TECHNISCHE UNIVERSITÄT DRESDEN

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

LTCS–Report

A New n-ary Existential Quantifier in Description Logics

Franz Baader, Carsten Lutz, Eldar Karabaev, and Manfred Theißen

LTCS-Report 05-08

Short versions of this report have also appeared in
Proc. of KI'05 and Proc. of DL'05.

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

Hans-Grundig-Str. 25
01062 Dresden
Germany

A New n-ary Existential Quantifier in Description Logics

Franz Baader, Carsten Lutz, Eldar Karabaev
Institut für Theoretische Informatik
TU Dresden
`{last_name}}@tcs.inf.tu-dresden.de`

Manfred Theißen
Lehrstuhl für Prozesstechnik
RWTH Aachen
`theissen@lpt.rwth-aachen.de`

Abstract

Motivated by a chemical process engineering application, we introduce a new concept constructor in Description Logics (DLs), an n-ary variant of the existential restriction constructor, which generalizes both the usual existential restrictions and so-called qualified number restrictions. We show that the new constructor can be expressed in $\mathcal{ALCQ}$, the extension of the basic DL $\mathcal{ALC}$ by qualified number restrictions. However, this representation results in an exponential blow-up. By giving direct algorithms for $\mathcal{ALC}$ extended with the new constructor, we can show that the complexity of reasoning in this new DL is actually not harder than the one of reasoning in $\mathcal{ALCQ}$. Moreover, in our chemical process engineering application, a restricted DL that provides only the new constructor together with conjunction, and satisfies an additional restriction on the occurrence of roles names, is sufficient. For this DL, the subsumption problem is polynomial.

Contents

1 Introduction

Description Logics (DLs) [3] are a class of knowledge representation formalisms in the tradition of semantic networks and frames, which can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way. DL systems provide their users with inference services (like computing the subsumption hierarchy) that deduce implicit knowledge from the explicitly represented knowledge. For these inference services to be feasible, the underlying inference problems must at least be decidable, and preferably of low complexity. This is only possible if the expressiveness of the DL employed by the system is restricted in an appropriate way. Because of this restriction of the expressive power of DLs, various application-driven language extensions have been proposed in the literature (see, e.g., [4, 10, 23, 17]), some of which have been integrated into state-of-the-art DL systems [16, 14].

The present paper considers a new concept constructor that is motivated by a process engineering application [24]. This constructor is an n-ary variant of the usual existential restriction operator available in most DLs. To motivate the need for this new constructor, 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 the 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 with the help of the usual existential restriction operator as

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

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 does *not* capture the intended meaning of a plant having *two* reactors, one with a 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 individuals having n *different* r-successors $d_1, \ldots, d_n$ such that d_i belongs to C_i ($i = 1, \ldots, n$). Given this constructor, our concept can correctly be described as

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

The situation differs from other application-driven language extensions in that the new constructor can actually be expressed using constructors available in the DL $\mathcal{ALCQ}$, which can be handled by state-of-the-art DL systems (Section 3).

Name	Syntax	Semantics
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
at-least qualified number restriction	$\geqslant n\, r.C$	$\{x \mid card(\{y \mid (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}) \geq n\}$

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

Thus, the new constructor can be seen as syntactic sugar; nevertheless, it makes sense to introduce it explicitly since this speeds up reasoning. In fact, expressing the new constructor with the ones available in $\mathcal{ALCQ}$ results in an exponential blow-up. In addition, the translation introduces many "expensive" constructors (disjunction and qualified number restrictions). For this reason, even highly optimized DL systems like RACER [14] cannot handle the translated concepts in a satisfactory way. In contrast, the direct introduction of the new constructor into $\mathcal{ALCQ}$ does not increase the complexity of reasoning (Section 4). Moreover, in the process engineering application [24] mentioned above, the rather inexpressive DL $\mathcal{EL}^{(n)}$ that provides only the new constructor together with conjunction is sufficient. In addition, only concept descriptions are used where in each conjunction there is at most one n-ary existential restriction for each role. For this restricted DL, the subsumption problem is polynomial (Section 5). If this last restriction is removed, then subsumption is in coNP, but the exact complexity of the subsumption problem in $\mathcal{EL}^{(n)}$ is still open (Section 6). If one allows to impose disjointness statements between concept names (Section 7), then subsumption between restricted $\mathcal{EL}^{(n)}$-concept descriptions remains polynomial. In the case of unrestricted $\mathcal{EL}^{(n)}$-concept descriptions, subsumption can then be shown to be coNP-complete.

2 The DL $\mathcal{ALCQ}$

Concept descriptions are inductively defined with the help of a set of *constructors*, starting with a set N_C of *concept names* and a set N_R of *role names*. The constructors determine the expressive power of the DL. In this section, we restrict the attention to the DL $\mathcal{ALCQ}$, whose concept descriptions are formed using the constructors shown in Table 1. Using these constructors, several other constructors can be defined as abbreviations:

- $C \sqcup D := \neg(\neg C \sqcap \neg D)$ (disjunction),
- $\top := A \sqcup \neg A$ for a concept name A (top-concept),
- $\exists r.C := \geqslant 1\, r.C$ (existential restriction),
- $\forall r.C := \neg \exists r.\neg C$ (value restriction),

- $\leqslant n\, r.C := \neg(\geqslant (n+1)\, r.C)$ (at-most restriction).

