before defining least and greatest fixpoint semantics formally, let us illustrate their effect on an example.

Example 1 Assume that our interpretations are graphs where we have nodes (elements of the concept name Node) and edges (represented by the role edge), and we want to define the concept Inode of all nodes lying on an infinite (possibly cyclic) path of the graph. The following is a possible definition of Inode: $\mathsf{Inode} \equiv \mathsf{Node} \sqcap \exists \mathsf{edge}.\mathsf{Inode}$.

Now consider the following interpretation of the primitive concepts and roles:

$$\begin{aligned}\Delta^{\mathcal{J}} &:= \{m_0, m_1, m_2, \ldots\} \cup \{n_0\},\\ \mathsf{Node}^{\mathcal{J}} &:= \Delta^{\mathcal{J}},\\ \mathsf{edge}^{\mathcal{J}} &:= \{(m_i, m_{i+1}) \mid i \geq 0\} \cup \{(n_0, n_0)\}.\end{aligned}$$

Where $M := \{m_0, m_1, m_2, \ldots\}$ and $N := \{n_0\}$, there are four possible ways of extending this interpretation of the primitive concepts and roles to a model of the TBox consisting of the above concept definition: Inode can be interpreted by $M \cup N$, M, N, or $\emptyset$. All these models are admissible w.r.t. descriptive semantics, whereas the first is the gfp-model and the last is the lfp-model of the TBox. Obviously, only the gfp-model captures the intuition underlying the definition (namely, nodes lying on an infinite path) correctly.

It should be noted, however, that in other cases descriptive semantics appears to be more appropriate. For example, consider the definitions

$$\begin{aligned}\mathsf{Tiger} &\equiv \mathsf{Animal} \sqcap \exists \mathsf{parent}.\mathsf{Tiger} \quad \text{and}\\ \mathsf{Lion} &\equiv \mathsf{Animal} \sqcap \exists \mathsf{parent}.\mathsf{Lion}.\end{aligned}$$

With respect to gfp-semantics, the defined concepts Tiger and Lion must always be interpreted as the same set whereas this is not the case for descriptive semantics.[1]

Before we can define lfp- and gfp-semantics formally, we must introduce some notation. Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox containing the roles N_{role}, the primitive concepts N_{prim}, and the defined concepts $N_{def} := \{A_1, \ldots, A_k\}$. A *primitive interpretations* $\mathcal{J}$ for $\mathcal{T}$ is given by a domain $\Delta^{\mathcal{J}}$, an interpretation of the roles $r \in N_{role}$ by binary relations $r^{\mathcal{J}}$ on $\Delta^{\mathcal{J}}$, and an interpretation of the primitive concepts $P \in N_{prim}$ by subsets $P^{\mathcal{J}}$ of $\Delta^{\mathcal{J}}$. Obviously, a primitive interpretation differs from an interpretation in that it does not interpret the defined concepts in N_{def}. We say that the interpretation $\mathcal{I}$ is *based on* the primitive interpretation $\mathcal{J}$ iff it has the same domain as $\mathcal{J}$ and coincides with $\mathcal{J}$ on N_{role} and N_{prim}. For a fixed primitive interpretation $\mathcal{J}$, the interpretations $\mathcal{I}$ based on it are uniquely determined by the tuple $(A_1^{\mathcal{I}}, \ldots, A_k^{\mathcal{I}})$ of the interpretations of the defined concepts in N_{def}. We define

$$Int(\mathcal{J}) := \{\mathcal{I} \mid \mathcal{I} \text{ is an interpretation based on } \mathcal{J}\}.$$

Interpretations based on $\mathcal{J}$ can be compared by the following ordering, which realizes a pairwise inclusion test between the

[1]This example is similar to the "humans and horses" example used by Nebel [1991] to illustrate the difference between descriptive semantics and gfp-semantics in $\mathcal{ALN}$.

respective interpretations of the defined concepts: if $\mathcal{I}_1, \mathcal{I}_2 \in Int(\mathcal{J})$, then

$$\mathcal{I}_1 \preceq_{\mathcal{J}} \mathcal{I}_2 \text{ iff } A_i^{\mathcal{I}_1} \subseteq A_i^{\mathcal{I}_2} \text{ for all } i, 1 \leq i \leq k.$$

It is easy to see that $\preceq_{\mathcal{J}}$ induces a *complete lattice* on $Int(\mathcal{J})$, i.e., every subset of $Int(\mathcal{J})$ has a least upper bound (lub) and a greatest lower bound (glb). Using *Tarski's fixpoint theorem* [Tarski, 1955] for complete lattices, it is not hard to show [Nebel, 1991] that, for a given primitive interpretation $\mathcal{J}$, there is always a greatest and a least (w.r.t. $\preceq_{\mathcal{J}}$) model of $\mathcal{T}$ based on $\mathcal{J}$. We call these models respectively the greatest fixpoint model (gfp-model) and the least fixpoint model (lfp-model) of $\mathcal{T}$. *Greatest (least) fixpoint semantics* considers only gfp-models (lfp-models) as admissible models.

