Master's thesis

on topic

Reasoning in the Description Logic $\mathcal{EL}$ Extended with an n-ary Existential Quantifier

by

Eldar Karabaev
Born on 15th June 1979 in Moscow

Technische Universität Dresden
Department of Computer Science

submitted on 23rd August 2005

Overseeing Professor: Prof. Franz Baader
Supervisor: Dr. Carsten Lutz

Abstract

Motivated by a chemical process engineering application, we introduce a new concept constructor, namely an n-ary variant of the existential restriction, into the Description Logic (DL) $\mathcal{EL}$. We refer to the resulting logic as $\mathcal{EL}^{(n)}$ and to its fragment that matches the needs of the real world application as restricted $\mathcal{EL}^{(n)}$.

Although the new constructor can be expressed in the DL $\mathcal{ALCQ}$, its translation is exponential and introduces many expensive constructors, thus making the translation-based reasoning impractical. In the present work, we design direct algorithms for deciding the main inference problem, namely subsumption, in restricted $\mathcal{EL}^{(n)}$. We show that reasoning in restricted $\mathcal{EL}^{(n)}$ is polynomial when we allow for acyclic TBoxes. Additionally, we examine the complexity of reasoning in (unrestricted) $\mathcal{EL}^{(n)}$ with general TBoxes. In particular, we show that subsumption in $\mathcal{EL}^{(n)}$ with GCIs is ExpTime-complete.

In order to test the practical efficiency of our approach, we implement the polynomial algorithm for restricted $\mathcal{EL}^{(n)}$ with acyclic TBoxes in a system called Eln. Comparison between Eln and the state-of-the-art DL reasoner Racer demonstrates a considerable advantage of the direct algorithm over the translation-based approach.

Contents

Acknowledgments

First of all, I am deeply grateful to my advisor Carsten Lutz, whose thoughtful guidance, insightful vision, and continuing support have lead me to grow immensely over the last year. Carsten's ability to select interesting problems, his high standards, and hard work are contagious. I am thankful for the opportunity to learn from Carsten.

I also want to thank Professor Franz Baader for helpful suggestions and stimulating discussions on this thesis. I am indebted to him for attracting me with such interesting and promising research area as Description Logics and helping to get acquainted with it.

Moreover, I would like to acknowledge the support by a scholarship provided by the International Quality Network on Rational Mobile Agents for joining the Computational Logic (CL) Master's Programme at the Dresden University of Technology. In particular, I would like to thank Professor Steffen Hölldobler for his trust in my ability to successfully complete the studies within the CL programme.

Most importantly, is the endless love, support, and encouragement given to me by my parents and my wife Olga.

Chapter 1

Introduction

Description logics (DLs) are a family of knowledge representation formalisms that are designed to represent and reason about terminological knowledge. Following the ideas of semantic networks [Bra79] and frames [Min81], DLs provide means for representing the knowledge of an application domain in a structured and formally well-understood way. Similarly to their predecessors, DLs express knowledge using *concepts* that capture the important notions of the domain and *roles* that reflect relationships between domain objects. Using a variety of concept constructors, concept names and role names that are atomic concepts and roles, respectively, are combined into *concept terms*, or *concept descriptions*. For example, the following is a concept term in the description logic $\mathcal{ALC}$ [SSS91]:

$$human \sqcap \exists has\_child.human \sqcap \forall has\_child.male,$$

where *human* and *male* are concept names, *has_child* is a role name, and $\exists$, $\forall$ and $\sqcap$ are concept constructors. In contrast to semantic networks and frames, DLs are equipped with a formal, logic-based semantics. Recalling the above example, the meaning of the concept term is uniquely defined based on the interpretation of the concepts *human* and *male* and the role *has_child*. More precisely, this concept term captures exactly those parents that have only sons.

Usually, a DL-based reasoning system consists of a set of axioms, referred to as the TBox, or terminology, and a set of assertions, referred to as the ABox. TBoxes define complex concepts over the simple ones, thereby representing the concept hierarchy. ABoxes are employed to assert relations between domain individuals or to assign domain individuals to specific concepts. The standard inference services in a DL-based system include *subsumption* that verifies, whether one of two given concepts is more general than the other one, and *satisfiability* that checks whether a concept is consistent or not.

The expressiveness of a particular DL is usually characterized by the set of constructors available in its syntax. There exists a huge diversity of DLs varying from the ones of low expressive power, e.g., $\mathcal{EL}$ [Baa03] that contains conjunction ($C \sqcap D$) and existential restrictions ($\exists r.C$) as the only possible constructors, to the very expressive logics, like, e.g., $\mathcal{SHIQ}$ [HST00] that along with standard constructors includes qualified number restrictions, inverse and transitive roles. In [BL84], it was argued that there is a trade-off between the expressiveness of a DL language and the tractability of reasoning in that language. In other words, the more expressive the language, the harder usually the reasoning. For example, subsumption in $\mathcal{EL}$ can be solved in polynomial time, whereas in $\mathcal{SHIQ}$ it becomes ExpTime-complete.

In order to ensure a reasonable behaviour of a DL-based system, inference problems for the DL underlying the system should be at least decidable, and preferably of low complexity. Consequently, the expressive power of the DL must be restricted in an appropriate way. Because of this restriction of the expressive power of DLs, various application-driven language extensions have been recently proposed in the literature, e.g., [BH93, CLN94, Sat96]. Some of these extensions have been integrated into state-of-the-art DL systems [Hor98, HM01].

The present work considers a new concept constructor that is motivated by a process engineering application [TvW04]. This constructor is an n-ary variant of the usual existential restriction operator available in most DLs. In this application, a rather inexpressive DL that provides only the new constructor together with conjunction is sufficient. Therefore, we opt for $\mathcal{EL}$, that allows for conjunction and existential restrictions, as the underlying logic for our further investigations.

1.1 The description logic $\mathcal{EL}$

Having a rather simple syntax, the DL $\mathcal{EL}$ enjoys nice algorithmic properties. It was recently shown that subsumption in $\mathcal{EL}$ stays tractable with respect to both acyclic and cyclic TBoxes [Baa03] and in the presence of general concept inclusion axioms (GCIs) [Bra04]. Moreover, in [BBL05], it was proven that $\mathcal{EL}$ augmented with the bottom-concept $\bot$ (and thus, disjointness statements), nominals and concrete domains stays polynomial in the presence of GCIs and role inclusion axioms (RIs).

Despite the simplicity of the syntax of $\mathcal{EL}$, there are application areas where the expressive power of $\mathcal{EL}$ appears to be sufficient. For example, the Systematized Nomenclature of Medicine, referred to as Snomed, employs $\mathcal{EL}$ with an acyclic TBox [Spa00]. Large parts of the medical knowledge base Galen [RNG93] can also be expressed in $\mathcal{EL}$ with GCIs and transitive roles [RH97].

1.2 The need for a more general existential quantifier

The quest for extending the existing DLs by new constructors is always motivated by a specific application area. The main motivation for this work comes from the area of chemical process engineering, where the use of mathematical modelling becomes more and more popular [Mar94, vW04]. The reason for using modelling is that it allows, e.g., to analyze the behaviour of a chemical plant without building a real one but using a computer instead.

In [TvW04], it was argued that a DL-based reasoning system could be successfully used to determine reusable parts of a chemical process or equipment. The authors have also stated that the underlying DL should be able to express a more general existential quantifier that they refer to as an *n-ary existential quantifier*. We illustrate the need for this new constructor with the following example. Assume that we want to describe a chemical plant that has a reactor with a main reaction, and *in addition* a reactor with a main and a side reaction. Also assume that concepts *Reactor_with_main_reaction* and *Reactor_with_main_and_side_reaction* are defined such that the first concept subsumes the second one. We could try to model this chemical plant using the usual existential restriction as follows:

$$\begin{aligned}&\mathit{Plant} \sqcap \exists \mathit{has\_part}.\mathit{Reactor\_with\_main\_reaction} \sqcap \\ &\qquad \exists \mathit{has\_part}.\mathit{Reactor\_with\_main\_and\_side\_reaction}.\end{aligned}$$

However, because of the subsumption relationship between the two reactor concepts, this concept is equivalent to

$$\mathit{Plant} \sqcap \exists \mathit{has\_part}.\mathit{Reactor\_with\_main\_and\_side\_reaction},$$

and thus it does not capture the intended meaning of a plant having two reactors, one with main reaction and the other with a main and a side reaction. To overcome this problem, we consider a new concept constructor of the form

$$\exists r.(C_1, \ldots, C_n)$$

with the intended meaning that it describes all domain individuals that have n different r-successors $d_1, \ldots, d_n$ such that each d_i belongs to C_i, $i = 1, \ldots, n$. Given this constructor, our concept can correctly be described as

$$\begin{aligned}&\mathit{Plant} \sqcap \exists \mathit{has\_part}.(\mathit{Reactor\_with\_main\_reaction}, \\ &\qquad\qquad \mathit{Reactor\_with\_main\_and\_side\_reaction}).\end{aligned}$$

In [TvW04], it has been demonstrated that this new constructor can be expressed in the DL $\mathcal{ALCQ}$ [HB91]. Thus, the n-ary existential quantifier does not extend the expressive power of existing DLs. In fact, the translation of the new constructor into $\mathcal{ALCQ}$ is exponential and, in addition, introduces many expensive constructors such as disjunction and qualified number restrictions. For this reason, even highly optimized DL systems like Racer [HM01] cannot handle the translated concepts in a satisfactory way. Moreover, in the process engineering application [TvW04], the full expressiveness of $\mathcal{ALCQ}$ is not needed. The DL $\mathcal{EL}$ augmented in a restricted way with the new n-ary existential quantifier appears to be sufficient. The resulting logic is referred to as restricted $\mathcal{EL}^{(n)}$.

In the present work, we start with the DL $\mathcal{EL}$ and investigate the effect on the complexity of the subsumption problem that is caused by the addition of the new constructor. We prove that the subsumption problem in restricted $\mathcal{EL}^{(n)}$ remains tractable in the presence of acyclic TBoxes. In addition, we prove that subsumption in $\mathcal{EL}^{(n)}$ in the presence of GCIs is ExpTime-complete. In order to test the practical efficiency of our approach, we implement the polynomial algorithm for restricted $\mathcal{EL}^{(n)}$ with acyclic TBoxes in a system called Eln. Comparison between Eln and a state-of-the-art DL reasoner Racer shows a considerable advantage of the direct algorithm over the translation-based approach.

The present work is organized as follows. In Chapter 2, we introduce the syntax and semantics of the DL $\mathcal{EL}^{(n)}$ and its restricted fragment and show that subsumption between two restricted $\mathcal{EL}^{(n)}$-concept terms is polynomial. Chapter 3 defines the notion of a restricted $\mathcal{EL}^{(n)}$-TBox and shows how to extend the polynomial algorithm developed in Chapter 2 in order to reason with respect to this kind of TBoxes. In Chapter 4, we consider reasoning in the (unrestricted) logic $\mathcal{EL}^{(n)}$. We show that subsumption in $\mathcal{EL}^{(n)}$ with general TBoxes in ExpTime-complete. In addition, we demonstrate that subsumption in the extension of $\mathcal{EL}^{(n)}$ with the complement operator $\neg$ is also ExpTime-complete in the presence of general TBoxes. In Chapter 5, we present the experimental evaluation of the polynomial algorithm for solving subsumption in restricted $\mathcal{EL}^{(n)}$ w.r.t. acyclic TBoxes that is developed in Chapter 3 and compare it with a highly optimized DL system Racer. Finally, we make conclusions and propose several directions of the future work in Chapter 6.

Chapter 2

The Description Logics $\mathcal{EL}^{(n)}$ and Restricted $\mathcal{EL}^{(n)}$

Our primary goal is to augment the DL $\mathcal{EL}$ with a new constructor referred to as the *n-ary existential quantifier*. In the following, we refer to this new DL as $\mathcal{EL}^{(n)}$. In the next two chapters, we consider a restricted variant of $\mathcal{EL}^{(n)}$ that matches the needs of the process engineering application [TvW04]. We refer to this fragment as *restricted* $\mathcal{EL}^{(n)}$.

Firstly, we introduce the syntax and semantics of the logic $\mathcal{EL}^{(n)}$ and its restricted fragment, and then we consider the complexity of the main inference problem in the restricted $\mathcal{EL}^{(n)}$, namely the subsumption problem.

2.1 Syntax and Semantics

The set of restricted $\mathcal{EL}^{(n)}$-concept terms is inductively defined with the help of the set of constructors, starting with a set N_c of concept names and a set N_r of role names.

Definition 1 (Syntax) *Let N_c and N_r be disjoint sets of concept and role names, respectively. Then the set of $\mathcal{EL}^{(n)}$-*concept terms *is defined as follows:*

- $\top$ *is an $\mathcal{EL}^{(n)}$-concept term;*
- *every $A \in N_c$ is an $\mathcal{EL}^{(n)}$-concept term;*
- *If $C, D, C_1, \ldots, C_n$ are $\mathcal{EL}^{(n)}$-concept terms, for some $n > 0$, and $r \in N_r$, then the following are $\mathcal{EL}^{(n)}$-concept terms: $C \sqcap D$, $\exists r.(C_1, \ldots, C_n)$.*

The application that motivates this work does not require the full expressive power of the logic $\mathcal{EL}^{(n)}$. We define now the fragment of $\mathcal{EL}^{(n)}$ referred to as *restricted* $\mathcal{EL}^{(n)}$ which covers the needs of the process engineering application [TvW04]. Reasoning in the (unrestricted) logic $\mathcal{EL}^{(n)}$ will be considered in more details in Chapter 4.

Definition 2 (Restricted $\mathcal{EL}^{(n)}$) *Let N_r and N_c be disjoint sets of role and concept names, respectively. Then the set of* restricted $\mathcal{EL}^{(n)}$-concept terms *is defined as follows:*

- $\top$ *is a restricted $\mathcal{EL}^{(n)}$-concept term;*
- *every $A \in N_c$ is a restricted $\mathcal{EL}^{(n)}$-concept term;*
- *If $P_1, \dots, P_k \in N_c$, $C_1^1, \dots, C_{n_m}^m$ are restricted $\mathcal{EL}^{(n)}$-concept terms and $r_1, \dots, r_m \in N_r$, where $k, m > 0$, $n_1 \geq 1, \dots, n_m \geq 1$ and $r_i \neq r_j$ for all $i \neq j$, then the following is a restricted $\mathcal{EL}^{(n)}$-concept term:*

$$P_1 \sqcap \ldots \sqcap P_k \sqcap \exists r_1.(C_1^1, \dots, C_{n_1}^1) \sqcap \ldots \sqcap \exists r_m.(C_1^m, \dots, C_{n_m}^m).$$

In the following, we use r, r_i and s to denote role names, A, B to denote concept names, and C, D, C_i and D_i to denote concept terms ($i = 1, 2, \dots$).

One should note that Definition 2 requires that in a restricted $\mathcal{EL}^{(n)}$-concept term $P_1 \sqcap \ldots \sqcap P_k \sqcap \exists r_1.(C_1^1, \dots, C_{n_1}^1) \sqcap \ldots \sqcap \exists r_m.(C_1^m, \dots, C_{n_m}^m)$, all the role names $r_1, \dots, r_m$ are distinct. Thus, every role name can appear not more than once in the same conjunction. This is the reason why we call the logic *restricted*. In the Chapters 2 and 3, we will consider the restricted $\mathcal{EL}^{(n)}$, only, and sometimes we omit explicit usage of the word *restricted* for better readability.