The semantics of $\mathcal{ALCQ}$-concept descriptions is defined in terms of an *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$. The domain $\Delta^{\mathcal{I}}$ of $\mathcal{I}$ is a non-empty set of individuals and the interpretation function $\cdot^{\mathcal{I}}$ maps each concept name $A \in N_C$ to a subset $A^{\mathcal{I}}$ of $\Delta^{\mathcal{I}}$ and each role $r \in N_R$ to a binary relation $r^{\mathcal{I}}$ on $\Delta^{\mathcal{I}}$. The extension of $\cdot^{\mathcal{I}}$ to arbitrary concept descriptions is inductively defined, as shown in the third column of Table 1. Here, the function *card* yields the cardinality of the given set.

A *general $\mathcal{ALCQ}$-TBox* is a finite set of general concept inclusions (GCIs) $C \sqsubseteq D$ where C, D are $\mathcal{ALCQ}$-concept descriptions. The interpretation $\mathcal{I}$ is a model of the general $\mathcal{ALCQ}$-TBox $\mathcal{T}$ iff it satisfies all its GCIs, i.e., if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all GCIs $C \sqsubseteq D$ in $\mathcal{T}$.

We use $C \equiv D$ as an abbreviation of the two GCIs $C \sqsubseteq D$, $D \sqsubseteq C$. An *acyclic $\mathcal{ALCQ}$-TBox* is a finite set of *concept definitions* of the form $A \equiv C$ (where A is a concept name and C an $\mathcal{ALCQ}$-concept description) that does not contain multiple definitions or cyclic dependencies between the definitions. Concept names occurring on the left-hand side of a concept definition are called *defined* whereas the others are called *primitive*.

Given two $\mathcal{ALCQ}$-concept descriptions C, D we say that C *is subsumed by D w.r.t. the general TBox $\mathcal{T}$* ($C \sqsubseteq_{\mathcal{T}} D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$. Subsumption w.r.t. an acyclic TBox and subsumption between concept descriptions (where $\mathcal{T}$ is empty) are special cases of this definition. In the latter case we write $C \sqsubseteq D$ in place of $C \sqsubseteq_{\emptyset} D$. The concept description C is *satisfiable* (w.r.t. the general TBox $\mathcal{T}$) iff there is an interpretation $\mathcal{I}$ (a model $\mathcal{I}$ of $\mathcal{T}$) such that $C^{\mathcal{I}} \neq \emptyset$.

The complexity of the subsumption problem in $\mathcal{ALCQ}$ depends on the presence of GCIs. Subsumption of $\mathcal{ALCQ}$-concept descriptions (with or without acyclic TBoxes) is PSpace-complete and subsumption w.r.t. a general $\mathcal{ALCQ}$-TBox is ExpTime-complete [25].[1] These results hold both for unary and binary coding of the numbers in number restriction, but in this paper we restrict the attention to unary coding (where the size of the number n is counted as n rather than $\log n$).

3 The new constructor

The general syntax of the new constructor is

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

[1] In [25], acyclic TBoxes are not considered, but it is easy to show that the usual approach for handling acyclic TBoxes without using exponential space [19] extends to $\mathcal{ALCQ}$ (see [7]).

where $r \in N_R$, $n \geq 1$, and $C_1, \ldots, C_n$ are concept descriptions. We call this expression an *n-ary existential restriction*. Its semantics is defined as

$$\begin{aligned}\exists r.(C_1, \ldots, C_n)^{\mathcal{I}} := \{x \mid \exists y_1, \ldots, y_n.\ (x, y_1) \in r^{\mathcal{I}} \wedge \ldots \wedge (x, y_n) \in r^{\mathcal{I}} \wedge \\ y_1 \in C_1^{\mathcal{I}} \wedge \ldots \wedge y_n \in C_n^{\mathcal{I}} \wedge \textstyle\bigwedge_{1 \leq i < j \leq n} y_i \neq y_j\}.\end{aligned}$$

We call the DL whose concept descriptions are formed using the constructors conjunction, negation, and n-ary existential restriction $\mathcal{EL}^{(n)}\mathcal{C}$. It is an immediate consequence of the semantics of n-ary existential restrictions that the at-least restriction $\geqslant n\, r.C$ can be expressed by the n-ary existential restriction $\exists r.(C, \ldots, C)$.[2] Consequently, all of $\mathcal{ALCQ}$ can be expressed within $\mathcal{EL}^{(n)}\mathcal{C}$.