Definition 2 Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and A, B be defined concepts[2] occurring in $\mathcal{T}$. Then,

- A is subsumed by B w.r.t. descriptive semantics ($A \sqsubseteq_{\mathcal{T}} B$) iff $A^{\mathcal{I}} \subseteq B^{\mathcal{I}}$ holds for all models $\mathcal{I}$ of $\mathcal{T}$.
- A is subsumed by B w.r.t. gfp-semantics ($A \sqsubseteq_{gfp,\mathcal{T}} B$) iff $A^{\mathcal{I}} \subseteq B^{\mathcal{I}}$ holds for all gfp-models $\mathcal{I}$ of $\mathcal{T}$.
- A is subsumed by B w.r.t. lfp-semantics ($A \sqsubseteq_{lfp,\mathcal{T}} B$) iff $A^{\mathcal{I}} \subseteq B^{\mathcal{I}}$ holds for all lfp-models $\mathcal{I}$ of $\mathcal{T}$.

We will show in the following that all three subsumption problems are decidable in polynomial time. To do this, we represent $\mathcal{EL}$-TBoxes as graphs.

3 Description graphs and simulations

$\mathcal{EL}$-TBoxes as well as primitive interpretations can be represented as description graphs. Before we can translate $\mathcal{EL}$-TBoxes into description graphs, we must normalize them. In the following, let $\mathcal{T}$ be an $\mathcal{EL}$-TBox, N_{def} the defined concepts of $\mathcal{T}$, N_{prim} the primitive concepts of $\mathcal{T}$, and N_{role} the roles of $\mathcal{T}$.

We say that the $\mathcal{EL}$-TBox $\mathcal{T}$ is *normalized* iff $A \equiv D \in \mathcal{T}$ implies that D is of the form

$$P_1 \sqcap \ldots \sqcap P_m \sqcap \exists r_1.B_1 \sqcap \ldots \sqcap \exists r_\ell.B_\ell,$$

for $m, \ell \geq 0$, $P_1, \ldots, P_m \in N_{prim}$, $r_1, \ldots, r_\ell \in N_{role}$, and $B_1, \ldots, B_\ell \in N_{def}$. If $m = \ell = 0$, then $D = \top$.

Since there is a polynomial translation of general TBoxes into normalized ones [Baader, 2002], one can restrict the attention to normalized TBoxes. Thus, we will assume that all TBoxes are normalized. Normalized $\mathcal{EL}$-TBoxes can be viewed as graphs whose nodes are the defined concepts, which are labeled by sets of primitive concepts, and whose edges are given by the existential restrictions. For the rest of this section, we fix a normalized $\mathcal{EL}$-TBox $\mathcal{T}$ with primitive concepts N_{prim}, defined concepts N_{def}, and roles N_{role}.

Definition 3 An $\mathcal{EL}$*-description graph* is a graph $\mathcal{G} = (V, E, L)$ where

- V is a set of nodes;
- $E \subseteq V \times N_{role} \times V$ is a set of edges labeled by role names;
- $L\colon V \to 2^{N_{prim}}$ is a function that labels nodes with sets of primitive concepts.

[2]Obviously, we can restrict the attention to subsumption between defined concepts since subsumption between arbitrary concept descriptions can be reduced to this problem by introducing definitions for the descriptions.

The normalized TBox $\mathcal{T}$ can be translated into the following $\mathcal{EL}$-description graph $\mathcal{G}_{\mathcal{T}} = (N_{def}, E_{\mathcal{T}}, L_{\mathcal{T}})$:

- the nodes of $\mathcal{G}_{\mathcal{T}}$ are the defined concepts of $\mathcal{T}$;
- if A is a defined concept and
$$A \equiv P_1 \sqcap \ldots \sqcap P_m \sqcap \exists r_1.B_1 \sqcap \ldots \sqcap \exists r_\ell.B_\ell$$
its definition in $\mathcal{T}$, then
  - $L_{\mathcal{T}}(A) = \{P_1, \ldots, P_m\}$, and
  - A is the source of the edges $(A, r_1, B_1), \ldots, (A, r_\ell, B_\ell) \in E_{\mathcal{T}}$.