Obviously, any restricted $\mathcal{EL}^{(n)}$-concept term is an $\mathcal{EL}^{(n)}$-concept term. Thus, it is enough to define the semantics for general $\mathcal{EL}^{(n)}$-concept terms.

Definition 3 (Semantics) *The semantics for $\mathcal{EL}^{(n)}$-concept terms is given by means of* interpretations

$$\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}),$$

where $\Delta^{\mathcal{I}}$ is a non-empty set (usually referred to as the domain *of the interpretation $\mathcal{I}$), and $\cdot^{\mathcal{I}}$ is a mapping which maps each concept name A to a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each role name r to a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. For complex concept terms, we define the extension inductively as follows:*

$$\begin{aligned} \top^{\mathcal{I}} &:= \Delta^{\mathcal{I}} \\ (C_1 \sqcap C_2)^{\mathcal{I}} &:= C_1{}^{\mathcal{I}} \cap C_2^{\mathcal{I}} \\ \big(\exists r.(C_1, \dots, C_n)\big)^{\mathcal{I}} &:= \{x \in \Delta^{\mathcal{I}} | \exists y_1 \in C_1^{\mathcal{I}}, \dots, y_n \in C_n^{\mathcal{I}}, (x, y_i) \in r^{\mathcal{I}}, 1 \leq i \leq n, \\ & \qquad y_i \neq y_j, 1 \leq i, j \leq n, i \neq j\} \end{aligned}$$

We say, that a concept term C is subsumed *by a concept term D, written $C \sqsubseteq D$, iff for any interpretation $\mathcal{I}$ the following holds: $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.*

2.2 Graph representation of restricted $\mathcal{EL}^{(n)}$-concept terms

In order to justify that subsumption between two restricted $\mathcal{EL}^{(n)}$-concept terms is decidable in polynomial time, we follow [BKM99], where it was shown that subsumption in $\mathcal{EL}$ corresponds to the existence of a homomorphism between the description trees corresponding to the concept terms. This, in turn, showed that subsumption between $\mathcal{EL}$-concept terms is decidable in polynomial time since the existence of a homomorphism between trees is a polynomial time problem.

For this reason, we represent restricted $\mathcal{EL}^{(n)}$-concept terms as finite trees referred to as $\mathcal{EL}^{(n)}$-description trees and interpretations as graphs referred to as $\mathcal{EL}^{(n)}$-description graphs.

Definition 4 ($\mathcal{EL}^{(n)}$-description graph) *Let N_r and N_c be the sets of role and concept names, respectively. An $\mathcal{EL}^{(n)}$-*description graph *is a triple $\mathcal{G} = (V, E, \ell)$ where*

- *V is a set of nodes,*
- *$E \subseteq V \times N_r \times V$ is a set of labeled edges, and*
- *$\ell : V \to 2^{N_c}$ is a node labeling function.*

*An $\mathcal{EL}^{(n)}$-description graph $\mathcal{G} = (V, E, \ell)$ is called $\mathcal{EL}^{(n)}$-*description tree *if (V, E) is a tree. A subtree of an $\mathcal{EL}^{(n)}$-description tree $\mathcal{T}$ with the root node v, is denoted $\mathcal{T}(v)$.*

Considering an $\mathcal{EL}^{(n)}$-description graph $\mathcal{G} = (V, E, \ell)$, we define, for every role name $r \in N_r$, an additional successor-function $S_r^E : V \longrightarrow 2^V$ as follows: $S_r^E(v) := \big\{w | (v, r, w) \in E\big\}$.

In order to construct an $\mathcal{EL}^{(n)}$-description tree for some restricted $\mathcal{EL}^{(n)}$-concept term C, we use the notion of *role depth* of C denoted as $\mathsf{rdepth}(C)$ which is inductively defined as follows:

- $\mathsf{rdepth}(\top) = \mathsf{rdepth}(P_1 \sqcap \ldots \sqcap P_k) := 0$, for any $P_1, \ldots, P_k \in N_c$, $k > 0$;
- $\mathsf{rdepth}\big(P_1 \sqcap \ldots \sqcap P_k \sqcap \exists r_1.(C_1^1, \ldots, C_{n_1}^1) \sqcap \ldots \sqcap \exists r_m.(C_1^m, \ldots, C_{n_m}^m)\big) := 1 + max\big(\mathsf{rdepth}(C_1^1), \ldots, \mathsf{rdepth}(C_{n_m}^m)\big)$, if $m > 0$.

Each restricted $\mathcal{EL}^{(n)}$-concept term $C = P_1 \sqcap \ldots \sqcap P_k \sqcap \exists r_1.(C_1^1, \ldots, C_{n_1}^1) \sqcap \ldots \sqcap \exists r_m.(C_1^m, \ldots, C_{n_m}^m)$, can be inductively translated into the corresponding $\mathcal{EL}^{(n)}$-description tree $\mathcal{T}_C = (V, E, \ell)$ as follows:

- If $\mathsf{rdepth}(C) = 0$, then $V := \{v_0\}, E := \emptyset$ and $\ell(v_0) := \{P_1, \ldots, P_k\} \backslash \{\top\}$.

- If $\mathsf{rdepth}(C) > 0$, then for $1 \leq i \leq m$, $1 \leq j \leq n_i$ let $\mathcal{T}_j^i = (V_j^i, E_j^i, \ell_j^i)$ be the inductively defined $\mathcal{EL}^{(n)}$-description trees corresponding to C_j^i, where w.l.o.g., all the V_j^i are pairwise disjoint. Let v_j^i denote the root of the tree $\mathcal{T}_j^i$. Then
  - $V := \{v_0\} \cup \bigcup_{1 \leq i \leq m, 1 \leq j \leq n_i} V_j^i$,
  - $E := \{(v_0, r_i, v_j^i) | 1 \leq i \leq m, 1 \leq j \leq n_i\} \cup \bigcup_{1 \leq i \leq m, 1 \leq j \leq n_i} E_j^i$,
  - $\ell(v) := \begin{cases} \{P_1, \ldots, P_k\} \backslash \{\top\}, & v = v_0 \\ \ell_j^i(v), & v \in V_j^i, 1 \leq i \leq m, 1 \leq j \leq n_i. \end{cases}$

One can observe that the size of every component of the tree $\mathcal{T}_C$ is linearly bounded by the size of C. Thus, the size of the tree $\mathcal{T}_C$ is linear in the size of C.

Conversely, any $\mathcal{EL}^{(n)}$-description tree $\mathcal{T}$ can be translated into the corresponding $\mathcal{EL}^{(n)}$-concept term $C_\mathcal{T}$. For any $\mathcal{EL}^{(n)}$-description tree $\mathcal{T} = (V, E, \ell)$ with root node v_0, we denote the length of the maximal path in $\mathcal{T}$ as $\mathsf{depth}(\mathcal{T})$. The translation procedure can now be defined by induction on $\mathsf{depth}(\mathcal{T})$:

- If $\mathsf{depth}(\mathcal{T}) = 0$ we know that $V = \{v_0\}$ and $E = \emptyset$. If $l(v_0) = \emptyset$, then $C_\mathcal{T} := \top$, otherwise, let $\ell(v_0) = \{P_1, \ldots, P_k\}$, $k > 0$, and we define $C_\mathcal{T} := P_1 \sqcap \ldots \sqcap P_k$.

- If $\mathsf{depth}(\mathcal{T}) > 0$, then let $\ell(v_0) = \{P_1, \ldots, P_k\}$, $k \geq 0$. We define
  - $C(v)$ to be the inductively defined $\mathcal{EL}^{(n)}$-concept term corresponding to the subtree $\mathcal{T}(v)$ of $\mathcal{T}$, for each $v \in V$, and
  - For every role name $r \in N_r$, $\mathcal{E}_r := \exists r.(C(v_1), \ldots, C(v_n))$

  Finally, we define $C_\mathcal{T} := P_1 \sqcap \ldots \sqcap P_k \sqcap \bigsqcap_{r \in N_r, S_r^E(v_0) \neq \emptyset} \mathcal{E}_r$.

Similarly, any interpretation $\mathcal{I}$ can be translated into an $\mathcal{EL}^{(n)}$-description graph $\mathcal{G}_\mathcal{I} = (V_\mathcal{I}, E_\mathcal{I}, \ell_\mathcal{I})$ by setting

- $V_\mathcal{I} := \Delta^\mathcal{I}$,

- $E_\mathcal{I} := \bigcup_{r \in N_r} \{(v, r, w) | (v, w) \in r^\mathcal{I}\}$, and

- $\ell_\mathcal{I}(v) := \{P \in N_c | v \in P^\mathcal{I}\}$.

Example 1 *Let $N_c = \{P, Q, R\}$ and $N_r = \{r, s\}$. Figure 2.1 shows the $\mathcal{EL}^{(n)}$-description tree $\mathcal{T}_C$ and the graph $\mathcal{G}_\mathcal{I}$ for the following $\mathcal{EL}^{(n)}$-concept term C and the interpretation $\mathcal{I}$:*

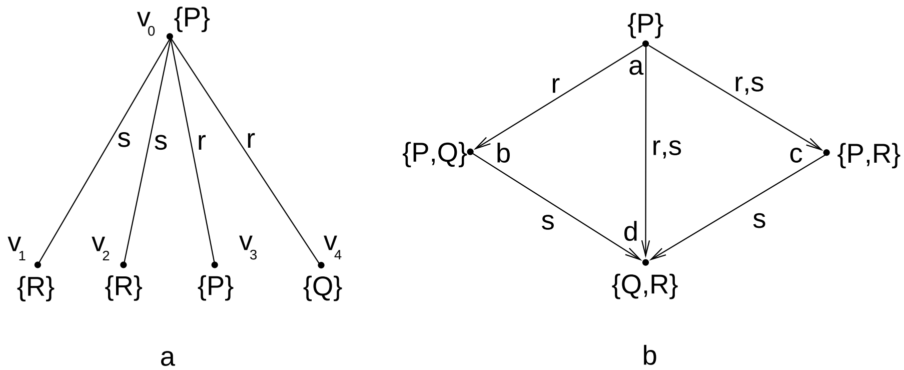

Figure 2.1: $\mathcal{EL}^{(n)}$-description tree $\mathcal{T}_C$ for the restricted $\mathcal{EL}^{(n)}$-concept term C (a) and the $\mathcal{EL}^{(n)}$-description graph $\mathcal{G}_\mathcal{I}$ for the interpretation $\mathcal{I}$ (b) from Example 1.

- $C := P \sqcap \exists s.(R, R) \sqcap \exists r.(P, Q)$,
- $\Delta^\mathcal{I} := \{a, b, c, d\}$, $P^\mathcal{I} := \{a, b, c\}$, $Q^\mathcal{I} := \{b, d\}$, $R^\mathcal{I} := \{c, d\}$, $r^\mathcal{I} := \{(a, b), (a, c), (a, d)\}$, $s^\mathcal{I} := \{(a, c), (a, d), (b, d), (c, d)\}$.

2.3 Subsumption in restricted $\mathcal{EL}^{(n)}$

In this section, we show that subsumption between two restricted $\mathcal{EL}^{(n)}$-concept terms corresponds to the existence of a local monomorphism between the corresponding $\mathcal{EL}^{(n)}$-description trees. In addition, we demonstrate that subsumption in restricted $\mathcal{EL}^{(n)}$ is decidable in polynomial time, since the existence of a local monomorphism between $\mathcal{EL}^{(n)}$-description trees is a polynomial time problem.

Definition 5 (Local monomorphism) *Let $\mathcal{G}_i = (V_i, E_i, \ell_i)$, $i = 1, 2$ be two $\mathcal{EL}^{(n)}$-description graphs. A mapping $h : V_1 \longrightarrow V_2$ is called* local monomorphism *from $\mathcal{G}_1$ to $\mathcal{G}_2$ if the following conditions are satisfied for every node $v \in V_1$:*

(h1) $\ell_1(v) \subseteq \ell_2\big(h(v)\big)$, *and*

(m1) For all $r \in N_r$, if $S_r^{E_1}(v) = \{w_1, \ldots, w_n\}$ for some $n > 0$, then there exist pairwise distinct $w'_1, \ldots, w'_n \in S_r^{E_2}\big(h(v)\big)$ such that $w'_i = h(w_i)$, for $i = 0, \ldots, n$.

Recalling Example 1, we note that for the concept term C and the interpretation $\mathcal{I}$, there exists a local monomorphism $h : \mathcal{T}_C \longrightarrow \mathcal{G}_\mathcal{I}$ that can be defined, e.g., as follows: $h(v_0) := a$, $h(v_1) := d$, $h(v_2) := c$, $h(v_3) := b$, $h(v_4) := d$.

Indeed, h satisfies all conditions that a local monomorphism should satisfy according to the Definition 5.

Theorem 1 *Let C be a restricted $\mathcal{EL}^{(n)}$-concept term, $\mathcal{T}_C = (V_1, E_1, \ell_1)$ the corresponding $\mathcal{EL}^{(n)}$-description tree with root v_0, and $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ an interpretation with the corresponding graph $\mathcal{G}_{\mathcal{I}} = (V_2, E_2, \ell_2)$. Then for every $a \in \Delta^{\mathcal{I}}$, the following are equivalent:*

1. *$a \in C^{\mathcal{I}}$*
2. *There exists a local monomorphism $h : \mathcal{T}_C \longrightarrow \mathcal{G}_{\mathcal{I}}$ with $h(v_0) = a$.*

Proof

$(1 \rightarrow 2)$ Let $a \in C^{\mathcal{I}}$. We prove the existence of an appropriate local monomorphism h by induction on $\mathsf{rdepth}(C)$.

Base case. Let $\mathsf{rdepth}(C) = 0$. Then $C = P_1 \sqcap \ldots \sqcap P_k$, i.e., $\ell_1(v_0) = \{P_1, \ldots, P_k\}$. Since $a \in C^{\mathcal{I}}$, then in particular $a \in P_1^{\mathcal{I}} \cap \ldots \cap P_k^{\mathcal{I}}$, i.e., $\{P_1, \ldots, P_k\} \subseteq \ell_2(a)$. Thus, $\ell_1(v_0) \subseteq \ell_2(a)$, and the mapping h such that $h(v_0) = a$ satisfies Condition *(h1)* from Definition 5. Condition *(m1)* is trivially satisfied by h since $S_r^{E_1} = \emptyset$ for each role name $r \in N_r$.

Induction step. Let $\mathsf{rdepth}(C) = n$, for some $n > 0$. Assume that $C = P_1 \sqcap \ldots \sqcap P_k \sqcap \exists r_1.(C_1^1, \ldots, C_{n_1}^1) \sqcap \ldots \sqcap \exists r_m.(C_1^m, \ldots, C_{n_m}^m)$. Since $a \in C^{\mathcal{I}}$, for every $i \in \{1, \ldots, m\}$, there exist n_i pairwise distinct r_i-successors $b_1^i, \ldots, b_{n_i}^i$ of the node a such that $b_1^i \in \left(C_1^i\right)^{\mathcal{I}}, \ldots, b_{n_i}^i \in \left(C_{n_i}^i\right)^{\mathcal{I}}$. Let $\mathcal{T}_1^i, \ldots, \mathcal{T}_{n_i}^i$ be the $\mathcal{EL}^{(n)}$-description trees corresponding to $C_1^i, \ldots, C_{n_i}^i$, respectively. Let v_j^i denote the root of the $\mathcal{EL}^{(n)}$-description tree $\mathcal{T}_j^i$, for each $j = 1, \ldots, n_i$, and assume without loss of generality that all the trees $\mathcal{T}_j^i$ are disjoint. Obviously, $\mathsf{rdepth}(C_j^i) < n$ for all j, which by induction hypothesis yields the existence of a local monomorphisms $h_j^i : \mathcal{T}_j^i \longrightarrow G_{\mathcal{I}}$ with $h_j^i(v_j^i) = b_j^i$, for each $j = 1, \ldots, n_i$. We define the mapping h as follows:

$$h(v) := \begin{cases} a, & \text{if } v = v_0 \\ h_j^i(v), & \text{if } v \text{ is a node of some tree } \mathcal{T}_j^i \end{cases}$$

and show that h satisfies the conditions *(h1)* and *(m1)*, for every node v of the tree $\mathcal{T}_C$. Let $v \in V_1$ be an arbitrary node of $\mathcal{T}_C$. We make a case distinction.