Conversely, can we express n-ary existential restrictions within $\mathcal{ALCQ}$? We have seen in the introduction that, in general, $\exists r.(C_1, \ldots, C_n)$ cannot be replaced by the conjunction $\exists r.C_1 \sqcap \ldots \sqcap \exists r.C_n$ since this conjunction does not ensure the existence of n *different* r-successors. However, $\mathcal{ALCQ}$ provides us with the more expressive qualified number restriction constructor. Let us first consider the case $n = 2$. We claim that $\exists r.(C_1, C_2)$ can be expressed by the $\mathcal{ALCQ}$-concept description

$$D := (\geqslant 1\, r.C_1) \sqcap (\geqslant 1\, r.C_2) \sqcap (\geqslant 2\, r.(C_1 \sqcup C_2)).$$

It is clear that any individual belonging to $\exists r.(C_1, C_2)$ also belongs to D. Conversely, assume that x belongs to D. Then x has two distinct r-successors y_1, y_2, both belonging to $C_1 \sqcup C_2$. If one of them belongs to C_1 and the other to C_2, then we are done. Otherwise, we have two cases: (i) both belong to $C_1 \sqcap \neg C_2$, or (ii) both belong to $\neg C_1 \sqcap C_2$. We restrict our attention to the first case (since the second is symmetric). Due to the conjunct $\geqslant 1\, r.C_2$ in D, x has an r-successor in C_2, which is different from y_1 since y_1 does not belong to C_2. Consequently, there are two distinct r-successors of x, one belonging to C_1 and the other belonging to C_2, which shows that x belongs to $\exists r.(C_1, C_2)$.

This result can be extended to arbitrary n.

Theorem 3.1 *The n-ary existential restriction constructor can be expressed within $\mathcal{ALCQ}$, and thus $\mathcal{ALCQ}$ and $\mathcal{EL}^{(n)}\mathcal{C}$ have the same expressive power.*

To prove this theorem we show that $\exists r.(C_1, \ldots, C_n)$ can be expressed by the $\mathcal{ALCQ}$-concept description

$$D_n := \bigsqcap_{\{i_1,\ldots,i_k\} \subseteq \{1,\ldots,n\}} (\geqslant k\, r.(C_{i_1} \sqcup \ldots \sqcup C_{i_k})).$$

It is again clear that any individual belonging to the concept $\exists r.(C_1, \ldots, C_n)$ also belongs to D_n. The other direction is an easy consequence of Hall's theorem

[2] Since we assume unary coding of numbers in number restrictions, this translation is linear. Otherwise, it would be exponential.

[15]. Let $F = (S_1, \ldots, S_n)$ be a finite family of sets. This family has a *system of distinct representatives (SDR)* iff there are n distinct elements $s_1, \ldots, s_n$ such that $s_i \in S_i$ $(i = 1, \ldots, n)$.

Theorem 3.2 (Hall) *The family $F = (S_1, \ldots, S_n)$ has an SDR iff card$(S_{i_1} \cup \ldots \cup S_{i_k}) \geq k$ for all $\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$, where $i_1, \ldots, i_k$ are distinct.*

Now, assume that the individual x belongs to D_n. For $i = 1, \ldots, n$, let S_i be the set of r-successors of x that belong to C_i. By the definition of D_n, the family $(S_1, \ldots, S_n)$ satisfies the condition of Hall's theorem, and thus it has an SDR. This SDR obviously shows that x belongs to $\exists r.(C_1, \ldots, C_n)$.

The proof of Theorem 3.1 shows that the subsumption problem in $\mathcal{EL}^{(n)}\mathcal{C}$ can be reduced to the subsumption problem in $\mathcal{ALCQ}$, and thus DL systems like Racer that can handle $\mathcal{ALCQ}$ can in principle be used to compute subsumption in $\mathcal{EL}^{(n)}\mathcal{C}$. However, the translation from $\mathcal{EL}^{(n)}\mathcal{C}$ into $\mathcal{ALCQ}$ is obviously exponential. In addition, the constructs it introduces (disjunctions and qualified number restrictions) are hard to handle for tableau-based subsumption algorithms like the one used by Racer. In fact, faced with the $\mathcal{ALCQ}$-translations of the $\mathcal{EL}^{(n)}\mathcal{C}$-concept descriptions

$$\begin{array}{lcl} C & := & \exists r.(A_1 \sqcap B_1, A_2 \sqcap B_2, A_3 \sqcap B_3, A_4 \sqcap B_4), \\ D & := & \exists r.(A_1, A_2, A_3, A_4), \end{array}$$

it takes Racer[3] 57 minutes to find out that $C \sqsubseteq D$. For the 5-ary variant of this example, Racer did not finish its computation within 4 hours.

This problem can be due either to the inherently higher complexity of reasoning in $\mathcal{EL}^{(n)}\mathcal{C}$, or to the translation. We will see in the next section that the latter is the culprit.

4 Complexity of reasoning in $\mathcal{EL}^{(n)}\mathcal{C}$

The exponential translation of $\mathcal{EL}^{(n)}\mathcal{C}$-concepts into $\mathcal{ALCQ}$-concepts together with the known complexity of the subsumption problem in $\mathcal{ALCQ}$ (see Section 2) yields the following complexity upper-bounds for the subsumption problem in $\mathcal{EL}^{(n)}\mathcal{C}$: ExpSpace for subsumption of concept descriptions and 2ExpTime for subsumption w.r.t. a general TBox. The next theorem shows that these upper-bounds are not optimal.