Any primitive interpretation $\mathcal{J} = (\Delta^{\mathcal{J}}, \cdot^{\mathcal{J}})$ can be translated into the following $\mathcal{EL}$-description graph $\mathcal{G}_{\mathcal{J}} = (\Delta^{\mathcal{J}}, E_{\mathcal{J}}, L_{\mathcal{J}})$:

- the nodes of $\mathcal{G}_{\mathcal{J}}$ are the elements of $\Delta^{\mathcal{J}}$;
- $E_{\mathcal{J}} := \{(x, r, y) \mid (x, y) \in r^{\mathcal{J}}\}$;
- $L_{\mathcal{J}}(x) := \{P \in N_{prim} \mid x \in P^{\mathcal{J}}\}$ for all $x \in \Delta^{\mathcal{J}}$.

The translation between $\mathcal{EL}$-TBoxes (primitive interpretations) and $\mathcal{EL}$-description graphs works in both directions, i.e., any $\mathcal{EL}$-description graph can also be view as an $\mathcal{EL}$-TBox (primitive interpretation).

Simulations are binary relations between nodes of two $\mathcal{EL}$-description graphs that respect labels and edges in the sense defined below.

Definition 4 Let $\mathcal{G}_i = (V_i, E_i, L_i)$ $(i = 1, 2)$ be two $\mathcal{EL}$-description graphs. The binary relation $Z \subseteq V_1 \times V_2$ is a *simulation* from $\mathcal{G}_1$ to $\mathcal{G}_2$ iff

(S1) $(v_1, v_2) \in Z$ implies $L_1(v_1) \subseteq L_2(v_2)$; and

(S2) if $(v_1, v_2) \in Z$ and $(v_1, r, v_1') \in E_1$, then there exists a node $v_2' \in V_2$ such that $(v_1', v_2') \in Z$ and $(v_2, r, v_2') \in E_2$.

We write $Z\colon \mathcal{G}_1 \overset{\rightharpoonup}{\sim} \mathcal{G}_2$ to express that Z is a simulation from $\mathcal{G}_1$ to $\mathcal{G}_2$.

It is easy to see that the set of all simulations from $\mathcal{G}_1$ to $\mathcal{G}_2$ is closed under arbitrary unions. Consequently, there always exists a greatest simulation from $\mathcal{G}_1$ to $\mathcal{G}_2$. If $\mathcal{G}_1, \mathcal{G}_2$ are finite, then this greatest simulation can be computed in polynomial time [Henzinger *et al.*, 1995]. The following proposition is an easy consequence of this fact (see [Baader, 2002]).

Proposition 5 *Let $\mathcal{G}_1, \mathcal{G}_2$ be two finite $\mathcal{EL}$-description graphs, v_1 a node of $\mathcal{G}_1$ and v_2 a node of $\mathcal{G}_2$. Then we can decide in polynomial time whether there is a simulation $Z\colon \mathcal{G}_1 \overset{\rightharpoonup}{\sim} \mathcal{G}_2$ such that $(v_1, v_2) \in Z$.*

4 Subsumption w.r.t. gfp-semantics

In the following, let $\mathcal{T}$ be a normalized $\mathcal{EL}$-TBox with primitive concepts N_{prim}, defined concepts N_{def}, and roles N_{role}. Before characterizing subsumption w.r.t. gfp-semantics, we give a characterization of when an individual of a gfp-model belongs to a defined concept in this model.

Proposition 6 *Let $\mathcal{J}$ be a primitive interpretation and $\mathcal{I}$ the gfp-model of $\mathcal{T}$ based on $\mathcal{J}$. Then the following are equivalent for any $A \in N_{def}$ and $x \in \Delta^{\mathcal{J}}$:*

1. *$x \in A^{\mathcal{I}}$.*
2. *There is a simulation $Z\colon \mathcal{G}_{\mathcal{T}} \overset{\rightharpoonup}{\sim} \mathcal{G}_{\mathcal{J}}$ such that $(A, x) \in Z$.*

This proposition (whose proof can be found in [Baader, 2002]), can now be used to prove the following characterization of subsumption w.r.t. gfp-semantics in $\mathcal{EL}$.