Case $v = v_0$. Analogously to the base case, $\ell_1(v) = \{P_1, \ldots, P_k\} \subseteq \ell_2(a)$ satisfying *(h1)*. The condition *(m1)* is satisfied for $v = v_0$ since by construction of h, $h(v_j^i) = b_j^i$, for every $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n_i\}$, where $\{v_1^i, \ldots, v_{n_i}^i\} = S_{r_i}^{E_1}(v_0)$ and $\{b_1^i, \ldots, b_{n_i}^i\} \subseteq S_{r_i}^{E_2}(a)$ with all the b_j^i pairwise distinct.

Case $v \neq v_0$. Since $v \neq v_0$, v is a node of a tree $\mathcal{T}_j^i$, for some i, j. The mapping h satisfies the conditions *(h1)* and *(m1)* for v, since for all nodes w of the tree $\mathcal{T}_j^i$, $h(w) = h_j^i(w)$ and h_j^i is a local monomorphism from $\mathcal{T}_j^i$ to $\mathcal{G}_{\mathcal{I}}$.

$(2 \rightarrow 1)$ Let h be a local monomorphism from $\mathcal{T}_C$ to $\mathcal{G}_{\mathcal{I}}$ such that $h(v_0) = a$. We show that $a \in C^{\mathcal{I}}$ by induction on $\mathsf{depth}(\mathcal{T}_C)$.

Base case. Let $\mathsf{depth}(\mathcal{T}_C) = 0$. Then $C = P_1 \sqcap \ldots \sqcap P_k$ and $V_1 = \{v_0\}$, $E_1 = \emptyset$. Since $h(v_0) = a$ and h satisfies *(h1)*, we know that $\{P_1, \ldots, P_k\} = \ell_1(v_0) \subseteq \ell_2(a)$. Thus, $a \in P_1^{\mathcal{I}} \cap \ldots \cap P_k^{\mathcal{I}} = C^{\mathcal{I}}$.

Induction step. Let $\mathsf{depth}(\mathcal{T}_C) = n$, for some $n > 0$. Then $C \equiv P_1 \sqcap \ldots \sqcap P_k \sqcap \exists r_1.(C_1^1, \ldots, C_{n_1}^1) \sqcap \ldots \sqcap \exists r_m.(C_1^m, \ldots, C_{n_m}^m)$. Analogously to the base case, we know that $a \in P_1^{\mathcal{I}} \cap \ldots \cap P_k^{\mathcal{I}}$. We show now that, for every $r_i \in \{r_1, \ldots, r_m\}$, $a \in \exists r_i.(C_1^i, \ldots, C_{n_i}^i)$. Since h satisfies *(m1)*, we know that there exist pairwise distinct nodes $b_1^i, \ldots, b_{n_i}^i \in S_{r_i}^{E_2}(a)$ such that $h(v_1^i) = b_1^i, \ldots, h(v_{n_i}^i) = b_{n_i}^i$, where v_j^i is the root of the tree $\mathcal{T}_j^i$ corresponding to the concept term C_j^i, for every $j = 1, \ldots, n_i$. Induction hypothesis then yields the following: $b_1^i \in \big(C_1^i\big)^{\mathcal{I}}, \ldots, b_{n_i}^i \in \big(C_{n_i}^i\big)^{\mathcal{I}}$, making $a \in \exists r_i.(C_1^i, \ldots, C_{n_i}^i)$. Thus, $a \in C^{\mathcal{I}}$. □

We are now ready to characterize subsumption in $\mathcal{EL}^{(n)}$.

Theorem 2 *For any restricted $\mathcal{EL}^{(n)}$-concept terms C and D, the following are equivalent:*

1. $C \sqsubseteq D$
2. *There exists a local monomorphism $h : \mathcal{T}_D \longrightarrow \mathcal{T}_C$ which maps the root of the $\mathcal{EL}^{(n)}$-description tree $\mathcal{T}_D$ to the root of the $\mathcal{EL}^{(n)}$-description tree $\mathcal{T}_C$.*

Proof $(2 \rightarrow 1)$ It suffices to show that, for each interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and each $a \in \Delta^{\mathcal{I}}$, we have $a \in C^{\mathcal{I}}$ implies $a \in D^{\mathcal{I}}$.

Assume $a \in C^{\mathcal{I}}$. Then, Theorem 1 yields existence of a local monomorphism $g : \mathcal{T}_C \longrightarrow \mathcal{G}_{\mathcal{I}}$ such that $g(w) = a$, where w is the root of the tree $\mathcal{T}_C$. The composition $g \circ h : \mathcal{T}_D \longrightarrow \mathcal{G}_{\mathcal{I}}$ that is defined as $(g \circ h)(u) := g\big(h(u)\big)$ is a local monomorphism with $g\big(h(v)\big) = g(w) = a$, where v is the root of the tree $\mathcal{T}_D$. Again, with Theorem 1, we obtain that $a \in D^{\mathcal{I}}$.

$(1 \rightarrow 2)$ Assume that there exists no local monomorphism from $\mathcal{T}_D$ to $\mathcal{T}_C$ such that v is mapped to w. The $\mathcal{EL}^{(n)}$-description tree $\mathcal{T}_C$ can be viewed as an $\mathcal{EL}^{(n)}$-description graph, i.e., there exists an interpretation $\mathcal{I}$ such that the $\mathcal{EL}^{(n)}$-description graph $\mathcal{G}_{\mathcal{I}}$ of $\mathcal{I}$ coincides with $\mathcal{T}_C$. For the element $w \in \Delta^{\mathcal{I}}$, we have:

- $w \in C^{\mathcal{I}}$, since the identity mapping is a local monomorphism from $\mathcal{T}_C$ to $\mathcal{T}_C = \mathcal{G}_{\mathcal{I}}$ that maps w to w.
- $w \notin D^{\mathcal{I}}$, since there exists no local monomorphism from $\mathcal{T}_D$ to $\mathcal{T}_C$ that maps v to w.

This yields the desired result $C \not\sqsubseteq D$. □

Theorem 2 establishes the correspondence between the subsumption problem in restricted $\mathcal{EL}^{(n)}$ and the existence of a local monomorphism between $\mathcal{EL}^{(n)}$-description trees. In order to show that subsumption in restricted $\mathcal{EL}^{(n)}$ is polynomial, we have to show that the existence of a local monomorphism between the respective trees can be decided in polynomial time. For this, we employ a slight modification of the algorithm proposed in [BKM99]. But before doing so, we revise some additional notions from graph theory.

Definition 6 (Bipartite graph) *A triple $\mathcal{G} = (I, O, E)$ is called a* bipartite graph *if I and O are disjoint sets of vertices and $E \subseteq I \times O$ is a set of edges such that each edge $e \in E$ is incident to two nodes $a \in I$ and $b \in O$.*

The partitioning of the set of vertices of a bipartite graph into two partitions I and O is crucial in the context of finding the so-called *matching* within the graph.

Definition 7 (Matching) *Let $\mathcal{G} = (I, O, E)$ be a bipartite graph. A* matching *in $\mathcal{G}$ is a set of edges $M \subseteq E$, such that for all $m_1, m_2 \in M$, if $m_1 = (a_1, b_1)$ and $m_2 = (a_2, b_2)$ then $m_1 \neq m_2$ implies $a_1 \neq a_2$ and $b_1 \neq b_2$. A matching M is called* maximum *in $\mathcal{G}$ if it has the maximum cardinality among all possible matchings in $\mathcal{G}$. A matching M is called* left-total *if for each $a \in I$ there exists a node $b \in O$ such that $(a, b) \in M$.*

One should note that a maximum matching in a bipartite graph must not be unique. It is possible to have several different maximum matchings with the same cardinality.

Let $\mathcal{T}_C = (V_1, E_1, \ell_1)$, and $\mathcal{T}_D = (V_2, E_2, \ell_2)$ be two $\mathcal{EL}^{(n)}$-description trees with root nodes v_1 and v_2, respectively. Assume that we check the existence of a local monomorphism from $\mathcal{T}_D$ to $\mathcal{T}_C$.

First, we introduce an additional marking function $\ell' : V_2 \longrightarrow 2^{V_1}$. The meaning of ℓ' can be described as follows: $w \in \ell'(v)$ iff there exists a local monomorphism $h' : \mathcal{T}_D(v) \longrightarrow \mathcal{T}_C(w)$ with $h'(v) = w$. Thus, the required local monomorphism $h : \mathcal{T}_D \longrightarrow \mathcal{T}_C$ exists if $v_1 \in \ell'(v_2)$.

The marking function is constructed bottom-up. Namely, for a node $v \in V_2$, we check whether the following conditions are satisfied, for each node $w \in V_1$:

(h1') $\ell_2(v) \subseteq \ell_1(w)$,

(m1') For each $r \in N_r$, there exists a left-total matching M in the bipartite graph $\mathcal{G} = \big(S_r^{E_2}(v), S_r^{E_1}(w), E\big)$, where $E := \big\{(a_2, a_1) \in S_r^{E_1}(w) \times S_r^{E_2}(v) | a_1 \in \ell'(a_2)\big\}$.

The marking function $\ell'(v)$ contains precisely those nodes $w \in V_1$ that satisfy the above Conditions *(h1')* and *(m1')*. Due to the bottom-up construction, the relevant markings for the nodes from V_2 that are successors of v have already been computed before processing the node v itself. The algorithm performs $\mathcal{O}\big(n^3\mu_M(n)\big)$ steps, where n is the size of the input and $\mu_M(n)$ is the complexity of the problem of finding a left-total matching in a bipartite graph $\mathcal{G} = (I, O, E)$ with $|I \cup O| = n$. This problem can be reduced to finding a maximum matching M in $\mathcal{G}$ and checking, whether $|M| \geq |I|$. In the literature, there exist several algorithms for solving the maximum matching problem, including the one in [HK73] with the complexity bounds of $\mathcal{O}(n^{5/2})$.

The following Lemma states the correctness of the algorithm for construction the marking function ℓ'.

Lemma 3 *Let $\mathcal{T}_i = (V_i, E_i, \ell_i)$, $i = 1, 2$, be two $\mathcal{EL}^{(n)}$-description trees. Then, for every $v_1 \in V_1$ and every $v_2 \in V_2$, if $v_1 \in \ell'(v_2)$ then there exists a local monomorphism $h : \mathcal{T}_2(v_2) \longrightarrow \mathcal{T}_1(v_1)$ with $h(v_2) = v_1$.*

Proof We reason by induction on $\mathsf{depth}\big(\mathcal{T}_2(v_2)\big)$.

Base case. Let $\mathsf{depth}\big(\mathcal{T}_2(v_2)\big) = 0$, i.e., v_2 is a leaf node. By construction of the marking function ℓ', for every $v_1 \in V_1$, $v_1 \in \ell'(v_2)$ iff $\ell_2(v_2) \subseteq \ell_1(v_1)$. But then, a mapping $h : \mathcal{T}_2(v_2) \longrightarrow \mathcal{T}_1(v_1)$ with $h(v_2) = v_1$ satisfies the condition *(h1)* of being a local monomorphism. The condition *(m1)* is trivially satisfied by h, since $S_r^{E_2}(v_2) = \emptyset$, for each $r \in N_r$.

Induction step. Let $\mathsf{depth}\big(\mathcal{T}_2(v_2)\big) = n$, for some $n > 0$. Let v_1 be an arbitrary node of the tree $\mathcal{T}_1$ such that $v_1 \in \ell'(v_2)$. Since $v_1 \in \ell'(v_2)$, we know by condition *(h1')* that $\ell_2(v_2) \subseteq \ell_1(v_1)$. The condition *(m1')* together with the definition of a left-total matching imply that for every role name r, if $S_r^{E_2}(v_2) = \{w_1^r, \ldots, w_n^r\}$ then there exist pairwise distinct nodes $u_1^r, \ldots, u_n^r \in S_r^{E_1}(v_1)$ such that $u_1^r \in \ell'(w_1^r), \ldots, u_n^r \in \ell'(w_n^r)$. By induction hypothesis, the latter implies the existence of local monomorphisms $h_i^r : \mathcal{T}_1(w_i^r) \longrightarrow \mathcal{T}_2(u_i^r)$ with $h_i^r(w_i^r) = u_i^r$, for $i = 1, \ldots, n$.

Thus, we can construct a local monomorphism $h : \mathcal{T}_1(v_1) \longrightarrow \mathcal{T}_2(v_2)$ as follows:

$$h(v) := \begin{cases} v_2, & \text{if } v = v_1 \\ h_i^r(v), & \text{if } v \text{ is a node of } \mathcal{T}_1(w_i^r)\text{, for some } r \in N_r,\quad w_i^r \in S_r^{E_1}(v_1) \end{cases}$$

□

Based on the aforementioned observations, we can now state the main result of this section.

Theorem 4 *Subsumption in restricted $\mathcal{EL}^{(n)}$ can be decided in polynomial time.*

Chapter 3

Reasoning in restricted $\mathcal{EL}^{(n)}$ with respect to acyclic TBoxes

In the previous chapter, we have shown that subsumption between two restricted $\mathcal{EL}^{(n)}$-concept terms can be decided in polynomial time. In this chapter, we investigate the complexity of subsumption in restricted $\mathcal{EL}^{(n)}$ with respect to acyclic terminologies (or TBoxes, for short).

Similar to the case of restricted $\mathcal{EL}^{(n)}$-concept terms, we show that subsumption in restricted $\mathcal{EL}^{(n)}$ w.r.t. acyclic TBoxes corresponds to the existence of an embedding between the corresponding $\mathcal{EL}^{(n)}$-description forests. We also provide a polynomial time algorithm for verifying the existence of such an embedding.

3.1 Restricted $\mathcal{EL}^{(n)}$-TBoxes

Here, we define formally the notion of a TBox and then we look closely at how allowing for TBoxes influences the complexity of subsumption.

A *concept definition* is a definition of the form $A \equiv D$, where A is a concept name and D is an $\mathcal{EL}^{(n)}$-concept term.

An *acyclic* $\mathcal{EL}^{(n)}$*-TBox* is a set of concept definitions $\mathcal{T} = \{A_1 \equiv D_1, \ldots, A_n \equiv D_n\}$ that contains no definitions $B_1 \equiv C_1, \ldots, B_n \equiv C_m \in \mathcal{T}$, $m > 0$ such that

- C_i contains B_{i+1}, for $1 \leq i < m$ and
- C_m contains B_1

and satisfies the condition that, for every concept name A, there is at most one definition $A \equiv D$ in $\mathcal{T}$, for some $\mathcal{EL}^{(n)}$-concept term D. The concept names $A_1, \ldots, A_n$ are called *defined* and the set of defined concept names is denoted with N_{def}. The concept names that are not defined are called *primitive*.