[3] Racer Version 1.7.23; on a Pentium 4 machine, 2 Ghz, 2 GB memory; under Redhat Linux.

Theorem 4.1 *The subsumption problem in $\mathcal{EL}^{(n)}\mathcal{C}$ is* PSPACE*-complete for subsumption between concept descriptions and* EXPTIME*-complete for subsumption w.r.t. a general TBox.*

The hardness results are an immediate consequence of the corresponding hardness results [12] for the subsumption problem in $\mathcal{ALC}$ (which allows for conjunction, negation, and existential restrictions). Since $\mathcal{EL}^{(n)}\mathcal{C}$ is closed under negation, it is enough to prove the upper bounds for the satisfiability problem. To show the PSPACE-upper bound, we adapt the "witness algorithm" (also called **K**-worlds algorithm) commonly used in modal logics to show that satisfiability in the modal logic **K** is in PSPACE (see, e.g., [8]). The EXPTIME-upper bound is proved by an adaptation of Pratt's "elimination of Hintikka sets" approach to show that satisfiability in propositional dynamic logic (PDL) is in EXPTIME (see also [8]). But first, we must introduce some notation.

In the following, we assume that all concept descriptions are built using only the constructors conjunction, negation, and n-ary existential restriction. We use $\mathsf{sub}(C)$ to denote the set of all *subconcepts* of C, $\mathsf{sub}(\mathcal{T})$ to denote

$$\bigcup_{C \sqsubseteq D \in \mathcal{T}} (\mathsf{sub}(C) \cup \mathsf{sub}(D)),$$

and define the *closure* of C and $\mathcal{T}$ as

$$\mathsf{cl}(C, \mathcal{T}) := \mathsf{sub}(C) \cup \mathsf{sub}(\mathcal{T}) \cup \{\neg D \mid D \in \mathsf{sub}(C) \cup \mathsf{sub}(\mathcal{T})\}.$$

We use $\mathsf{cl}(C)$ as an abbreviation for $\mathsf{cl}(C, \emptyset)$. Let Γ be a set of concept descriptions. A set $\Psi \subseteq \Gamma$ is a *type for* Γ iff it satisfies the following conditions:

- for all $C \sqcap D \in \Gamma$: $C \sqcap D \in \Psi$ iff $\{C, D\} \subseteq \Psi$;
- for all $\neg(C \sqcap D) \in \Gamma$: $\neg(C \sqcap D) \in \Psi$ iff $\{\neg C, \neg D\} \cap \Psi \neq \emptyset$;
- for all $\neg C \in \Gamma$: $\neg C \in \Psi$ iff $C \notin \Psi$.

Intiuitively, a type for $\mathsf{cl}(C, \mathcal{T})$ can be used to describe to which subconcepts of $C, \mathcal{T}$ an individual of a given interpretation belongs or not. Individuals having identical types behave the same w.r.t. subconcepts of $C, \mathcal{T}$, and thus, in the algorithms, types can be used to represent the relevant properties of individuals. Basically, the EXPTIME-upper bound is due to the fact that there are only exponentially many types for $\mathsf{cl}(C, \mathcal{T})$. In case $\mathcal{T}$ is empty, there are still exponentially many types, but the way one goes through them is such that only polynomially many of them need to be held in memory at the same time.

Let Γ be a set of concept descriptions. Then $\mathsf{rol}_\exists(\Gamma)$ denotes the set of role names r such that $\exists r.(C_1, \ldots, C_k) \in \Gamma$ for some sequence of concept descriptions

$C_1, \ldots, C_k$; moreover, for every role name r we set

$$\begin{array}{rcl} r\text{-}\mathsf{con}(\Gamma) & := & \{C_1, \ldots, C_k \mid \exists r.(C_1, \ldots, C_k) \in \Gamma \text{ or } \neg\exists r.(C_1, \ldots, C_k) \in \Gamma\}, \\ r\text{-}\mathsf{cl}(\Gamma) & := & \{D, \neg D \mid D \in \mathsf{sub}(E) \text{ for some } E \in r\text{-}\mathsf{con}(\Gamma)\}, \\ N_r(\Gamma) & := & \sum_{\exists r.(C_1,\ldots,C_k)\in\Gamma} k. \end{array}$$

Finally, let $\Psi \subseteq \Gamma$, $\Phi_0, \ldots, \Phi_{n-1}$ a (possibly empty) sequence of subsets of Γ, and r a role name. Then $\Phi_0, \ldots, \Phi_{n-1}$ is a *successor candidate* for Ψ w.r.t. r and Γ if, for all $\exists r.(C_1, \ldots, C_k) \in \Gamma$, 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 $1 \leq j \leq k$ and $i_j \neq i_\ell$ for $1 \leq j < \ell \leq k$.