Theorem 7 *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and A, B defined concepts in $\mathcal{T}$. Then the following are equivalent:*

1. *$A \sqsubseteq_{gfp,\mathcal{T}} B$.*
2. *There is a simulation $Z\colon \mathcal{G}_{\mathcal{T}} \overset{\rightharpoonup}{\sim} \mathcal{G}_{\mathcal{T}}$ with $(B, A) \in Z$.*

Proof. ($2 \Rightarrow 1$) Assume that the simulation $Z\colon \mathcal{G}_{\mathcal{T}} \overset{\rightharpoonup}{\sim} \mathcal{G}_{\mathcal{T}}$ satisfies $(B, A) \in Z$. Let $\mathcal{J}$ be a primitive interpretation and $\mathcal{I}$ the gfp-model of $\mathcal{T}$ based on $\mathcal{J}$. We must show that $x \in A^{\mathcal{I}}$ implies $x \in B^{\mathcal{I}}$.

By Proposition 6, $x \in A^{\mathcal{I}}$ implies that there is a simulation $Y\colon \mathcal{G}_{\mathcal{T}} \overset{\rightharpoonup}{\sim} \mathcal{G}_{\mathcal{J}}$ such that $(A, x) \in Y$. It is easy to show that the composition $X := Z \circ Y$ is a simulation from $\mathcal{G}_{\mathcal{T}}$ to $\mathcal{G}_{\mathcal{J}}$ such that $(B, x) \in X$. By Proposition 6, this implies $x \in B^{\mathcal{I}}$.

($1 \Rightarrow 2$) Assume that $A \sqsubseteq_{gfp,\mathcal{T}} B$. We consider the graph $\mathcal{G}_{\mathcal{T}}$, and view it as an $\mathcal{EL}$-description graph of a primitive interpretation. Thus, let $\mathcal{J}$ be the primitive interpretation with $\mathcal{G}_{\mathcal{T}} = \mathcal{G}_{\mathcal{J}}$, and let $\mathcal{I}$ be the gfp-model of $\mathcal{T}$ based on $\mathcal{J}$.

Since the identity is a simulation $Id\colon \mathcal{G}_{\mathcal{T}} \overset{\rightharpoonup}{\sim} \mathcal{G}_{\mathcal{T}} = \mathcal{G}_{\mathcal{J}}$ that satisfies $(A, A) \in Id$, Proposition 6 yields $A \in A^{\mathcal{I}}$. But then $A \sqsubseteq_{gfp,\mathcal{T}} B$ implies $A \in B^{\mathcal{I}}$, and thus Proposition 6 yields the existence of a simulation $Z\colon \mathcal{G}_{\mathcal{T}} \overset{\rightharpoonup}{\sim} \mathcal{G}_{\mathcal{J}} = \mathcal{G}_{\mathcal{T}}$ such that $(B, A) \in Z$. □

The theorem together with Proposition 5 shows that subsumption w.r.t. gfp-semantics in $\mathcal{EL}$ is tractable.

Corollary 8 *Subsumption w.r.t. gfp-semantics in $\mathcal{EL}$ can be decided in polynomial time.*

Example 9 Consider the TBox $\mathcal{T}$ consisting of the following concept definitions:

$$B \equiv \exists r.C,\ C \equiv \exists r.D,\ D \equiv \exists r.C,\ A \equiv \exists r.A',\ A' \equiv \exists r.D.$$

The $\mathcal{EL}$-description graph $\mathcal{G}_{\mathcal{T}}$ corresponding to this TBox can be found in Figure 1. Let $V_{\mathcal{T}} = \{A, A', B, C, D\}$ denote the set of nodes of this graph. Then $Z := V_{\mathcal{T}} \times V_{\mathcal{T}}$ is a simulation from $\mathcal{G}_{\mathcal{T}}$ to $\mathcal{G}_{\mathcal{T}}$. Consequently, all the defined concepts in $\mathcal{T}$ subsume each other w.r.t. gfp-semantics.

5 Subsumption w.r.t. lfp-semantics

For the sake of completeness, we also treat lfp-semantics. It should be noted, however, that the results of this section demonstrate that lfp-semantics is not interesting in $\mathcal{EL}$.

Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and $\mathcal{G}_{\mathcal{T}}$ the corresponding $\mathcal{EL}$-description graph. Where A, B are nodes of $\mathcal{G}_{\mathcal{T}}$, we write $A \overset{*}{\rightarrow}_{\mathcal{T}} B$ to denote that there is a path in $\mathcal{G}_{\mathcal{T}}$ from A to B, and $A \overset{+}{\rightarrow}_{\mathcal{T}} B$ to denote that there is a non-empty path in $\mathcal{G}_{\mathcal{T}}$ from A to B. We define the set $Cyc_{\mathcal{T}}$ as

$$\{A \mid \text{there exists a node } B \text{ such that } A \overset{*}{\rightarrow}_{\mathcal{T}} B \overset{+}{\rightarrow}_{\mathcal{T}} B\},$$