Any defined concept name can be expanded with respect to an acyclic TBox $\mathcal{T}$. By expanding we mean exhaustively replacing every defined concept name A with the corresponding concept term D, where $A \equiv D$ is the definition of A in $\mathcal{T}$. Since we consider now only restricted concept terms, we should guarantee that expanding preserves restrictedness of concept terms.

Definition 8 (Restricted $\mathcal{EL}^{(n)}$-TBox) *Let $\mathcal{T} = \{A_1 \equiv D_1, \ldots, A_n \equiv D_n\}$ be an acyclic $\mathcal{EL}^{(n)}$-TBox. Its* expansion *is the set of concept definitions $\mathcal{T}' = \{A_1 \equiv D'_1, \ldots, A_n \equiv D'_n\}$, which is obtained by expanding all concepts on the right-hand sides of the definitions in $\mathcal{T}$. $\mathcal{T}$ is called a* restricted $\mathcal{EL}^{(n)}$-TBox, *if in the expansion $\mathcal{T}'$ of $\mathcal{T}$, every D'_i is a restricted $\mathcal{EL}^{(n)}$-concept term, for $i = 1, \ldots, n$.*

In the rest of this chapter we will use the notions of *restricted $\mathcal{EL}^{(n)}$-TBox* and *$\mathcal{EL}^{(n)}$-TBox* interchangeably, with both terms meaning a restricted $\mathcal{EL}^{(n)}$-TBox.

Definition 9 (Semantics) *An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is called a* model *for a TBox $\mathcal{T}$ iff, for every $A \equiv D \in \mathcal{T}$, $A^{\mathcal{I}} = D^{\mathcal{I}}$. A concept term C is said to be* subsumed *by a concept term D* with respect to $\mathcal{T}$, *written $C \sqsubseteq_{\mathcal{T}} D$ iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for every model $\mathcal{I}$ of $\mathcal{T}$.*

The notion of an $\mathcal{EL}^{(n)}$-TBox is illustrated with the following example.

Example 2 *Let $\mathcal{T}$ be the following set of concept definitions:*

$$\begin{aligned} \mathcal{T} := \big\{ A_1 &\equiv P_1 \sqcap \exists r_1.(P_1, P_2, P_3) \\ A_2 &\equiv A_1 \sqcap A_3 \sqcap P_2 \sqcap \exists r_2.(P_3, P_4) \\ A_3 &\equiv P_2 \sqcap P_3 \sqcap \exists r_3.(A_1 \sqcap P_2, P_3) \big\}, \end{aligned}$$

where P_1, P_2, P_3 are primitive concept names and r_1, r_2, r_3 are role names. It is easy to see that $\mathcal{T}$ is an acyclic $\mathcal{EL}^{(n)}$-TBox. In order to check whether $\mathcal{T}$ is restricted, we build the expansion $\mathcal{T}'$ of $\mathcal{T}$:

$$\begin{aligned} \mathcal{T}' := \big\{ A_1 &\equiv P_1 \sqcap \exists r_1.(P_1, P_2, P_3) \\ A_2 &\equiv P_1 \sqcap P_2 \sqcap P_3 \sqcap \exists r_1.(P_1, P_2, P_3) \sqcap \exists r_2.(P_3, P_4) \sqcap \exists r_3.(A_1 \sqcap P_2, P_3) \\ A_3 &\equiv P_2 \sqcap P_3 \sqcap \exists r_3.(A_1 \sqcap P_2, P_3) \big\}, \end{aligned}$$

Since all the concept terms on the right-hand sides of the definitions in $\mathcal{T}'$ are restricted $\mathcal{EL}^{(n)}$-concept terms, $\mathcal{T}$ is indeed a restricted $\mathcal{EL}^{(n)}$-TBox.

3.2 $\mathcal{EL}^{(n)}$-description forests

In this section, we extend the notion of $\mathcal{EL}^{(n)}$-description tree from a single concept term to a TBox. When building such extension, one should be particularly careful since one concept definition in a TBox may contain concept names defined in other concept definitions. The extended structure should reflect these dependencies.

Definition 10 *An* extended $\mathcal{EL}^{(n)}$-description graph *is a structure* $\mathcal{G} = (V, E, \ell, \mathcal{E})$*, where*

- (V, E, ℓ) *is an* $\mathcal{EL}^{(n)}$*-description graph, and*
- $\mathcal{E}$ *is a binary relation on* V*.*

An extended $\mathcal{EL}^{(n)}$*-description graph* $\mathcal{G} = (V, E, \ell, \mathcal{E})$ *is called* $\mathcal{EL}^{(n)}$-description forest *if* (V, E) *is a forest.*

$\mathcal{EL}^{(n)}$-description forests are used to represent $\mathcal{EL}^{(n)}$-TBoxes, whereas extended $\mathcal{EL}^{(n)}$-description graphs represent interpretations. The construction of extended $\mathcal{EL}^{(n)}$-description graphs and $\mathcal{EL}^{(n)}$-description forests proceeds as follows.

Let $\mathcal{I} =$ be an interpretation. The *extended* $\mathcal{EL}^{(n)}$*-description graph corresponding to* $\mathcal{I}$ is the tuple $\mathcal{G}_\mathcal{I} = (V, E, \ell, \mathcal{E})$, where

- (V, E, ℓ) is the $\mathcal{EL}^{(n)}$-description graph corresponding to $\mathcal{I}$,
- $\mathcal{E}$ is the empty relation.

Let $\mathcal{T} = \{A_1 \equiv D_1, \ldots, A_n \equiv D_n\}$, $n > 0$, be a restricted $\mathcal{EL}^{(n)}$-TBox. For $1 \leq i \leq n$, let $\mathcal{T}_i = (V_i, E_i, \ell_i)$ be the $\mathcal{EL}^{(n)}$-description tree corresponding to D_i with v_i denoting the root of $\mathcal{T}_i$. W.l.o.g., assume that all V_i are pairwise disjoint. The $\mathcal{EL}^{(n)}$*-description forest* $\mathcal{F}_\mathcal{T} := (V, E, \ell, \mathcal{E})$ *corresponding to* $\mathcal{T}$ is defined as follows:

- $V := \bigcup_{1 \leq i \leq n} V_i$,
- $E := \bigcup_{1 \leq i \leq n} E_i$,
- for every $v \in V_i$, $\ell(v) := \ell_i(v) \setminus N_{def}$, $1 \leq i \leq n$,
- $\mathcal{E} \subseteq V \times \{v_1, \ldots, v_n\}$ is defined as follows:

$$\mathcal{E} := \{(v, v_i) | \exists j \in \{1, \ldots, n\}. v \in V_j, A_i \in \ell_j(v)\}.$$

The binary relation $\mathcal{E}$ in the definition of an extended $\mathcal{EL}^{(n)}$-description graph $\mathcal{G} = (V, E, \ell, \mathcal{E})$ plays a crucial role in the construction of the $\mathcal{EL}^{(n)}$-description forests for restricted $\mathcal{EL}^{(n)}$-TBoxes. Thus, we state the meaning of $\mathcal{E}$ explicitly.

Two nodes v_1 and v_2 are in the reflexive-transitive closure $\mathcal{E}^*$ of $\mathcal{E}$, i.e., $(v_2, v_1) \in \mathcal{E}^*$ if the node v_2 in addition to its own labels and connected nodes, inherits those of the node v_1.

Assume that we are given a TBox $\mathcal{T}$ consisting of two concept definitions $A_1 \equiv C$ and $A_2 \equiv A_1 \sqcap D$, where C and D are $\mathcal{EL}^{(n)}$-concept terms. Note, that the definition of A_2 contains the defined concept name A_1 as a top-level conjunct. If we were to construct an $\mathcal{EL}^{(n)}$-description tree $\mathcal{T}_2$ for the definition of A_2 with respect to $\mathcal{T}$, we would need to insert a copy of the $\mathcal{EL}^{(n)}$-description tree $\mathcal{T}_1$ built for the definition of A_1 into $\mathcal{T}_2$. In general, such expanding of concept definitions and their corresponding $\mathcal{EL}^{(n)}$-description trees leads to an exponential blow-up.

In order to prevent such expensive construction, we employ the so-called structure sharing technique by means of the relation $\mathcal{E}$. Namely, in order to reflect the top-level dependency between concept names A_1 and A_2, we simply require that the root nodes v_1 and v_2 of the corresponding trees $\mathcal{T}_1$ and $\mathcal{T}_2$ are in the relation $\mathcal{E}$.

We illustrate the construction of an $\mathcal{EL}^{(n)}$-description forest and the importance of the relation $\mathcal{E}$ by the following example

Example 3 *Let $\mathcal{T}$ be the TBox defined in Example 2, i.e.*

$$\begin{aligned} \mathcal{T} := \{ A_1 &\equiv P_1 \sqcap \exists r_1.(P_1, P_2, P_3) \\ A_2 &\equiv A_1 \sqcap A_3 \sqcap P_2 \sqcap \exists r_2.(P_3, P_4) \\ A_3 &\equiv P_2 \sqcap P_3 \sqcap \exists r_3.(A_1 \sqcap P_2, P_3) \}. \end{aligned}$$

Figure 3.1 shows the $\mathcal{EL}^{(n)}$-description forest $\mathcal{F}_\mathcal{T}$ that corresponds to the TBox $\mathcal{T}$. The relation $\mathcal{E}$ is depicted by means of dashed lines.

One should note that the relation $\mathcal{E}$ in the $\mathcal{EL}^{(n)}$-description forest corresponding to the *expansion* $\mathcal{T}'$ of some TBox $\mathcal{T}$ is always empty, and thus, such a forest can be seen as an extended $\mathcal{EL}^{(n)}$-description graph corresponding to some interpretation $\mathcal{I}$.

In order to exploit the structure sharing technique explained above, it is necessary to introduce additional notions.

Definition 11 *Let $\mathcal{F} = (V, E, \ell, \mathcal{E})$ be an $\mathcal{EL}^{(n)}$-description forest. Let $\mathcal{E}^*$ be the reflexive-transitive closure of $\mathcal{E}$, and let $\mathcal{E}^*(x) := \{x' | (x, x') \in \mathcal{E}^*\}$. We define an additional labeling function $\ell^* : V \to 2^{N_c}$ as follows:*

$$\ell^*(x) := \bigcup_{x' \in \mathcal{E}^*(x)} \ell(x').$$

Similarly to the case of $\mathcal{EL}^{(n)}$-description trees, we define the $\mathcal{EL}^{(n)}$-concept term $C(v)$ corresponding to a node $v \in V$ as follows:

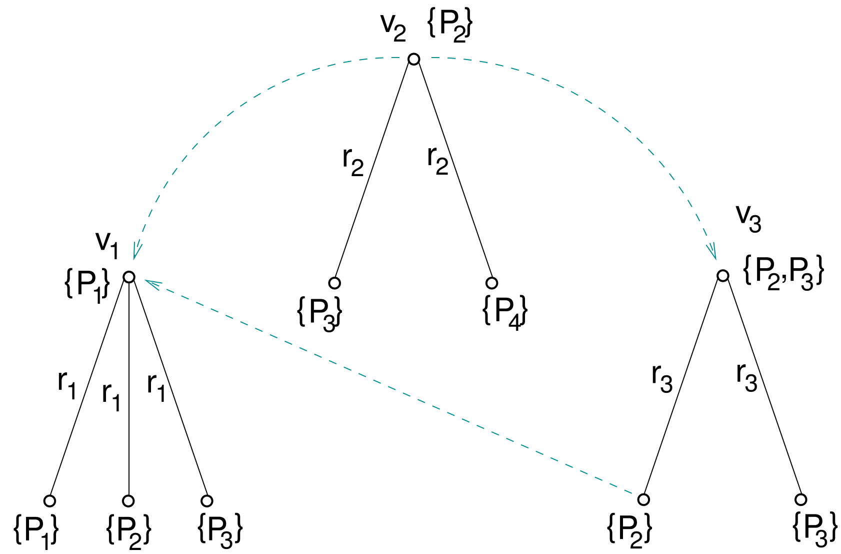

Figure 3.1: $\mathcal{EL}^{(n)}$-description forest $\mathcal{F}_\mathcal{T}$ for the TBox $\mathcal{T}$ from Example 3.

$$C(v) := \bigsqcap_{P \in \ell^*(v)} P \sqcap \bigsqcap_{\substack{r \in N_r, \\ w \in \mathcal{E}^*(v), \\ \emptyset \neq \{v_1, \ldots, v_n\} = S_r^E(w)}} \sqcap \exists r.\big(C(v_1), \ldots, C(v_n)\big),$$

where $S_r^E(w) = \{u | (w, r, u) \in E\}$.

Note that $C(v)$ does not contain defined concept names; it is an $\mathcal{EL}^{(n)}$-concept term that is expanded with respect to the underlying TBox. Thus, in particular, $C(v)$ is restricted if the TBox is restricted.

3.3 Subsumption in restricted $\mathcal{EL}^{(n)}$ with acyclic TBoxes

Analogously to the case of restricted $\mathcal{EL}^{(n)}$-concept terms, we exploit graph representations in order to prove the tractability of subsumption in restricted $\mathcal{EL}^{(n)}$ w.r.t. acyclic TBoxes. We characterize the subsumption problem between two concept names A and B defined in an $\mathcal{EL}^{(n)}$-TBox $\mathcal{T}$ by the existence of an embedding between the roots of the $\mathcal{EL}^{(n)}$-description trees corresponding to the definitions of A and B, respectively.

Let $\mathcal{F} = (V, E, \ell, \mathcal{E})$ be an extended $\mathcal{EL}^{(n)}$-description graph. We define the function $\mathsf{reach} : V \to 2^V$ as follows: For each $w \in V$, $\mathsf{reach}(w) \subseteq V$ is the smallest set such that

- $w \in \mathsf{reach}(w)$,

- if $v \in \mathsf{reach}(w)$ and $(v, r, u) \in E$, for some $r \in N_r$, then $u \in \mathsf{reach}(w)$,
- if $v \in \mathsf{reach}(w)$ and $(v, u) \in \mathcal{E}$, then $u \in \mathsf{reach}(w)$.

Intuitively, $\mathsf{reach}(w)$ contains exactly those nodes $v \in V$, that are reachable from w via the edges from $E \cup \mathcal{E}$.

Definition 12 (Embedding) *Let $\mathcal{G}_i = (V_i, E_i, \ell_i, \mathcal{E}_i)$, $i = 1, 2$, be two extended $\mathcal{EL}^{(n)}$-description graphs. Let $v_1 \in V_1$ and $v_2 \in V_2$. An* embedding *from v_1 to v_2 is a binary relation $\mathcal{H} \subseteq \mathsf{reach}(v_1) \times \mathsf{reach}(v_2)$ with $(v_1, v_2) \in \mathcal{H}$ such that, for each $(v, u) \in \mathcal{H}$:*

(s1) $\ell_1^*(v) \subseteq \ell_2^*(u)$, *and*

(s2) for all $r \in N_r$, if $S_r^{E_1}(v') = \{v_1, \dots, v_n\}$, for some $n > 0$, $v' \in \mathcal{E}_1^*(v)$, then there exists $u' \in \mathcal{E}_2^*(u)$ and pairwise distinct $u_1, \dots, u_n \in S_r^{E_2}(u')$ such that $(v_i, u_i) \in \mathcal{H}$, for $i = 1, \dots, n$.

v_1 *is* embeddable *into v_2 if there exists an embedding from v_1 to v_2.*

The notion of embedding allows us to characterize a relation between elements of an interpretation and the nodes of an $\mathcal{EL}^{(n)}$-description forest.