Lemma 4.2 *Let Γ be a set of concept descriptions and $\Psi, \Phi_0, \ldots, \Phi_{n-1}$ subsets of Γ. It is decidable in polynomial time whether $\Phi_0, \ldots, \Phi_{n-1}$ is a successor candidate for Ψ w.r.t. r and Γ.*

Proof. It is enough to show that, for each $\exists r.(C_1, \ldots, C_k) \in \Gamma$, we can decide in polynomial time whether there are $i_1, \ldots, i_k < n$ such that $C_j \in \Phi_{i_j}$ for $1 \leq j \leq k$ and $i_j \neq i_\ell$ for $1 \leq j < \ell \leq k$.

For each $j, 1 \leq j \leq k$ we define the set

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

Then $(S_1, \ldots, S_k)$ has an SDR iff there are distinct indices $i_1, \ldots, i_k < n$ such that $C_j \in \Phi_{i_j}$ for $1 \leq j \leq k$. The existence of an SDR can be decided in polynomial time by a reduction to the maximum bipartite matching problem (see Section 5.2 for more details). □

define procedure $\mathcal{EL}^{(n)}\mathcal{C}$-$\mathsf{World}(\Delta, \Gamma)$
  if Δ is not a type for Γ **then**
    return false
  for all $r \in \mathsf{rol}_\exists(\Delta)$ **do**
    non-deterministically choose an $n \leq N_r(\Gamma)$ and sets $\Psi_0, \ldots, \Psi_{n-1} \subseteq r\text{-}\mathsf{cl}(\Delta)$
    if $\Psi_0, \ldots, \Psi_{n-1}$ is not a successor candidate for Δ w.r.t. r and Γ **then**
      return false
    for all $i < n$ **do**
      if $\mathcal{EL}^{(n)}\mathcal{C}$-$\mathsf{World}(\Psi_i, r\text{-}\mathsf{cl}(\Delta)) = \mathsf{false}$ **then**
        return false
  return true

Figure 1: The procedure $\mathcal{EL}^{(n)}\mathcal{C}$-World.

The following lemma shows that the procedure $\mathcal{EL}^{(n)}\mathcal{C}$-$\mathsf{World}$ introduced in Fig. 1 decides the satisfiability of $\mathcal{EL}^{(n)}\mathcal{C}$-concept descriptions. Since, with every recursive call of the procedure, the maximal role depth of concept descriptions occurring in its arguments decreases, the resursion depth of the algorithm is bounded polynomially.[4] Thus, $\mathcal{EL}^{(n)}\mathcal{C}$-$\mathsf{World}$ is a non-deterministic polynomial space algorithm for $\mathcal{EL}^{(n)}\mathcal{C}$-satisfiability. Because of Savitch's theorem, which says that PSpace = NPSpace, this yields the desired PSpace upper-bound.

Lemma 4.3 *The $\mathcal{EL}^{(n)}\mathcal{C}$-concept description C_0 is satisfiable iff there exists a set $\Psi \subseteq \mathsf{cl}(C_0)$ with $C_0 \in \Psi$ such that $\mathcal{EL}^{(n)}\mathcal{C}$-$\mathsf{World}(\Psi, \mathsf{cl}(C_0))$ returns* true.

Proof. First suppose that $C_0 \in \Psi$ and $\mathcal{EL}^{(n)}\mathcal{C}$-$\mathsf{World}(\Psi, \mathsf{cl}(C_0))$ returns true. Let T be the recursion tree of a successful run of $\mathcal{EL}^{(n)}\mathcal{C}$-$\mathsf{World}(\Psi, \mathsf{cl}(C_0))$, i.e., $T = (V, E, \ell_\Delta, \ell_\Gamma)$ is a finite tree where the node labelling function ℓ_Δ (ℓ_Γ) assigns, to each node, the first (second) argument of the corresponding recursive call. Additionally, we assume that, for each node $v \in V$ except the root, $\ell_R(v)$ returns the role name that the **for all** loop was processing when making the recursion call that generated v. We define an interpretation $\mathcal{I}$ as follows:

$$
\begin{array}{rcl}
\Delta^\mathcal{I} & := & V \\
A^\mathcal{I} & := & \{v \in V \mid A \in \ell_\Delta(v)\} \\
r^\mathcal{I} & := & \{(v, v') \mid v' \text{ is a successor of } v \text{ in } T \text{ and } \ell_R(v') = r\ \}
\end{array}
$$

We prove by structural induction on C that, for all $v \in \Delta^\mathcal{I}$ and all $C \in \ell_\Gamma(v)$, we have $v \in C^\mathcal{I}$ iff $C \in \ell_\Delta(v)$. For the root v_0 this implies $v_0 \in C_0^\mathcal{I}$ since $C_0 \in \Psi = \ell_\Delta(v_0)$.

The base case is straightforward by the definition of $\mathcal{I}$. The Boolean cases are easy since, for each $v \in V$, $\ell_\Delta(v)$ is a type and $\ell_\Gamma(v)$ is closed under building subconcepts. The remaining case concerns the n-ary existential restriction constructor.