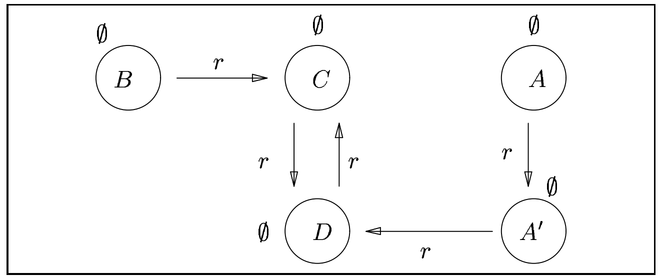

Figure 1: The $\mathcal{EL}$-description graph $\mathcal{G}_{\mathcal{T}}$ of Example 9.

i.e., $Cyc_{\mathcal{T}}$ consists of the nodes in $\mathcal{G}_{\mathcal{T}}$ that can reach a cyclic path in $\mathcal{G}_{\mathcal{T}}$. The following proposition is an easy consequence of the definition of lfp-semantics and of $Cyc_{\mathcal{T}}$ (see [Baader, 2002]).

Proposition 10 *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and A a defined concept in $\mathcal{T}$. If $A \in Cyc_{\mathcal{T}}$, then A is unsatisfiable w.r.t. lfp-semantics, i.e., $A^{\mathcal{I}} = \emptyset$ holds for all lfp-models $\mathcal{I}$ of $\mathcal{T}$.*

In Example 9, all the defined concepts belong to $Cyc_{\mathcal{T}}$, and thus they are all unsatisfiable w.r.t. lfp-semantics.

Since all the defined concepts in $Cyc_{\mathcal{T}}$ are unsatisfiable, their definitions can be removed from the TBox without changing the meaning of the concepts not belonging to $Cyc_{\mathcal{T}}$. (Their definition cannot refer to an element of $Cyc_{\mathcal{T}}$.) This leaves us with an acyclic terminology, on which gfp- and lfp-semantics coincide [Nebel, 1991]. Thus, subsumption w.r.t. lfp-semantics in $\mathcal{EL}$ can be reduced to subsumption w.r.t. gfp-semantics.

Corollary 11 *Subsumption w.r.t. lfp-semantics in $\mathcal{EL}$ can be decided in polynomial time.*

6 Subsumption w.r.t. descriptive semantics

Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox and $\mathcal{G}_{\mathcal{T}}$ the corresponding $\mathcal{EL}$-description graph. Since every gfp-model of $\mathcal{T}$ is a model of $\mathcal{T}$, $A \sqsubseteq_{\mathcal{T}} B$ implies $A \sqsubseteq_{gfp,\mathcal{T}} B$. Consequently, $A \sqsubseteq_{\mathcal{T}} B$ implies that there is a simulation $Z\colon \mathcal{G}_{\mathcal{T}} \overset{\rightharpoonup}{\sim} \mathcal{G}_{\mathcal{T}}$ with $(B, A) \in Z$. In the following we will show what additional properties the simulation Z must satisfy for the implication in the other direction to hold.

To get an intuition on the difference between gfp- and descriptive semantics, let us consider Example 9. With respect to gfp-semantics, all the defined concepts of $\mathcal{T}$ are equivalent (i.e., subsume each other). With respect to descriptive semantics, A, B, D are still equivalent, C is equivalent to A', but A' is not equivalent to B, and C and D are also not equivalent (in both cases, the concepts are not even comparable w.r.t. subsumption).

To see that C and A' are equivalent w.r.t. descriptive semantics, it is enough to note that the following identities hold in every model $\mathcal{I}$ of $\mathcal{T}$: $A'^{\mathcal{I}} = (\exists r.D)^{\mathcal{I}} = C^{\mathcal{I}}$. A similar argument shows that B and D are equivalent. In addition, equivalence of C and A' also implies equivalence of A and B. The following model of $\mathcal{T}$ is a counterexample to the other subsumption relationships:

1. $\Delta^{\mathcal{I}} := \{c, d\}$;

$$
\begin{array}{ccccccccccc}
B & = & B_0 & \stackrel{r_1}{\to} & B_1 & \stackrel{r_2}{\to} & B_2 & \stackrel{r_3}{\to} & B_3 & \stackrel{r_4}{\to} & \cdots \\
 & & Z\downarrow & & Z\downarrow & & Z\downarrow & & Z\downarrow & & \\
A & = & A_0 & \stackrel{r_1}{\to} & A_1 & \stackrel{r_2}{\to} & A_2 & \stackrel{r_3}{\to} & A_3 & \stackrel{r_4}{\to} & \cdots
\end{array}
$$

Figure 2: A (B, A)-simulation chain.