Theorem 5 *Let $\mathcal{T}$ be a TBox with the corresponding $\mathcal{EL}^{(n)}$-description forest $\mathcal{F}_\mathcal{T} = (V_1, E_1, \ell_1, \mathcal{E}_1)$, and $\mathcal{I}$ an interpretation with the extended $\mathcal{EL}^{(n)}$-description graph $\mathcal{G}_\mathcal{I} = (V_2, E_2, \ell_2, \mathcal{E}_2)$. Then for every $w \in V_1$, $a \in V_2$, the following are equivalent:*

1. *$a \in C(w)^\mathcal{I}$,*
2. *There exists an embedding $\mathcal{H}$ from w to a.*

Proof $(1 \to 2)$. Let $a \in C(w)^\mathcal{I}$. We prove the existence of the embedding $\mathcal{H}$ from w to a by induction on $\mathsf{cdepth}\big(C(w)\big)$.

Base case. Let $\mathsf{cdepth}\big(C(w)\big) = 0$, i.e., $C(w) = P_1 \sqcap \ldots \sqcap P_k$. Then $\ell_1^*(w) = \{P_1, \dots, P_k\}$, $S_r^{E_1}(u) = \emptyset$, for each $u \in \mathcal{E}_1^*(w)$ and each $r \in N_r$. Thus, $\mathsf{reach}(w) = \mathcal{E}_1^*(w)$. We define the embedding $\mathcal{H}$ as follows: $\mathcal{H} := \bigcup_{u \in \mathcal{E}_1^*(w)} \{(u, a)\}$.

Obviously, since $w \in \mathcal{E}_1^*(w)$, we have that $(w, a) \in \mathcal{H}$, and now we show that $\mathcal{H}$ satisfies Conditions *(s1)* and *(s2)* from Definition 12.

(s1) holds since the assumption that $a \in C(w)^\mathcal{I}$ implies that, for each $u \in \mathcal{E}_1^*(w)$, $\ell_1^*(u) \subseteq \ell_1^*(w) = \{P_1, \dots, P_k\} \subseteq \ell_2(a) = \ell_2^*(a)$.

Condition *(s2)* is trivially satisfied since $S_r^{E_1}(u) = \emptyset$, for each $u \in \mathcal{E}_1^*(w)$, $r \in N_r$.

Induction step. Let $\mathsf{cdepth}\big(C(w)\big) = n > 0$, i.e., $C(w) = P_1 \sqcap \ldots \sqcap P_k \sqcap \exists r_1.(C_1^1, \dots, C_{n_1}^1) \sqcap \ldots \sqcap \exists r_m.(C_1^m, \dots, C_{n_m}^m)$. For each $i = 1, \dots, m$, $j = 1, \dots, n_i$, let $\mathcal{T}_j^i$ denote the $\mathcal{EL}^{(n)}$-description tree corresponding to C_j^i in the $\mathcal{EL}^{(n)}$-description forest $\mathcal{F}_\mathcal{T}$. With v_j^i we denote the root of $\mathcal{T}_j^i$. We note that any node from $\mathsf{reach}(w)$ either equals w or is a node of some tree $\mathcal{T}_j^i$. Since $a \in C(w)^\mathcal{I}$ then, in particular, for every $i = 1, \ldots, m$, $a \in \big(\exists r_i.(C_1^i, \ldots, C_{n_i}^i)\big)^\mathcal{I}$ and thus there exist pairwise distinct nodes $b_1^i, \ldots, b_{n_i}^i \in S_r^{E_2}(a)$ such that $b_1^i \in \big(C_1^i\big)^\mathcal{I}, \ldots, b_{n_i}^i \in \big(C_{n_i}^i\big)^\mathcal{I}$. By induction hypothesis, there exist embeddings $\mathcal{H}_j^i$ from v_j^i to b_j^i, for each $i = 1, \ldots, m$, $j = 1, \ldots, n_j$. We define now the embedding $\mathcal{H}$ as follows:

$$\mathcal{H} := \bigcup_{u \in \mathcal{E}_1^*(w)} \{(u, a)\} \cup \bigcup_{i,j} \mathcal{H}_j^i$$

and show now that $\mathcal{H}$ satisfies the conditions *(s1)* and *(s2)*.

(s1). Let $(u, b) \in \mathcal{H}$. Case $b = a$. Then $\ell_1^*(u) \subseteq \ell_2^*(w) = \{P_1, \ldots, P_k\}$. Since $a \in C(w)^\mathcal{I}$, $\{P_1, \ldots, P_k\} \subseteq \ell_2^*(a)$, and thus $\ell_1^*(u) \subseteq \ell_2^*(b)$. Case $b \neq a$. Then *(s1)* is satisfied by $\mathcal{H}$, since $(u, b) \in \mathcal{H}_j^i$ and $\mathcal{H}_j^i$ is an embedding.

(s2). Let $(u, b) \in \mathcal{H}$, $r \in N_r$ and $S_r^{E_1}(u) = \{u_1, \ldots, u_n\}$. Case $b = a$. Then $r = r_i$, for some $i \in \{1, \ldots, m\}$ and $\{u_1, \ldots, u_n\} = \{v_1^i, \ldots, v_{n_i}^i\}$. By construction of $\mathcal{H}$ we have that

$(v_1^i, b_1^i) \in \mathcal{H}, \ldots, (v_{n_i}^i, b_{n_i}^i) \in \mathcal{H}$ with all $b_1^i, \ldots, b_{n_i}^i$ being pairwise distinct elements of $S_{r_i}^{E_2}(a)$. Thus, *(s2)* is satisfied. Case $b \neq a$. Then *(s2)* is satisfied by h for u since in this case $(u, b) \in \mathcal{H}_j^i$ and $\mathcal{H}_j^i$ is an embedding.

$(2 \to 1)$. Let $\mathcal{H}$ be an embedding from w to a. We prove by induction on $\mathsf{cdepth}\big(C(w)\big)$ that $a \in C(w)^\mathcal{I}$.

Base case. Let $\mathsf{cdepth}\big(C(w)\big) = 0$, i.e., $C(w) = P_1, \sqcap \ldots \sqcap P_k$. Then $\ell_1^*(w) = \{P_1, \ldots, P_l\}$. By condition *(s1)* we know that $\{P_1, \ldots, P_k\} \subseteq \ell_2^*(a) = \ell_2(a)$. Thus, $a \in P_1^\mathcal{I} \cap \ldots \cap P_k^\mathcal{I} = C(w)^\mathcal{I}$.

Induction step. Let $\mathsf{cdepth}\big(C(w)\big) = n > 0$, i.e., $C(w) = P_1 \sqcap \ldots \sqcap P_k \sqcap \exists r_1.(C_1^1, \ldots, C_{n_1}^1) \sqcap \ldots \sqcap \exists r_m.(C_1^m, \ldots, C_{n_m}^m)$. For each $i = 1, \ldots, m$, $j = 1, \ldots, n_i$, let $\mathcal{T}_j^i$ denote the $\mathcal{EL}^{(n)}$-description tree corresponding to C_j^i with v_j^i denoting the root of $\mathcal{T}_j^i$. Analogously to the base case, we have that $a \in P_1^\mathcal{I} \cap \ldots \cap P_k^\mathcal{I}$. We show now that $a \in \exists r_i.(C_1^i, \ldots, C_{n_i}^i)^\mathcal{I}$, for an arbitrary index $i \in \{1, \ldots, m\}$. By condition *(s2)*, we know that there exist pairwise distinct nodes $b_1^i, \ldots, b_{n_i}^i \in S_{r_i}^{E_2}(a)$ such that $(v_1^i, b_1^i) \in \mathcal{H}, \ldots, (v_{n_i}^i, b_{n_i}^i) \in \mathcal{H}$.

Since $\mathcal{H}$ is an embedding from w to a then, in particular, the conditions *(s1)* and *(s2)* are satisfied, for each $(u, b) \in \mathcal{H}$.

We define now relations $\mathcal{H}_1^i, \ldots, \mathcal{H}_{n_i}^i$ by setting $\mathcal{H}_j^i := \{(u, b) \in \mathcal{H} | u \in \mathsf{reach}(v_i^j)\}$, for each $j = 1, \ldots, n_i$. Obviously, $\mathsf{reach}(v_j^i) \subseteq \mathsf{reach}(w)$, and thus each $\mathcal{H}_j^i$ is, by construction, an embedding from v_j^i to b_j^i. By induction hypothesis, we have that $b_j^i \in \big(C(v_j^i)\big)^\mathcal{I} = \big(C_j^i\big)^\mathcal{I}$, for $j = 1, \ldots, n_i$. Since all b_j^i are pairwise distinct, we obtain the required result that $a \in \exists r_i.(C_1^i, \ldots, C_{n_i}^i)^\mathcal{I}$. Thus, $a \in C(w)^\mathcal{I}$. □

Theorem 5 establishes a connection between a node of an $\mathcal{EL}^{(n)}$-description forest and a node of an interpretation. Let $\mathcal{T} = \{A_1 \equiv D_1, \ldots, A_n \equiv D_n\}$ be an acyclic $\mathcal{EL}^{(n)}$-TBox, for some $n > 0$, with the corresponding $\mathcal{EL}^{(n)}$-description forest $\mathcal{F}_\mathcal{T} = (V, E, \ell, \mathcal{E})$. Let $v_i \in V$ be the root of the $\mathcal{EL}^{(n)}$-description tree $\mathcal{T}_i$ corresponding to the $\mathcal{EL}^{(n)}$-concept term D_i, for some $i \in \{1, \ldots, n\}$. From the definition of the function $C(w)$, it follows that $C(v_i)$ is the $\mathcal{EL}^{(n)}$-concept term D_i expanded with respect to the TBox $\mathcal{T}$. Or alternatively, $C(v_i) = D'_i$, where $\{A_1 \equiv D'_1, \ldots, A_n \equiv D'_n\}$ is the expansion $\mathcal{T}'$ of $\mathcal{T}$. Since any TBox $\mathcal{T}$ is equivalent to its expansion $\mathcal{T}'$ [BCM+03], the $\mathcal{EL}^{(n)}$-concept terms D_i and D'_i are equivalent, i.e., for any interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, for each $a \in \Delta^\mathcal{I}$, $a \in D_i^\mathcal{I}$ if and only if $a \in D_i'^\mathcal{I}$. Thus, D_i and $C(v_i)$ are equivalent, which lets us conclude with the following proposition.

Theorem 6 *Let $\mathcal{T} = \{A_1 \equiv D_1, \ldots, A_n \equiv D_n\}$ be a TBox with the corresponding $\mathcal{EL}^{(n)}$-description forest $\mathcal{F}_\mathcal{T}$ and $\mathcal{I}$ an interpretation with the extended $\mathcal{EL}^{(n)}$-description graph $\mathcal{G}_\mathcal{I} = (V_2, E_2, \ell_2, \mathcal{E}_2)$. Then for every defined concept name A_i, $i \in \{1, \ldots, n\}$ and any node $a \in V_2$, the following are equivalent:*

1. *$a \in A_i^\mathcal{I}$,*

2. *There exists an embedding $\mathcal{H}$ from v_i to a,*

where v_i is the root of the $\mathcal{EL}^{(n)}$-description tree $\mathcal{T}_i$ corresponding to D_i in $\mathcal{F}_\mathcal{T}$.

In the following, we extend the polynomial algorithm that has been introduced in Chapter 2 for deciding subsumption between restricted $\mathcal{EL}^{(n)}$-concept terms to the case of $\mathcal{EL}^{(n)}$-TBoxes. Let C_1 and C_2 be two $\mathcal{EL}^{(n)}$-concept terms, $\mathcal{T}$ a TBox and assume that we are requested to decide whether C_2 subsumes C_1 with respect to $\mathcal{T}$, i.e., whether $C_1 \sqsubseteq_\mathcal{T} C_2$. We reduce the subsumption problem from restricted concept terms to defined concept names by introducing two new concept definitions $A_1 \equiv C_1$ and $A_2 \equiv C_2$ to $\mathcal{T}$, where A_1 and A_2 are new concept names. In effect, we obtain a new TBox $\mathcal{T}'$ and the problem of whether C_2 subsumes C_1 w.r.t. $\mathcal{T}$ is reduced to the problem of whether A_2 subsumes A_1 w.r.t. $\mathcal{T}'$.

The following Theorem establishes the characterization of the subsumption problem between two defined concept names A_1 and A_2 with respect to a TBox through the existence of an embedding between the roots of the $\mathcal{EL}^{(n)}$-description trees that correspond to the definitions of A_1 and A_2 in the expanded $\mathcal{EL}^{(n)}$-description forest (i.e., in the $\mathcal{EL}^{(n)}$-description forest corresponding to the expansion of the TBox).

Theorem 7 *Let $\mathcal{T}$ be a TBox, $\hat{\mathcal{T}}$ its expansion and A_1, A_2 two defined concept names. Let $\hat{\mathcal{F}}$ be the $\mathcal{EL}^{(n)}$-description forest corresponding $\hat{\mathcal{T}}$, and v_1 and v_2 the roots of $\mathcal{EL}^{(n)}$-description trees $\hat{\mathcal{T}}_1$ and $\hat{\mathcal{T}}_2$ in $\hat{\mathcal{F}}$ that correspond to A_1 and A_2, respectively. Then the following are equivalent:*

1. $A_1 \sqsubseteq_{\hat{\mathcal{T}}} A_2$.

2. There exists an embedding $\hat{\mathcal{H}}$ *from* v_2 *to* v_1*.*

Proof ($2 \rightarrow 1$). It suffices to show that, for an arbitrary model $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ of $\hat{\mathcal{T}}$ and each $a \in \Delta^{\mathcal{I}}$, we have that $a \in A_1^{\mathcal{I}}$ implies $a \in A_2^{\mathcal{I}}$.

Assume $a \in A_1^{\mathcal{I}}$. By Theorem 6, there exists an embedding $\mathcal{G}$ from v_1 to a. Since the relational composition $\mathcal{G} \circ \hat{\mathcal{H}} := \{(u, w) \in V_2 \times V_1 | \exists v.(u, v) \in \hat{\mathcal{H}} \wedge (v, w) \in \mathcal{G}\}$ is an embedding from v_2 to a, by Theorem 6, we obtain that $a \in A_2^{\mathcal{I}}$.

($1 \rightarrow 2$) Assume that there exists no embedding from v_2 to v_1. Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be the interpretation such that its corresponding extended $\mathcal{EL}^{(n)}$-description graph is $\hat{\mathcal{F}}$. For the element $v_1 \in \Delta^{\mathcal{I}}$, we have:

- $v_1 \in A_1^{\mathcal{I}}$, since the identity relation is an embedding from v_1 to v_1.
- $v_1 \notin A_2^{\mathcal{I}}$, since there exists no embedding from v_2 to v_1.

Hence, $A_1 \not\sqsubseteq_{\mathcal{T}'} A_2$. □

Theorem 7 uses embeddings in the expanded $\mathcal{EL}^{(n)}$-description forest as a characterization of subsumption. However, working with expanded TBoxes and forests is very inefficient, since they are of exponential size in the size of the original TBox or forest. We now establish a similar characterization of subsumption using embeddings in (unexpanded) $\mathcal{EL}^{(n)}$-description forests.