For the "if" direction, let $v \in V$ and $\exists r.(C_1, \ldots, C_k) \in \ell_\Delta(v)$. Then $r \in \mathsf{rol}_\exists(\ell_\Delta(v))$, and there exists a successor candidate $\Psi_0, \ldots, \Psi_{n-1}$ for $\ell_\Delta(v)$ w.r.t. r and $\ell_\Gamma(v)$ and (distinct) nodes $v_1, \ldots, v_{n-1}$ such that, for $i < n$, $\ell_\Delta(v_i) = \Psi_i$, v_i is a successor of v in T, and $\ell_R(v_i) = r$. By the definition of $\mathcal{I}$, we have $(v, v_i) \in r^\mathcal{I}$ for $i < n$, and by the definition of successor candidates, there are k distinct indices $i_1, \ldots, i_k$ such that $C_j \in \Psi_{i_j}$ for $1 \leq j \leq k$. The induction hypothesis yields $v_{i_j} \in C_j^\mathcal{I}$ for $1 \leq j \leq k$. This shows that $v \in (\exists r.(C_1, \ldots, C_k))^\mathcal{I}$.

For the "only if" direction, let $v \in (\exists r.(C_1, \ldots, C_k))^\mathcal{I}$. Then there are distinct nodes $v_1, \ldots, v_k \in \Delta^\mathcal{I}$ such that, for $1 \leq j \leq k$, $(v, v_j) \in r^\mathcal{I}$ and $v_j \in C_j^\mathcal{I}$. By the construction of $\mathcal{I}$, the nodes $v_1, \ldots, v_k$ are (distinct) successors of v in T and $\ell_R(v_j) = r$ for $1 \leq j \leq k$. It follows that $r \in \mathsf{rol}_\exists(v)$, and the definition

[4] The *role depth* of a concept is the nesting depth of existential constructors in this concept.

of the procedure $\mathcal{EL}^{(n)}\mathcal{C}$-World implies that there exists a successor candidate $\Psi_0, \ldots, \Psi_{n-1}$ for $\ell_\Delta(v)$ w.r.t. r and $\ell_\Gamma(v)$ and distinct indices $i_1, \ldots, i_k$ such that $\ell_\Delta(v_j) = \Psi_{i_j}$ for $1 \leq j \leq k$. Since $C_j \in r\text{-}\mathsf{cl}(\ell_\Delta(v)) = \ell_\Gamma(v_j)$, we can apply the induction hypothesis. Thus, $v_j \in C_j^{\mathcal{I}}$ implies $C_j \in \ell_\Delta(v_j) = \Psi_{i_j}$ for $1 \leq j \leq k$. By the definition of successor candidates, this implies $\exists r.(C_1, \ldots, C_k) \in \ell_\Delta(v)$.

Now assume that C_0 is satisfiable, let $\mathcal{I}$ be a model of C_0, and $x_0 \in C_0^{\mathcal{I}}$. For $x \in \Delta^{\mathcal{I}}$ and Γ a set of concepts, we define

$$\mathsf{tp}_\Gamma(x) := \{C \in \Gamma \mid x \in C^{\mathcal{I}}\}.$$

We use $\mathcal{I}$ to guide the non-deterministic choices of $\mathcal{EL}^{(n)}\mathcal{C}$-World. To describe this in more detail, it is convenient to pass an element of $\Delta^{\mathcal{I}}$ as a "virtual" third argument to $\mathcal{EL}^{(n)}\mathcal{C}$-World. Initially, we call $\mathcal{EL}^{(n)}\mathcal{C}$-World with arguments $(\mathsf{tp}_{\mathsf{cl}(C_0)}(x_0), \mathsf{cl}(C_0), x_0)$.

Now, assume $\mathcal{EL}^{(n)}\mathcal{C}$-World is called with arguments (Δ, Γ, x). By induction, we assume that $\Delta = \mathsf{tp}_\Gamma(x)$. For every role $r \in \mathsf{rol}_\exists(\Delta)$ we must execute the body of the **for all** loop. First, we must determine the number n of components of the successor candidate to be chosen. For every $\exists r.(C_1, \ldots, C_k) \in \Delta$ we have $x \in \exists r.(C_1, \ldots, C_k)^{\mathcal{I}}$, and thus there are k distinct r-successors $x_1, \ldots, x_k$ of x in $\mathcal{I}$ such that $x_i \in C_i^{\mathcal{I}}$ for $i = 1, \ldots, k$. For a given such concept description $\exists r.(C_1, \ldots, C_k) \in \Delta$ there may be more than one such tuple of r-successor; then we just select an arbitrary one of them. Let $y_0, \ldots, y_{n-1}$ be the collection of all r-successors of x that are selected if we go through all $\exists r.(C_1, \ldots, C_k) \in \Delta$ in this way. By the definition of $N_r(\Gamma)$, we have $n \leq N_r(\Gamma)$, and thus n is an eligible choice for the size of the successor candidate. The components $\Psi_0, \ldots, \Psi_{n-1}$ of the successor candidate are obtained by setting $\Psi_i := \mathsf{tp}_{r\text{-}\mathsf{cl}(\Delta)}(y_i)$ for $i < n$. As the additional third argument, we pass y_i to the recursive call of $\mathcal{EL}^{(n)}\mathcal{C}$-World with first two arguments Ψ_i and $r\text{-}\mathsf{cl}(\Delta)$. It is routine to show that, when guided in this way, the algorithm returns true. $\square$