2. $r^{\mathcal{I}} := \{(c,d),(d,c)\}$;
3. $A^{\mathcal{I}} := B^{\mathcal{I}} := D^{\mathcal{I}} := \{d\}$, $A'^{\mathcal{I}} := C^{\mathcal{I}} := \{c\}$.

We will see below that the reason for A' and B not being equivalent is that in the infinite path in $\mathcal{G}_\mathcal{T}$ starting with A', one reaches D with an odd number of edges, whereas C is reached with an even number; for the path starting with B, it is just the opposite. In contrast, the infinite paths starting respectively with A and B "synchronize" after a finite number of steps. To formalize this intuition, we must introduce some notation.

Definition 12 The path p_1: $B = B_0 \stackrel{r_1}{\to} B_1 \stackrel{r_2}{\to} B_2 \stackrel{r_3}{\to} B_3 \stackrel{r_4}{\to} \cdots$ in $\mathcal{G}_\mathcal{T}$ is Z*-simulated* by the path p_2: $A = A_0 \stackrel{r_1}{\to} A_1 \stackrel{r_2}{\to} A_2 \stackrel{r_3}{\to} A_3 \stackrel{r_4}{\to} \cdots$ in $\mathcal{G}_\mathcal{T}$ iff $(B_i, A_i) \in Z$ for all $i \geq 0$. In this case we say that the pair (p_1, p_2) is a (B, A)*-simulation chain w.r.t.* Z (see Figure 2).

Consider the TBox $\mathcal{T}$ and the simulation Z introduced in Example 9. Then

$$
\begin{array}{ccccccccccc}
B & \stackrel{r}{\to} & C & \stackrel{r}{\to} & D & \stackrel{r}{\to} & C & \stackrel{r}{\to} & D & \stackrel{r}{\to} & \cdots \\
Z\downarrow & & Z\downarrow & & Z\downarrow & & Z\downarrow & & Z\downarrow & & \\
A & \stackrel{r}{\to} & A' & \stackrel{r}{\to} & D & \stackrel{r}{\to} & C & \stackrel{r}{\to} & D & \stackrel{r}{\to} & \cdots
\end{array}
$$

is a (B, A)-simulation chain w.r.t. Z, and

$$
\begin{array}{ccccccccccc}
B & \stackrel{r}{\to} & C & \stackrel{r}{\to} & D & \stackrel{r}{\to} & C & \stackrel{r}{\to} & D & \stackrel{r}{\to} & \cdots \\
Z\downarrow & & Z\downarrow & & Z\downarrow & & Z\downarrow & & Z\downarrow & & \\
A' & \stackrel{r}{\to} & D & \stackrel{r}{\to} & C & \stackrel{r}{\to} & D & \stackrel{r}{\to} & C & \stackrel{r}{\to} & \cdots
\end{array}
$$

is a (B, A')-simulation chain w.r.t. Z. Note that the first chain synchronizes after a finite number of steps in the sense that there is a Z-link between the same defined concept. In contrast, the second chain does not synchronize in this sense.

If $(B, A) \in Z$, then (S2) of Definition 4 implies that, for every infinite path p_1 starting with $B_0 := B$, there is an infinite path p_2 starting with $A_0 := A$ such that p_1 is Z-simulated by p_2. In the following we construct such a simulating path step by step. The main point is, however, that the decision which concept A_n to take in step n should depend only on the partial (B, A)-simulation chain already constructed, and *not* on the parts of the path p_1 not yet considered.

Definition 13 A *partial* (B, A)*-simulation chain* is of the form depicted in Figure 3. A *selection function* S for A, B and Z assigns to each partial (B, A)-simulation chain of this form a defined concept A_n such that (A_{n-1}, r_n, A_n) is an edge in $\mathcal{G}_\mathcal{T}$ and $(B_n, A_n) \in Z$.

Given a path $B = B_0 \stackrel{r_1}{\to} B_1 \stackrel{r_2}{\to} B_2 \stackrel{r_3}{\to} B_3 \stackrel{r_4}{\to} \cdots$ and a defined concept A such that $(B, A) \in Z$, one can use a selection function S for A, B and Z to construct a Z-simulating path. In this case we say that the resulting (B, A)-simulation chain is S*-selected*.

$$
\begin{array}{ccccccccccc}
B & = & B_0 & \stackrel{r_1}{\to} & B_1 & \stackrel{r_2}{\to} & \cdots & \stackrel{r_{n-1}}{\to} & B_{n-1} & \stackrel{r_n}{\to} & B_n \\
 & & Z\downarrow & & Z\downarrow & & & & Z\downarrow & & \\
A & = & A_0 & \stackrel{r_1}{\to} & A_1 & \stackrel{r_2}{\to} & \cdots & \stackrel{r_{n-1}}{\to} & A_{n-1} & &
\end{array}
$$

Figure 3: A partial (B, A)-simulation chain.