Before doing so, we look more closely at expanded $\mathcal{EL}^{(n)}$-description forests. As we mentioned above, similarly to building the expansion $\hat{\mathcal{T}}$ of a TBox $\mathcal{T}$, one can see the construction of the $\mathcal{EL}^{(n)}$-description forest $\hat{\mathcal{F}}$ corresponding to $\hat{\mathcal{T}}$ as the expansion of the forest $\mathcal{F} = (V, E, \ell, \mathcal{E})$ corresponding to $\mathcal{T}$. During the expansion of the forest $\mathcal{F}$, one should eliminate all $\mathcal{E}$-links as follows: for every $(u, v) \in \mathcal{E}$, one makes a copy of the tree in $\mathcal{F}$ which has the root node v and plugs a copy under the node u. After this operation is done, the pair (u, v) can be removed from $\mathcal{E}$, and one proceeds until $\mathcal{E} = \emptyset$.

When the expansion process of $\mathcal{F}$ is accomplished, one should observe the following properties of the resulting $\mathcal{EL}^{(n)}$-description forest $\hat{\mathcal{F}} = (\hat{V}, \hat{E}, \hat{\ell}, \hat{\mathcal{E}})$:

(e1) $\hat{\mathcal{E}} = \emptyset$ and thus, for every $u \in \hat{V}$: $\hat{\mathcal{E}}^*(u) = \{u\}$ and $\hat{\ell}^*(u) = \hat{\ell}(u)$;

(e2) there are families of nodes $\{u_1, \ldots, u_n\} \subseteq \hat{V}$ that consist of copies of some node $u \in V$. For each $u \in V$, we denote the set of all copies of u (including u itself) that have been created during expansion as $\delta(u)$. Thus, in our example, $\delta(u) = \{u, u_1, \ldots, u_n\}$;

(e3) For each $u \in V$, if $\delta(u) = \{u, u_1, \ldots, u_n\}$, for some $n \geq 0$, then $\ell^*(u) = \hat{\ell}^*(u_1) = \ldots = \hat{\ell}^*(u_n) = \hat{\ell}(u_1) = \ldots = \hat{\ell}(u_n)$.

For simplicity we assume that $V \subseteq \hat{V}$ and $E \subseteq \hat{E}$, i.e., that the expansion $\hat{\mathcal{F}}$ of a forest $\mathcal{F}$ contains $\mathcal{F}$ itself. Thus, for every node $u \in \hat{V}$, $u \in \delta(v)$, for some $v \in V$, i.e., every node in the expanded forest is either a node in the original one or a copy of such a node. The same holds for the edges in $\hat{E}$.

The latter observation lets us show that embeddability is preserved under the expansion of an $\mathcal{EL}^{(n)}$-description forest. Indeed, assume that, for some $v_1, v_2 \in V$, there is an embedding $\mathcal{H}$ from v_1 to v_2 in $\mathcal{F}$ and we need to construct an embedding $\hat{\mathcal{H}}$ from v_1 to v_2 in $\hat{\mathcal{F}}$. Since $\hat{\mathcal{F}}$ consists of $\mathcal{F}$ and additionally copies of nodes and edges from $\mathcal{F}$, we need to extend $\mathcal{H}$ using the function δ as follows: $\hat{\mathcal{H}} := \{(u', v') | \exists u, v \in V_i.(u, v) \in \mathcal{H}, (u', v') \in \delta(u) \times \delta(v)\}$.

Conversely, suppose that $\hat{\mathcal{H}}$ is an embedding from v_1 to v_2 in $\hat{\mathcal{F}}$. Note that we can construct an embedding $\mathcal{H}$ from v_1 to v_2 in $\mathcal{F}$ only if $v_1, v_2 \in V$. This is the case, e.g., when v_1, v_2 are the root nodes of the $\mathcal{EL}^{(n)}$-description trees that correspond to definitions of some defined concept names A_1, A_2, respectively. If $v_1, v_2 \in V$ then, in particular $v_i \in \delta(v_i)$, $i = 1, 2$ and the following restriction $\mathcal{H}$ of $\hat{\mathcal{H}}$ is an embedding from v_1 to v_2 in $\mathcal{F}$: $\mathcal{H} := \hat{\mathcal{H}} \cap (V \times V)$.

Thus, we can conclude that, for any nodes $v, u \in V$, there exists an embedding $\mathcal{H}$ from v to u in $\mathcal{F}$ iff there exists an embedding $\hat{\mathcal{H}}$ from v to u in $\hat{\mathcal{F}}$. The latter statement together with Theorem 7 results in the following characterization of subsumption in $\mathcal{EL}^{(n)}$ w.r.t. restricted $\mathcal{EL}^{(n)}$-TBoxes.

Theorem 8 *Let $\mathcal{T}$ be a restricted $\mathcal{EL}^{(n)}$-TBox with the corresponding $\mathcal{EL}^{(n)}$-description forest $\mathcal{F}$. For $i = 1, 2$, let A_i be a defined concept name and v_i be the root of the $\mathcal{EL}^{(n)}$-description tree in $\mathcal{F}$ that corresponds to the definition of A_i in $\mathcal{T}$. Then the following are equivalent:*

1. *$A_1 \sqsubseteq_{\mathcal{T}} A_2$.*
2. *There exists an embedding $\mathcal{H}$ from v_2 to v_1.*

It remains to provide an algorithm for verifying the existence of an embedding and to show the polynomial runtime of the algorithm. Let $\mathcal{F}_{\mathcal{T}} = (V, E, \ell, \mathcal{E})$ be the $\mathcal{EL}^{(n)}$-description forest corresponding to the TBox $\mathcal{T}$. Let v_i be the root of the $\mathcal{EL}^{(n)}$-description tree corresponding to the definition of A_i in $\mathcal{T}$, $i = 1, 2$. Analogously to the case of single $\mathcal{EL}^{(n)}$-concept terms in Chapter 2, we introduce an additional marking function $\ell' : \mathsf{reach}(v_2) \longrightarrow 2^{\mathsf{reach}(v_1)}$, with the following intended meaning: $w \in \ell'(v)$ iff there exists an embedding $\mathcal{H}'$ from v to w. Obviously, there exists an embedding $\mathcal{H}$ from v_2 to v_1, if $v_1 \in \ell'(v_2)$.

The marking function is constructed bottom-up, i.e., starting with the nodes that have no successors with respect to E and $\mathcal{E}$ and terminating at the node v_2. For a node $v \in \mathsf{reach}(v_2)$, we check for each node $w \in \mathsf{reach}(v_1)$, whether the following conditions are satisfied:

- $\ell^*(v) \subseteq \ell^*(w)$,

- For each $r \in N_r$ and for each $v' \in \mathcal{E}^*(v)$, there exist $w' \in \mathcal{E}^*(w)$ and a left-total matching M in the bipartite graph $\mathcal{G} = (S_r^E(v'), S_r^E(w'), E')$, where $E' := \{(a_2, a_1) \in S_r^E(v') \times S_r^E(w') | a_1 \in \ell'(a_2)\}$.

The marking function $\ell'(v)$ contains precisely those nodes $w \in \mathsf{reach}(v_1)$ that satisfy these conditions. Due to the bottom-up construction, the relevant markings for the nodes from $\mathsf{reach}(v_2)$ that are successors of v have already been computed before processing the node v itself. The algorithm performs $\mathcal{O}\big(n^5 \mu_M(n)\big)$ steps, where n is the size of the input, $\mu_M(n)$ is the complexity of the problem of finding a left-total matching in a bipartite graph $\mathcal{G}$ that contains n nodes. As we have shown in Chapter 2, $\mu_M(n)$ is polynomial in n.

Thus, our algorithm for verifying the existence of an embedding is of polynomial complexity in the size of input. It implies the major result of this chapter which is reflected in the following theorem.

Theorem 9 *The subsumption problem in restricted $\mathcal{EL}^{(n)}$ with acyclic TBoxes can be decided in polynomial time.*

Chapter 4

Reasoning in unrestricted $\mathcal{EL}^{(n)}$ with general TBoxes

In the previous chapters, we were considering the restricted extension of the description logic $\mathcal{EL}$ with the new constructor $\exists r.(C_1, \ldots, C_n)$ that we called *restricted* $\mathcal{EL}^{(n)}$. We have shown that subsumption in restricted $\mathcal{EL}^{(n)}$ is polynomial when we consider two isolated concept terms as well as in the presence of restricted acyclic TBoxes. In this chapter, we investigate the complexity of subsumption in *unrestricted* $\mathcal{EL}^{(n)}$ with respect to general TBoxes. We show that, in this case, reasoning is no longer tractable since it becomes ExpTime-complete.

The syntactic restriction adopted in restricted $\mathcal{EL}^{(n)}$ was that in a concept term, it is disallowed to have more than one existential restriction for the same role name at the same conjunction level. In this chapter, we abolish this requirement. Additionally, we no longer require TBoxes to be acyclic and to contain concept definitions, only.

The syntax and semantics of the logic $\mathcal{EL}^{(n)}$ were defined in Chapter 2. Now we define the notions of a general concept inclusion axiom and a general TBox.

A *general concept inclusion*(GCI) axiom is an axiom of the form $C \sqsubseteq D$, where both C and D are $\mathcal{EL}^{(n)}$-concept terms. A *general TBox* is a finite set of GCIs. General TBoxes can also express concept definitions of the form $A \equiv C$ with the help of two GCIs: $A \sqsubseteq C$ and $C \sqsubseteq A$.

An interpretation $\mathcal{I}$ is a *model* of a general TBox $\mathcal{T}$ if for every $C \sqsubseteq D \in \mathcal{T}$, we have that $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$. As before, we say that a concept term C_1 is subsumed by a concept term C_2 with respect to a general TBox $\mathcal{T}$, written $C_1 \sqsubseteq_{\mathcal{T}} C_2$, if $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$, for every model $\mathcal{I}$ of $\mathcal{T}$.

In the following, we show that allowing for unrestricted $\mathcal{EL}^{(n)}$-concept terms and GCIs dramatically effects the complexity of the subsumption problem. Namely, subsumption becomes ExpTime-complete.

As usual, we prove ExpTime-completeness of subsumption in $\mathcal{EL}^{(n)}$ with general TBoxes in two phases. Firstly, we show that subsumption is ExpTime-hard, and secondly, that it is in ExpTime.

Theorem 10 (ExpTime-hardness) *Subsumption in $\mathcal{EL}^{(n)}$ is* ExpTime*-hard in the presence of general TBoxes.*

Proof The result follows directly from the fact that the logic $\mathcal{EL}^{\geq 2}$, which is the extension of $\mathcal{EL}$ with at-least restrictions of the form $(\geq 2\ r)$, is ExpTime-complete in the presence of general TBoxes. The logic $\mathcal{EL}^{\geq 2}$ is a fragment of $\mathcal{EL}^{(n)}$, since the constructors $\top$ and $\sqcap$ are present in $\mathcal{EL}^{(n)}$ directly, and the remaining constructors of $\mathcal{EL}^{\geq 2}$ can be linearly translated to $\mathcal{EL}^{(n)}$ ones as follows:

- $\exists r.C \equiv \exists r.(C)$;
- $(\geq 2\ r) \equiv \exists r.(\top, \top)$.

The proof of the ExpTime-completeness of subsumption in $\mathcal{EL}^{\geq 2}$ can be found in [BBL05].

□

Now, we prove that subsumption in $\mathcal{EL}^{(n)}$ with general TBoxes is in ExpTime by showing that subsumption in $\mathcal{ALC}^{(n)}$, the extension of $\mathcal{EL}^{(n)}$ with the complement operator $\neg$, is in ExpTime in the presence of general TBoxes. We introduce syntax and semantics of $\mathcal{ALC}^{(n)}$ explicitly.

As before, let N_c and N_r be disjoint sets of concept and role names, respectively. The set of $\mathcal{ALC}^{(n)}$*-concept terms* is defined inductively as follows:

- $\top$ is an $\mathcal{ALC}^{(n)}$-concept term;
- A is an $\mathcal{ALC}^{(n)}$-concept term, for every $A \in N_c$;
- if $C, D, C_1, \ldots, C_n$ are $\mathcal{ALC}^{(n)}$-concept terms, for some $n > 0$, and r is a role name, then the following are $\mathcal{ALC}^{(n)}$-concept terms: $\neg C$, $C \sqcap D$, $\exists r.(C_1, \ldots, C_n)$.

Semantics of the constructors $\top$, $\sqcap$ and $\exists r.(C_1, \ldots, C_n)$ has already been given in Chapter 2. The complement constructor $\neg$ is interpreted as follows. Let $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ be an interpretation. Then, for any $\mathcal{ALC}^{(n)}$-concept term C, $(\neg C)^{\mathcal{I}} := \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$.

Note that the following constructors can be expressed in $\mathcal{ALC}^{(n)}$, although we do not introduce them explicitly:

$$\begin{aligned} \bot &= \neg\top \\ C \sqcup D &= \neg\big(\neg C \sqcap \neg D\big) \\ \forall r.C &= \neg\big(\exists r.(C)\big) \end{aligned}$$

In the following, we use the notion of *satisfiability* of a concept term which is formally defined as follows: an $\mathcal{ALC}^{(n)}$-concept term C is satisfiable (w.r.t. a general TBox $\mathcal{T}$), if there is an interpretation $\mathcal{I}$ (which is a model of $\mathcal{T}$) such that $C^{\mathcal{I}} \neq \emptyset$.

In order to show that subsumption in $\mathcal{ALC}^{(n)}$ is in ExpTime, we first reduce it to satisfiability: for arbitrary $\mathcal{ALC}^{(n)}$-concept terms C, D and a TBox $\mathcal{T}$, the following are equivalent:

- $C \sqsubseteq_{\mathcal{T}} D$
- $C \sqcap \neg D$ is unsatisfiable w.r.t. $\mathcal{T}$.

We note that this reduction is valid in any logic that is closed under the constructors $\sqcap$ and $\neg$.

We show now that satisfiability (and hence, subsumption) in $\mathcal{ALC}^{(n)}$ is in ExpTime. For this, we use a technique similar to the *elimination of Hintikka sets* approach that is used in [BdRV01] to prove that satisfiability in propositional dynamic logic is in ExpTime.

We have to introduce some notions. Let C_0 be an $\mathcal{ALC}^{(n)}$-concept term and $\mathcal{T}$ a general TBox such that it is to be decided whether C_0 is satisfiable w.r.t. $\mathcal{T}$. We use the abbreviations sub and cl that are defined, for any $\mathcal{EL}^{(n)}$-concept term C and any TBox $\mathcal{T}$, as follows:

- $\mathsf{sub}(C) := \{C' | C' \text{ is a subconcept of } C\}$;
- $\mathsf{sub}(\mathcal{T}) := \bigcup_{D \sqsubseteq E \in \mathcal{T}} \big(\mathsf{sub}(D) \cup \mathsf{sub}(E)\big)$;
- $\mathsf{sub}(C, \mathcal{T}) := \mathsf{sub}(C) \cup \mathsf{sub}(\mathcal{T})$, and
- $\mathsf{cl}(C, \mathcal{T}) := \mathsf{sub}(C, \mathcal{T}) \cup \{\neg D | D \in \mathsf{sub}(C, \mathcal{T})\}$.

We use $\mathsf{cl}(C)$ as an abbreviation for $\mathsf{cl}(C, \emptyset)$. Now, we introduce the notion of *type* which plays an important role in the rest of this chapter.

Definition 13 (Type) *A set $\Psi \subseteq \mathsf{cl}(C_0, \mathcal{T})$ is a* type *for $C_0, \mathcal{T}$ if the following conditions are satisfied:*

(t1) for all $\neg C \in \mathsf{sub}(C_0, \mathcal{T})$, $\neg C \in \Psi$ iff $C \notin \Psi$;

(t2) for all $C \sqcap D \in \mathsf{sub}(C_0, \mathcal{T})$, $C \sqcap D \in \Psi$ iff $\{C, D\} \subseteq \Psi$;

(t3) for all $\neg(C \sqcap D) \in \mathsf{sub}(C_0, \mathcal{T})$, $\neg(C \sqcap D) \in \Psi$ iff $\{C, D\} \cap \Psi \neq \emptyset$;

(t4) for all $D \sqsubseteq E \in \mathcal{T}$, $D \in \Psi$ implies $E \in \Psi$.