Let us now turn to the case of satisfiability w.r.t. a general TBox. Let C be a concept and $\mathcal{T}$ a TBox. A set $\Psi \subseteq \mathsf{cl}(C, \mathcal{T})$ is a *type for C and $\mathcal{T}$* if it is a type for $cl(C, \mathcal{T})$ and additionally satisfies the following property: for all $D \sqsubseteq E \in \mathcal{T}$, $D \in \Psi$ implies $E \in \Psi$.

A type Γ is called *moribund* 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 and $\mathsf{cl}(C, \mathcal{T})$.

Lemma 4.4 *The procedure $\mathcal{EL}^{(n)}\mathcal{C}$-Elim introduced in Fig. 2 decides satisfiability of C_0 w.r.t. $\mathcal{T}$ in exponential time.*

Proof. The **repeat** loop of $\mathcal{EL}^{(n)}\mathcal{C}$-Elim terminates after at most exponentially many steps since there are exponentially many types and, in each elimination

define procedure $\mathcal{EL}^{(n)}\mathcal{C}$-$\mathsf{Elim}(C, \mathcal{T})$

Set $i := 0$ and $\mathfrak{T}_0$ to the set of all types for C and $\mathcal{T}$

repeat

$\mathfrak{T}_{i+1} := \{\Gamma \in \mathfrak{T}_i \mid \Gamma \text{ is not moribund in } \mathfrak{T}_i\}$

$i := i + 1$

until $\mathfrak{T}_i = \mathfrak{T}_{i-1}$

if there is a $\Gamma \in \mathfrak{T}_i$ with $C \in \Gamma$ **then**

return true

return false

Figure 2: The procedure $\mathcal{EL}^{(n)}\mathcal{C}$-$\mathsf{Elim}$.

round, at least one type is eliminated. Checking whether a type is moribund can be done in exponential time since there are at most exponentially many sequences of types of length at most $N_r(\Gamma)$. By Lemma 4.2, for each such sequence, it can be checked in polynomial time whether it is a successor candidate. Thus, $\mathcal{EL}^{(n)}\mathcal{C}$-$\mathsf{Elim}$ is a (deterministic) exponential time procedure.

Assume that $\mathcal{EL}^{(n)}\mathcal{C}$-$\mathsf{Elim}$ terminates returning true, let $\mathfrak{T}$ be the set of types that have not been eliminated, and let $\Gamma_{C_0} \in \mathfrak{T}$ be such that $C_0 \in \Gamma_{C_0}$. Let $\Gamma \in \mathfrak{T}$ and $r \in \mathsf{rol}_\exists(\Gamma)$. Since Γ was not eliminated, it has a successor candidate $\Psi_0, \ldots, \Psi_{n-1}$ where all the components Ψ_i also belong to $\mathfrak{T}$. It should be noted, however, that these types need not be pairwise distinct. For this reason, it is not enough to take just the types in $\mathfrak{T}$ as the elements of our model. To have enough copies of each type available, we define

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

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

- $\Delta^\mathcal{I} := \{(\Gamma, i) \mid 1 \leq i \leq N \text{ and } \Gamma \in \mathfrak{T}\}$.
- $A^\mathcal{I} := \{(\Gamma, i) \in \Delta^\mathcal{I} \mid A \in \Gamma\}$ for all concept names A.
- Let $(\Gamma, i) \in \Delta^\mathcal{I}$ and $r \in \mathsf{rol}_\exists(\Gamma)$, and assume that Γ contains m existential restrictions for r. Since Γ was not eliminated, these restrictions have successor candidates $\Psi^j_1, \ldots, \Psi^j_{n_j}$ for $1 \leq j \leq m$. By our definition of N_r, we know that $\sum_{i=1}^m n_j \leq N_r(\Gamma) \leq N$. Thus, we can define the set

$$\{(\Psi^j_i, i + \sum_{\nu=1}^{j-1} n_\nu) \mid 1 \leq j \leq m \text{ and } 1 \leq i \leq n_j\}$$

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

It is straightforward to prove by structural induction C 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 $C \in \Gamma$.

It follows 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 $E \in \Gamma$ by the definition of the notion "type for C_0 and $\mathcal{T}$." Thus, we also have $(\Gamma, i) \in E^{\mathcal{I}}$. To sum up, we have shown that $\mathcal{I}$ is a model of $\mathcal{T}$ that interprets C_0 as a non-empty set.