Definition 14 The simulation Z: $\mathcal{G}_\mathcal{T} \stackrel{\rightharpoonup}{\sim} \mathcal{G}_\mathcal{T}$ is called (B, A)*-synchronized* iff there exists a selection function S for A, B and Z such that the following holds: for every infinite S-selected (B, A)-simulation chain of the form depicted in Figure 2 there exists an $i \geq 0$ such that $A_i = B_i$.

We are now ready to state our characterization of subsumption w.r.t. descriptive semantics (see [Baader, 2002] for the proof).

Theorem 15 *Let $\mathcal{T}$ be an $\mathcal{EL}$-TBox, and A, B defined concepts in $\mathcal{T}$. Then the following are equivalent:*

1. *$A \sqsubseteq_\mathcal{T} B$.*
2. *There is a (B, A)-synchronized simulation Z: $\mathcal{G}_\mathcal{T} \stackrel{\rightharpoonup}{\sim} \mathcal{G}_\mathcal{T}$ such that $(B, A) \in Z$.*

It remains to be shown that property (2) of the theorem can be decided in polynomial time. To this purpose, we construct a simulation Y such that (2) of Theorem 15 is equivalent to $(B, A) \in Y$ (see [Baader, 2002] for the proof that this is indeed the case):
We define $Y := \bigcup_{n \geq 0} Y_n$, where the relations Y_n are defined by induction on n: Y_0 is the identity on the nodes of $\mathcal{G}_\mathcal{T}$. If Y_n is already defined, then

$$Y_{n+1} := Y_n \cup \{(C, C') \mid (C, C') \text{ satisfies (1), (2), (3) below}\}.$$

1. $L_\mathcal{T}(C) \subseteq L_\mathcal{T}(C')$,
2. $(C, r_1, C_1), \ldots, (C, r_\ell, C_\ell)$ are all the edges in $\mathcal{G}_\mathcal{T}$ with source C,
3. there are edges $(C', r_1, C'_1), \ldots, (C', r_\ell, C'_\ell)$ in $\mathcal{G}_\mathcal{T}$ such that $(C_1, C'_1) \in Y_n, \ldots, (C_\ell, C'_\ell) \in Y_n$.

The relation Y can obviously be computed in time polynomial in the size of $\mathcal{G}_\mathcal{T}$. By using the techniques employed to decided Horn-SAT in linear time [Dowling and Gallier, 1984], it is not hard to show that Y can actually be computed in time quadratic in the size of $\mathcal{G}_\mathcal{T}$.

Corollary 16 *Subsumption w.r.t. descriptive semantics in $\mathcal{EL}$ can be decided in polynomial time.*

An alternative way for showing the polynomiality result would be to reduce the existence of a (B, A)-synchronized simulation Z satisfying $(B, A) \in Z$ to the strategy problem for a certain two-player game with a positional winning condition. The existence of a winning strategy is in this case a polynomial time problem [Grädel, 2002].

7 Future and related work

We have seen that subsumption in $\mathcal{EL}$ with cyclic terminologies is polynomial for the three types of semantics introduced by Nebel [1991]. In some applications, it would be interesting to have a mix of all three semantics, and it remains to be seen whether the polynomiality results also hold in such a setting (which would correspond to a restriction of the modal μ-calculus [Kozen, 1983]).

Sub-Boolean DLs (like $\mathcal{EL}$) have attracted renewed attention in the context of so-called non-standard inferences [Küsters, 2001] like computing the least common subsumer and the most specific concept. In [Baader, 2003] we have shown that the characterization of subsumption in $\mathcal{EL}$ w.r.t. gfp-semantics also yields an approach for computing the least common subsumer in $\mathcal{EL}$ w.r.t. gfp-semantics. In addition, we have extended the characterization of subsumption in $\mathcal{EL}$ w.r.t. gfp-semantics to the instance problem, and have shown how this can be used to compute the most specific concept.