Let Ψ be a type for $C_0, \mathcal{T}$, $\Phi_0, \ldots, \Phi_{n-1}$ a (possibly empty) sequence of types for $C_0, \mathcal{T}$, and r a role name. Then $\Phi_0, \ldots, \Phi_{n-1}$ is a *successor candidate* for Ψ w.r.t. r if for all $\exists r.(C_1, \ldots, C_k) \in \mathsf{cl}(C_0, \mathcal{T})$, we have $\exists r.(C_1, \ldots, C_k) \in \Psi$ iff there are $i_1, \ldots, i_k < n$ such that $C_j \in \Phi_{i_j}$ for all $j = 1, \ldots, k$ and $i_j \neq i_\ell$ for all j, ℓ, $1 \leq j < \ell \leq k$.
For a set of concept terms Γ, we define

$$\mathsf{rol}_\exists(\Gamma) := \{r \in N_r | \exists r.(C_1, \ldots, C_n) \in \mathsf{cl}(C_0, \mathcal{T}), \text{ for some } C_1, \ldots, C_n\}$$

and for every $r \in \mathsf{rol}_\exists(\Gamma)$,

$$N_r(\Gamma) := \sum_{\exists r.(C_1, \ldots, C_k) \in \Gamma} k$$

The following Lemma shows tractability of deciding whether a sequence $\Phi_0, \ldots, \Phi_{n-1}$ of sets of concepts is a successor candidate for Ψ w.r.t. r and Γ.

Lemma 11 *Let $\Psi, \Phi_0, \ldots, \Phi_{n-1}$ subsets of $\mathsf{cl}(C_0, \mathcal{T})$. It is decidable in polynomial time whether $\Phi_0, \ldots, \Phi_{n-1}$ is a successor candidate for Ψ w.r.t. r.*

Proof Firstly, we define an additional notion. A *system of distinct representatives (SDR)* for a family of sets $S_1, \ldots, S_k$ is a k-tuple $(a_1, \ldots, a_k)$ such that $a_i \in S_i$ for $i = 1, \ldots, k$ and all a_i are distinct, i.e., $a_i \neq a_j$, for all i, j, $1 \leq i < j \leq k$.

It is enough to show that, for each $\exists r.(C_1, \ldots, C_k) \in \mathsf{cl}(C_0, \mathcal{T})$, it is decidable in polynomial time whether there are indices $i_1, \ldots, i_k < n$ such that $C_j \in \Phi_{i_j}$ for $1 \leq j \leq k$ and $i_j \neq i_l$ for $1 \leq j < l \leq k$.
For each $j = 1, \ldots, k$, we define the set

$$S_j := \{i | 0 \leq i < n \text{ and } C_j \in \Phi_i\}.$$

Then there are indices $i_1, \ldots, j_k < n$ as required iff $(S_1, \ldots, S_k)$ has an SDR. The existence of an SDR can be decided in polynomial time by a reduction to the maximum bipartite matching problem, which is known to be polynomial [HK73].
□

A type Γ is called *bad* w.r.t. a set of types $\mathfrak{T}$ if there exists a role name $r \in \mathsf{rol}_\exists(\Gamma)$ such that there is no sequence $\Phi_0, \ldots, \Phi_{n-1} \in \mathfrak{T}$ with $n \leq N_r(\Gamma)$ that is a successor candidate for Γ w.r.t. r.

Figure 4.1 presents a procedure $\mathcal{ALC}^{(n)}$-Elim that decides satisfiability of an $\mathcal{ALC}^{(n)}$-concept term C_0 w.r.t. a TBox $\mathcal{T}$. The following proposition uses the procedure $\mathcal{ALC}^{(n)}$-Elim to show the exponential upper bound for satisfiability in $\mathcal{ALC}^{(n)}$.

Proposition 1 *The procedure $\mathcal{ALC}^{(n)}$-Elim introduced in Fig. 4.1 decides satisfiability of C_0 w.r.t. $\mathcal{T}$ in exponential time.*

define procedure $\mathcal{ALC}^{(n)}$-$\mathsf{Elim}(C_0, \mathcal{T})$
Set $i := 0$ and $\mathfrak{T}_0$ to the set of all types for C and $\mathcal{T}$
repeat
$\mathfrak{T}_{i+1} := \{\Gamma \in \mathfrak{T}_i \mid \Gamma \text{ is not bad w.r.t. } \mathfrak{T}_i\}$
$i := i + 1$
while $\mathfrak{T}_i \neq \mathfrak{T}_{i-1}$
if there exists a type $\Gamma \in \mathfrak{T}_i$ with $C \in \Gamma$ **then**
return true
return false

Figure 4.1: Procedure $\mathcal{ALC}^{(n)}$-$\mathsf{Elim}(C_0, \mathcal{T})$

Proof First, we show that the procedure $\mathcal{ALC}^{(n)}$-Elim defined in Fig. 4.1 terminates after at most exponentially many steps. The **repeat** loop is executed at most exponentially many times since there are exponentially many types and, in each iteration at least one type is eliminated. Checking whether a type is bad can be done in exponential time since there are at most exponentially many sequences of types of length at most $N_r\big(\mathsf{cl}(C_0, \mathcal{T})\big)$. By Lemma 11, for each such sequence, it can be checked in polynomial time whether it is a successor candidate. Thus, $\mathcal{ALC}^{(n)}$-Elim is a deterministic exponential time procedure.

We show now that $\mathcal{ALC}^{(n)}$-$\mathsf{Elim}(C_0, \mathcal{T})$ answers true iff C_0 is satisfiable w.r.t. $\mathcal{T}$. Assume that $\mathcal{ALC}^{(n)}$-Elim terminates returning true. We construct an interpretation $\mathcal{I}$ such that $C_0^{\mathcal{I}} \neq \emptyset$. Let $\mathfrak{T}$ be the set of types that have not been eliminated. We denote with $\Gamma_{C_0} \in \mathfrak{T}$ a type with $C_0 \in \Gamma_{C_0}$. Let $\Gamma \in \mathfrak{T}$ and $r \in \mathsf{rol}_\exists(\Gamma)$. Since Γ was not eliminated, it has a successor candidate $\Psi_0, \ldots, \Psi_{n-1}$ with all the $\Psi_i \in \mathfrak{T}$. These types, however, need not to be all distinct. For this reason, it is not enough to take just the types in $\mathfrak{T}$ as the domain elements of $\mathcal{I}$. To have enough copies of each type available, we define

$$N := max\big\{N_r\big(\mathsf{cl}(C_0, \mathcal{T})\big) | r \in \mathsf{rol}_\exists\big(\mathsf{cl}(C_0, \mathcal{T})\big)\big\},$$

and generate N copies of each type in $\mathfrak{T}$. Now, we define the interpretation $\mathcal{I}$ as follows:

- $\Delta^{\mathcal{I}} := \{(\Gamma, i) | 1 \leq i \leq N \text{ and } \Gamma \in \mathfrak{T}\}$.
- $A^{\mathcal{I}} := \{(\Gamma, i) \in \Delta^{\mathcal{I}} | A \in \Gamma\}$, for all concept names A.
- Let $(\Gamma, i) \in \Delta^{\mathcal{I}}$ and $r \in \mathsf{rol}_\exists(\Gamma)$. Since Γ was not eliminated, there exists a successor candidate $\Psi_1, \ldots, \Psi_n$ for Γ w.r.t. r. By the definition of $N_r(\Gamma)$, we know that $\sum_{i=1}^{m} n_j \leq N_r(\Gamma) \leq N$. Thus, we can define the set

$$\{(\Psi_i, i) | 1 \leq i \leq n\}$$

to be the set of r-successors of Γ in $\mathcal{I}$.

We prove now by structural induction on C the fact that, for all $(\Gamma, i) \in \Delta^{\mathcal{I}}$ and all $C \in \mathsf{cl}(C, \mathcal{T})$, we have $(\Gamma, i) \in C^{\mathcal{I}}$ iff $C \in \Gamma$.

Base case. Let C be a concept name. Then the fact that $(\Gamma, i) \in C^{\mathcal{I}}$ iff $C \in \Gamma$ follows directly from the above definition of $\mathcal{I}$.

Induction step. Let $C = \neg C'$. Then $(\Gamma, i) \in C^{\mathcal{I}}$ iff $(\Gamma, i) \notin C'^{\mathcal{I}}$ iff (by I.H.) $C' \notin \Gamma$ which, by definition of a type, is equivalent to $\neg C' \in \Gamma$. Thus, $(\Gamma, i) \in C^{\mathcal{I}}$ iff $C \in \Gamma$.
Let $C = C' \sqcap D'$. Then $(\Gamma, i) \in C^{\mathcal{I}}$ iff $(\Gamma, i) \in C'^{\mathcal{I}} \wedge (\Gamma, i) \in D'^{\mathcal{I}}$ iff (by I.H.) $C' \in \Gamma \wedge D' \in \Gamma$ iff $C' \sqcap D' \in \Gamma$.
Let $C = \exists r.(C_1, \ldots, C_n)$, for some $r \in N_r$. Then since Γ has a successor candidate w.r.t. r, we know that there exist $\Psi_1, \ldots, \Psi_n \in \mathfrak{T}$ with $C_k \in \Psi_k$, for $k = 1, \ldots, n$. By I.H., the latter is equivalent to $(\Psi_k, j_k) \in C_k^{\mathcal{I}}$, for $k = 1, \ldots, n$ and some j_k. Again by definition of $\mathcal{I}$ we know that all the pairs (Ψ_k, j_k) are r-successors of (Γ, i) and thus $(\Gamma, i) \in \exists r.(C_1, \ldots, C_n)^{\mathcal{I}} = C^{\mathcal{I}}$.

Now we have that $(\Gamma_{C_0}, 1) \in C_0^{\mathcal{I}}$. In addition, if $D \sqsubseteq E \in \mathcal{T}$ and $(\Gamma, i) \in D^{\mathcal{I}}$, then $D \in \Gamma$, and thus, by Condition *(t4)* of the definition of type, $E \in \Gamma$, which implies $(\Gamma, i) \in E^{\mathcal{I}}$. Thus, we have constructed the required interpretation $\mathcal{I}$ with $C^{\mathcal{I}} \neq \emptyset$.

Conversely, assume that C_0 is satisfiable w.r.t. $\mathcal{T}$, and let $\mathcal{I}$ be the model of $\mathcal{T}$ such that $x_0 \in C_0^{\mathcal{I}}$, for some $x_0 \in \Delta^{\mathcal{I}}$. For $x \in \Delta^{\mathcal{I}}$, we define

$$\mathsf{tp}(x) := \left\{C \in \mathsf{cl}(C, \mathcal{T}) | x \in C^{\mathcal{I}}\right\}.$$

Similarly to the inductive proof above, it is easy to prove by structural induction on C_0 that no type in $\mathfrak{T} := \{\mathsf{tp}(x) | x \in \Delta^{\mathcal{I}}\}$ is eliminated by $\mathcal{ALC}^{(n)}$-$\mathsf{Elim}(C_0, \mathcal{T})$. Since $\mathsf{tp}(x_0)$ contains C_0, $\mathcal{ALC}^{(n)}$-Elim returns true.

□

Summing up, we can conclude that satisfiability and, hence, subsumption in $\mathcal{ALC}^{(n)}$ w.r.t. general TBoxes is in ExpTime. This upper bound translates to $\mathcal{EL}^{(n)}$, which is a fragment of $\mathcal{ALC}^{(n)}$. The next theorem states that the ExpTime lower bound for subsumption in $\mathcal{EL}^{(n)}$ with general TBoxes is indeed optimal.

Theorem 12 *The subsumption problem in $\mathcal{EL}^{(n)}$ and $\mathcal{ALC}^{(n)}$ with respect to general TBoxes is* ExpTime*-complete.*

Chapter 5

Experimental evaluation

Since in the process engineering application, that motivated our work, the reasoning is performed in restricted $\mathcal{EL}^{(n)}$ with restricted TBoxes, we have implemented the polynomial algorithm for deciding subsumption that is developed in Chapter 3. The resulting system is referred to as Eln. In addition, we compared the performance of Eln with the state-of-the-art DL reasoner Racer [HM01].

5.1 Implementation

Eln is a straightforward C-implementation of the polynomial algorithm for deciding subsumption in $\mathcal{EL}^{(n)}$ with acyclic TBoxes that was developed in Chapter 3. It accepts the input data in the XML-based format DIG-1.0 [Bec02] recommended by the Description Logics Implementation Group (DIG). Usually, a DIG-1.0 task description consists of two parts: Tells and Asks. The Tells part contains TBox axioms like, e.g.,

```
<equalc>
  <catom name="C"/>
  <and>
    <catom name="D"/>
    <catom name="E"/>
  </and>
</equalc>
```

which encodes the axiom $C \equiv D \sqcap E$. The Asks part enumerates queries like the following ones:

```
<satisfiable>
  <catom name="C"/>
</satisfiable>
```

```
<subsumes>
  <catom name="C"/>
  <catom name="D"/>
</subsumes>
```

that should be read as follows: *is C satisfiable?* and *does $D \sqsubseteq C$ hold?*, respectively. Note that since any $\mathcal{EL}^{(n)}$-concept term is trivially satisfiable, we concentrate on subsumption tests only. All the queries in the Asks part are considered with respect to the TBox defined in the Tells part. We have taken the $\mathcal{EL}$-relevant fragment of DIG-1.0 and extended it with the n-ary existential quantifier such that the concept term $\exists r.(C, D, E)$ is encoded as follows:

```
<someN>
  <ratom name="r"/>
  <catom name="C"/>
  <catom name="D"/>
  <catom name="E"/>
</someN>
```

Eln uses the library Libxml to parse the input data and to perform all operations on XML-trees. After reading the input file, Eln performs the following manipulations:

- Syntax checking: Any syntactic element that is not a valid $\mathcal{EL}^{(n)}$-construction is removed with printing out the corresponding warning message;
- Acyclicity test: If the input TBox $\mathcal{T}$ contains a set of cyclic definitions, the program terminates with the respective error message;
- Graph construction: An $\mathcal{EL}^{(n)}$-description forest $\mathcal{F}_\mathcal{T}$ is constructed for $\mathcal{T}$;
- Queries execution: For every subsumption test $C \sqsubseteq D$ encoded in the Asks part, check the existence of a subsumption mapping from D to C in $\mathcal{F}_\mathcal{T}$.

In order to solve the maximum bipartite matching problem, Eln employs the LEDA library that implements highly optimized graph algorithms [MN99].

5.2 Experimental data and environment

In our experiments, we use two sources of input data. The first one is a family of artificially created TBoxes $\mathcal{T}_n$, for $n > 0$, that are of the following form:

$$\mathcal{T}_n := \{ \begin{array}{lcl} C & \equiv & \exists r.(C_1, \ldots, C_n) \\ D & \equiv & \exists r.(A_1, \ldots, A_n) \\ C_1 & \equiv & A_1 \sqcap B_1 \\ \vdots & & \\ C_n & \equiv & A_n \sqcap B_n \} \end{array}$$

It is easy to see that the ontologies $\mathcal{T}_n$, $n \geq 0$ are restricted $\mathcal{EL}^{(n)}$-TBoxes. Note, that for each $n > 0$, $C \sqsubseteq_{\mathcal{T}_n} D$ and $D \not\sqsubseteq_{\mathcal{T}_n} C$, i.e., C is subsumed by D w.r.t. $\mathcal{T}_n$ but not vice versa.

The second type of input data is a real world ontology from the chemical process engineering that is referred to as PEN (Process ENgineering). The PEN ontology is a restricted $\mathcal{EL}^{(n)}$-TBox that contains 109 concept definitions. About 30 of them use the n-ary existential quantifier $\exists r.(C_1, \ldots, C_n)$ with $n \in \{2, 3, 4, 5, 31\}$. The comparison between Eln and Racer on the PEN ontology is performed by means of a testing procedure that generates 500 random subsumption tests between concept names defined in these 30 definitions.

As for $\mathcal{T}_n$-TBoxes, both reasoners were to answer the queries $C \sqsubseteq^?_{\mathcal{T}_n} D$ and $D \sqsubseteq^?_{\mathcal{T}_n} C$ with the growing value of n.

All experiments presented in this chapter were carried out on a Pentium IV machine with 2.7GHz CPU and 3GB of RAM. The runtime for every subsumption test was limited by 600 seconds. A task was considered to be solved if a reasoner could provide the correct answer within the given time interval. The experimental results themselves are presented in Chapter 5.4.

5.3 Translation from $\mathcal{EL}^{(n)}$ to $\mathcal{ALCQ}$

In fact, the n-ary existential quantifier introduced in this work is not explicitly present in the syntax of existing DLs. Therefore, in order to run Racer on knowledge bases containing this new constructor, the latter should be translated into some DL supported by Racer.

In our experiments, we used the translation procedure proposed in [TvW04]. The idea of this translation is based on the fact that semantics of an $\mathcal{EL}^{(n)}$-concept term $\exists r.(C_1, \ldots, C_n)$ is closely related with the notion of a system of distinct representatives (SDR) introduced in Section 4. We remind here that an SDR for a family of sets $S_1, \ldots, S_k$ is a k-tuple $(a_1, \ldots, a_k)$ such that $a_i \in S_i$ for $i = 1, \ldots, k$ and all a_i are distinct, i.e., $a_i \neq a_j$, for all i, j, $1 \leq i < j \leq k$.

Indeed, let us consider an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ and some domain individual $a \in \Delta^{\mathcal{I}}$. We denote the set of all r-successors of a as $S_r(a)$. The following is a consequence of the semantics of $\exists r.(C_1, \ldots, C_n)$: $a \in \big(\exists r.(C_1, \ldots, C_n)\big)^{\mathcal{I}}$ iff there exists an SDR for $\big(S_r(a) \cap C_1^{\mathcal{I}}, \ldots, S_r(a) \cap C_n^{\mathcal{I}}\big)$. Hall's theorem [Hal35] gives necessary and sufficient condition of existence of an SDR for a family of sets.

n	$C \sqsubseteq^?_{\mathcal{T}_n} D$ (yes)		$D \sqsubseteq^?_{\mathcal{T}_n} C$ (no)	
	Racer time, sec.	Eln time, sec	Racer time, sec.	Eln time, sec
1	0.10	0.10	0.11	0.10
2	0.24	0.10	0.36	0.10
3	0.88	0.11	0.59	0.10
4	> 600	0.12	0.81	0.10
5	-	0.12	1	0.11
6	-	0.14	27	0.11
7	-	0.14	> 600	0.12
8	-	0.14	-	0.12
9	-	0.15	-	0.12
10	-	0.15	-	0.13

Figure 5.1: Runtime of Racer and Eln on subsumption tests w.r.t. ontologies $\mathcal{T}_n$.

Theorem 13 (Hall's theorem) *A family of sets $S_1, \dots, S_n$ has an SDR iff for any $i_1, \dots, i_k \in \{1, \dots, n\}$, $1 \leq k \leq n$, the following holds:*

$$|S_{i_1} \cup \ldots \cup S_{i_k}| \geq k.$$

The condition stated in Hall's theorem is expressible in $\mathcal{ALCQ}$, which is accepted by Racer. Thus, we obtain the equivalence preserving translation from $\mathcal{EL}^{(n)}$ to $\mathcal{ALCQ}$ proposed in [TvW04]:

$$\exists r.(C_1, \dots, C_n) \equiv \prod_{\mathcal{M} \subseteq \{1,\dots,n\}} \Big(\geq |\mathcal{M}| r. \big(\bigsqcup_{j \in \mathcal{M}} C_j \big) \Big).$$

This translation shows that augmenting the logic $\mathcal{ALCQ}$ with the n-ary existential restriction constructor does not extend its expressive power. But on the other hand, this translation is exponential in the size of the input and has an PSPACE-hard target logic. Thus, one can suppose that translation-based reasoning algorithms for $\mathcal{EL}^{(n)}$ do not scale well in comparison to direct ones. We present evidence of this fact in the next section.

5.4 Experimental results

We start with evaluating both reasoners Racer and Eln on the artificially created ontologies $\mathcal{T}_n$ described in Section 5.2. The timing results for these tests are depicted in Figure 5.1. We note that even for small values of n, namely $n = 1, \dots, 10$, Eln demonstrates advantageous behaviour in comparison to Racer. For small n, Eln finds the subsumption relationship immediately, i.e., with no measurable runtime. Whereas, Racer does not scale up to the problems with n

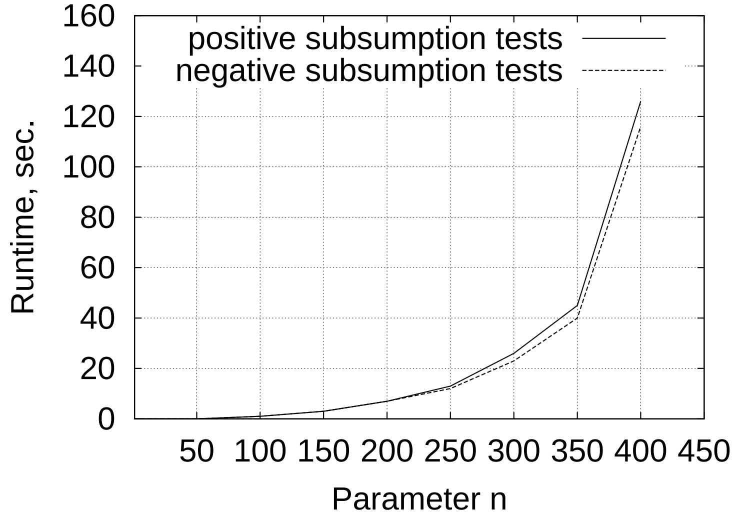

Figure 5.2: Runtime of Eln on ontologies $\mathcal{T}_n$ with the growth of n.

larger than 7. The latter observations are true for both kinds of subsumption tests. Namely, for positive ones of the kind $C \sqsubseteq^?_{\mathcal{T}_n} D$, where the subsumption between C and D holds, and for negative ones of the kind $D \sqsubseteq^?_{\mathcal{T}_n} C$, where the subsumption does not hold.

In order to analyze the computational behaviour of Eln on larger problems, we have executed the same subsumption tests, i.e., $C \sqsubseteq^?_{\mathcal{T}_n} D$ and $D \sqsubseteq^?_{\mathcal{T}_n} C$, with $n = 50, 100, \ldots, 500$. The results of these experiments are shown in Figure 5.2.

One should observe that even for $n = 100$ the runtime of our unoptimized implementation is just 1 second. We also note that Eln requires additional optimizations in order to scale up to problems of the size larger than n=400. We believe that optimized search procedures using, e.g., hash tables, or caching techniques would bring a substantial gain to our yet straightforward implementation. Moreover, we have realized that our choice to use XML trees for storing $\mathcal{EL}^{(n)}$-description graphs implies an unacceptably high memory consumption. In the future, it would be necessary to use more efficient data structures.

In addition to artificially created ontologies $\mathcal{T}_n$, we compare the computational behaviour of both reasoners on the real world ontology PEN that was presented in Section 5.2. As mentioned above, this experiment consisted of 500 random subsumption tests between PEN concept names that were defined using the n-ary existential quantifier. As the result, Eln could solve all of the 500 subsumption tests taking not more than 1 second per single test. Whereas Racer could not solve any test due to an extremely large size of the ontology after its translation into the DL $\mathcal{ALCQ}$.

Chapter 6

Conclusion

Motivated by the chemical process engineering application, we have extended the description logic $\mathcal{EL}$ by the n-ary existential quantifier. This constructor generalizes both the standard existential quantifier ($\exists r.C$) and the qualified number restrictions ($\geq nr.C$). The resulting logic is referred to as $\mathcal{EL}^{(n)}$. A fragment of $\mathcal{EL}^{(n)}$, referred to as *restricted* $\mathcal{EL}^{(n)}$ allows to formalize process engineering terminologies in a natural and concise way.

We have investigated the complexity of reasoning in restricted $\mathcal{EL}^{(n)}$. In particular, we have shown that the following instances of the subsumption problem are of polynomial complexity: Subsumption between concept terms (Chapter 2) and subsumption with respect to acyclic restricted TBoxes (Chapter 3).

Furthermore, we have justified that subsumption with respect to general TBoxes, i.e., TBoxes containing general concept inclusion axioms, is ExpTime-complete in $\mathcal{EL}^{(n)}$ as well as in $\mathcal{ALC}^{(n)}$, which is the extension of $\mathcal{EL}^{(n)}$ with the complement operator (Chapter 4).

We note that in the process engineering application, it is sufficient to restrict the subsumption problem to the case of restricted $\mathcal{EL}^{(n)}$-TBoxes. Therefore, we have implemented the polynomial algorithm for solving subsumption in restricted $\mathcal{EL}^{(n)}$ with respect to restricted TBoxes in a system Eln. Since the state-of-the-art DL reasoners like, e.g., Racer, do not allow for the n-ary existential quantifier, the latter should be translated using constructors available in existing DLs. In order to compare the performance of Eln with Racer, we have relied on the translation procedure that was recently developed in [TvW04]. This procedure demonstrates how the n-ary existential quantifier can be expressed in the DL $\mathcal{ALCQ}$. The comparison analysis between Eln and Racer has provided the first justification of the fact that the direct treatment of the n-ary existential quantifier leads to dramatic computational improvements. The latter result is valid for artificially constructed ontologies as well as for a real world ones from process engineering application.

One promising direction for future work could be investigating the complexity of reasoning in restricted $\mathcal{EL}^{(n)}$ in the presence of cyclic terminologies w.r.t. greatest fixpoint and descriptive semantics. For this we would expect to use the simulation based approach that was recently introduced in [Baa03] in order to analyze the complexity of the similar problem in $\mathcal{EL}$. It would be also challenging to check whether the tractability results presented in this work can be translated to the case of reasoning with ABoxes. It would also be very interesting, to extend tableaux algorithms for expressive DLs with the ability to treat the n-ary existential quantifier.

Bibliography

[Baa03] Franz Baader. Terminological cycles in a description logic with existential restrictions. In Georg Gottlob and Toby Walsh, editors, *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pages 325–330. Morgan Kaufmann, 2003.

[BBL05] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ envelope. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence IJCAI-05*, Edunburgh, UK, 2005. Morgan-Kaufmann Publishers.

[BCM+03] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[BdRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Compute Science*. Cambridge University Press, 2001.

[Bec02] Sean Bechhofer. *The DIG Description Logic Interface: DIG/1.0*. University of Manchester, Oxford Road Manchester M13 9PL, October 2002.

[BH93] F. Baader and P. Hanschke. Extensions of concept languages for a mechanical engineering application. In *Proceedings of the 16th German AI-Conference, GWAI-92*, volume 671 of *Lecture Notes in Computer Science*, pages 132–143, Bonn (Germany), 1993. Springer–Verlag.

[BKM99] 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.

[BL84] Ronald J. Brachman and Hector J. Levesque. The tractability of subsumption in frame-based description languages. In *Proceedings of the Fourth National Conference on Artificial Intelligence AAAI-84*, pages 34–37, 1984.

[Bra79] Ronald J. Brachman. On the epistemological status of semantic networks. In Nicholas V. Findler, editor, *Associative networks*, pages 3–50. Academic Press, 1979.

[Bra04] Sebastian Brandt. On ubusmption and instance problem in $\mathcal{ELH}$ w.r.t. general tboxes. In *Proceedings of the 2004 International Workshop on Description Logics(DL2004)*, CEUR-WS, 2004.

[CLN94] Diego Calvanese, Maurizio Lenzerini, and Daniele Nardi. A unified framework for class based representation formalisms. In J. Doyle, E. Sandewall, and P.Torasso, editors, *Proceedings of the fourth International COnference on the Principles of Knowledge Representation and Reasoning (KR'94)*, pages 109–120. Morgan Kaufmann, 1994.

[Hal35] Philip Hall. On representatives of subsets. *The Journal of the London Mathematical Society*, 10:26–30, 1935.

[HB91] B. Hollunder and F. Baader. Qualifying number restrictions in concept languages. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning,KR-91*, pages 335–346, Boston (USA), 1991.

[HK73] John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973.

[HM01] Volker Haarslev and Ralf Möller. RACER system description. In Rajeev Goré, Alexander Leitsch, and Tobias 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.

[Hor98] Ian 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.

[HST00] Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Practical reasoning for very expressive Description Logics. *Journal of the Interest Group in Pure and Applied Logic*, 8(3):239–264, 2000.

[Mar94] Wolfgang Marquardt. Modelbildung als grundlage der prozesssimulation. In H. Schuler, editor, *Prozesssimulation*. Verlag Chemie Weinheim, 1994.

[Min81] Marvin Minsky. A framework fo representing knowledge. In J. Haugeland, editor, *Mind Design*. The MIT Press, 1981.

[MN99] K. Mehlhorn and S. Näher. *The LEDA platform of combinatorial and geometric computing*. Cambridge University Press, 1999.

[RH97] Alan Rector and Ian Horrocks. Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions. In *Proc. of the Workshop on Ontological Engineering*, 1997.

[RNG93] A. L. Rector, W. A. Nowlan, and A. Glowinski. Goals for concept representation in the Galen project. In *17th Annual Symposium on Computer Applications in Medical Care (SCAMC '93)*, pages 414–418, Washington DC, USA, 1993.

[Sat96] Ulrike Sattler. A concept language extended with different kinds of transitive roles. In G. Görz and S. Hölldobler, editors, *20. Deutsche Jahrestagung für Künstliche Intelligenz*, number 1137 in Lecture Notes in Artificial Intelligence. Springer Verlag, 1996.

[Spa00] K.A. Spackman. Managing clinical terminology hierarchies using algorithmic calculation of subsumption: Experience with SNOMED-RT. *Journal of the American Medical Information Association*, 2000.

[SSS91] M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48(1):1–26, 1991.

[TvW04] Manfred Theißen and Lars von Wedel. The need for an n-ary existential quantor in description logics. In Sean Bechhofer, Volker Haarslev, Carsten Lutz, and Ralf Moeller, editors, *Proceedings of the KI-04 Workshop on Applications of Description Logics*, 2004.

[vW04] Lars von Wedel. *An environment for heterogeneous management in chemical process engineering*. PhD thesis, Lehrstuhl fuer Prozesstechnik, RWTH Aachen University, 2004.

Statement of academic honesty

I hereby declare that this thesis was written by me and I have not used any auxiliary sources for the present work other than have been cited in my thesis.

Hiermit versichere ich, dass die vorliegende Diplomarbeit von mir selbständig verfaßt wurde und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt wurden.

Dresden, 22nd August 2005 Eldar Karabaev