Simulations and bisimulations play an important rôle in modal logics (and thus also in description logics). However, until now they have mostly been considered for modal logics that are closed under all the Boolean operators, and they have usually not been employed for reasoning in the logic. A notable exception are [Kurtonina and de Rijke, 1997; 1999], where bisimulation characterizations are given for sub-Boolean Modal Logics and DLs. However, these characterizations are used to give a formal account of the expressive power of these logics. They are not employed for reasoning purposes.

In [Baader *et al.*, 1999], subsumption between $\mathcal{EL}$-concept descriptions was characterized through the existence of homomorphisms between the description trees (basically the syntax trees) associated with the descriptions. If one looks at the polynomial time algorithm for deciding the existence of such a homomorphism, then it is easy to see that it actually computes the greatest simulation between the trees. For trees, the existence of a homomorphism mapping the root to the root coincides with the existence of a simulation containing the tuple of the roots. For graphs, a similar connection does not hold. In fact, for graphs the existence of a homomorphism is an NP-complete problem. For simple conceptual graphs (or equivalently, conjunctive queries) the implication (containment) problem can be characterized via the existence of certain homomorphisms between graphs [Chein *et al.*, 1998], and is thus NP-complete.

References

[Baader, 2003] F. Baader. Least common subsumers and most specific concepts in a description logic with existential restrictions and terminological cycles. In *Proc. IJCAI'03*, 2003.

[Baader, 2002] F. Baader. Terminological cycles in a description logic with existential restrictions. LTCS-Report 02-02, TU Dresden, 2002. See http://lat.inf.tu-dresden.de/research/reports.html.

[Baader *et al.*, 1999] F. Baader, R. Küsters, and R. Molitor. Computing least common subsumers in description logics with existential restrictions. In *Proc. IJCAI'99*, 1999.

[Baader, 1990] F. Baader. Terminological cycles in KL-ONE-based knowledge representation languages. In *Proc. AAAI'90*, 1990.

[Baader, 1996] F. Baader. Using automata theory for characterizing the semantics of terminological cycles. *Ann. of Mathematics and Artificial Intelligence*, 18, 1996.

[Chein *et al.*, 1998] M. Chein, M.-L. Mugnier, and G. Simonet. Nested graphs: A graph-based knowledge representation model with FOL semantics. In *Proc. KR'98*, 1998.

[Cote *et al.*, 1993] 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.

[De Giacomo and Lenzerini, 1994] G. De Giacomo and M. Lenzerini. Concept language with number restrictions and fixpoints, and its relationship with μ-calculus. In *Proc. ECAI'94*, 1994.

[Dowling and Gallier, 1984] W. F. Dowling and J. Gallier. Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *J. Logic Programming*, 1(3), 1984.

[Grädel, 2002] E. Grädel. Model checking games. In *Proc. WOLLIC 02*, volume 67 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2002.

[Henzinger *et al.*, 1995] M. R. Henzinger, T. A. Henzinger, and P. W. Kopke. Computing simulations on finite and infinite graphs. In *36th Annual Symposium on Foundations of Computer Science*, 1995. IEEE Computer Society Press.

[Kozen, 1983] D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27, 1983.

[Kurtonina and de Rijke, 1997] N. Kurtonina and M. de Rijke. Simulating without negation. *J. Logic and Computation*, 7, 1997.

[Kurtonina and de Rijke, 1999] N. Kurtonina and M. de Rijke. Expressiveness of concept expressions in first-order description logics. *Artificial Intelligence*, 107(2), 1999.

[Küsters, 1998] R. Küsters. Characterizing the semantics of terminological cycles in $\mathcal{ALN}$ using finite automata. In *Proc. KR'98*, 1998.

[Küsters, 2001] R. Küsters. *Non-standard Inferences in Description Logics*, Springer LNAI 2100, 2001.

[Nebel, 1990] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems*, Springer LNAI 422, 1990.

[Nebel, 1991] B. Nebel. Terminological cycles: Semantics and computational properties. In J. F. Sowa, editor, *Principles of Semantic Networks*. Morgan Kaufmann, Los Altos, 1991.

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

[Schild, 1994] K. Schild. Terminological cycles and the propositional μ-calculus. In *Proc. KR'94*, 1994.

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

[Spackman *et al.*, 1997] K. Spackman, K. Campbell, and R. Cote. SNOMED RT: A reference terminology for health care. *J. of the American Medical Informatics Association*, 1997. Fall Symposium Supplement.

[Spackman, 2001] K. Spackman. Normal forms for description logic expressions of clinical concepts in SNOMED RT. *J. of the American Medical Informatics Association*, 2001. Symposium Supplement.

[Tarski, 1955] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific J. Mathematics*, 5:285–309, 1955.