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

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

It is readily checked that no type in $\mathfrak{T} := \{\mathsf{tp}(x) \mid x \in \Delta^{\mathcal{I}}\}$ is eliminated by $\mathcal{EL}^{(n)}\mathcal{C}$-$\mathsf{Elim}$. Since $\mathsf{tp}(x_0)$ contains C_0, $\mathcal{EL}^{(n)}\mathcal{C}$-$\mathsf{Elim}$ returns true. □

5 A tractable sublanguage

In the chemical process engineering application mentioned above [24], the full expressive power of $\mathcal{EL}^{(n)}\mathcal{C}$ is actually not needed. This application is concerned with supporting the construction of mathematical models of process systems by storing building blocks for such models in a class hierarchy. In order to retrieve building blocks, one can then either browse the hierarchy or formulate query classes. In both cases, the existence of efficient algorithms for computing subsumption between class descriptions is an important prerequisite.

5.1 Restricted $\mathcal{EL}^{(n)}$-concept descriptions

The frame-like formalism for describing classes of such building blocks introduced in [24] can be expressed in the *sublanguage* $\mathcal{EL}^{(n)}$ *of* $\mathcal{EL}^{(n)}\mathcal{C}$, which allows for conjunction, n-ary existential restrictions, and the top concept. Moreover, since in each frame a given slot-name can be used only once, it is sufficient to consider *restricted* $\mathcal{EL}^{(n)}$*-concept descriptions* where in each conjunction there is at most one n-ary existential restriction for each role: an $\mathcal{EL}^{(n)}$-concept description is *restricted* iff it is of the form

$$A_1 \sqcap \ldots \sqcap A_n \sqcap \exists r_1.(B_{1,1}, \ldots, B_{1,k_1}) \sqcap \ldots \sqcap \exists r_m.(B_{m,1}, \ldots, B_{m,k_m}),$$

where $A_1, \ldots, A_n$ are concept names, $r_1, \ldots, r_m$ are *distinct* role names, and $B_{1,1}, \ldots, B_{m,k_m}$ are restricted $\mathcal{EL}^{(n)}$-concept descriptions. For example, the $\mathcal{EL}^{(n)}$-concept description $\exists r.(A, \exists r.(B, C)) \sqcap \exists s.(A, A)$ is restricted whereas the description $\exists r.(A, \exists r.(B, C)) \sqcap \exists r.(A, A)$ is not.

As in the case of $\mathcal{EL}$ [5], the corresponding DL with unary existential restrictions, each restricted $\mathcal{EL}^{(n)}$-concept description C can be translated into an $\mathcal{EL}^{(n)}$-*description tree* T_C, where the nodes are labeled with sets of concept names and the edges are labeled with role names. Formally, this tree is described by a tuple $T_C = (V, E, v_0, \ell)$, where V is the finite set of nodes, $E \subseteq V \times N_R \times V$ is the set of N_R-labeled edges, $v_0 \in V$ is the root, and $\ell : V \longrightarrow 2^{N_C}$ is the node labeling function. The set of all concept names occurring in the top-level conjunction of C yields the label $\ell(v_0)$ of the root v_0, and each existential restriction $\exists r.(C_1, \ldots, C_n)$ in this conjunction yields n r-successor of v_0 that are the roots of the trees corresponding to C_i. For example, the restricted $\mathcal{EL}^{(n)}$-concept descriptions

$$A \sqcap \exists r.(A, B \sqcap \exists r.(B, A), \exists r.(A, A \sqcap B)) \quad \text{and} \quad A \sqcap \exists r.(A, B, \exists r.(A, A))$$

yield the description trees depicted in Fig. 3.

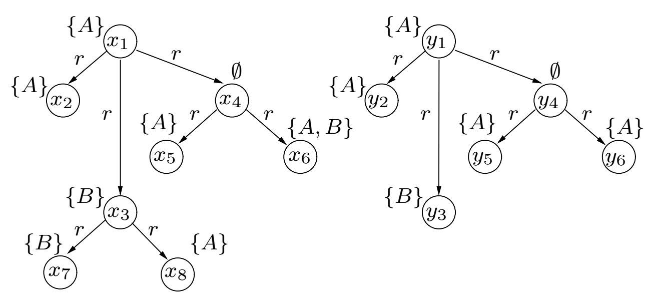

Figure 3: Two $\mathcal{EL}^{(n)}$-description trees.

In [5], it was shown that subsumption between $\mathcal{EL}$-concept descriptions corresponds to the existence of a homomorphism between the corresponding description trees. In $\mathcal{EL}^{(n)}$ we must additionally require that the homomorphism is injective.

Definition 5.1 *Given two $\mathcal{EL}^{(n)}$-description trees $T_1 = (V_1, E_1, v_{0,1}, \ell_1)$ and $T_2 = (V_2, E_2, v_{0,2}, \ell_2)$, a* homomorphism *$\varphi : T_1 \longrightarrow T_2$ is a mapping $\varphi : V_1 \longrightarrow V_2$ such that*

1. $\varphi(v_{0,1}) = v_{0,2}$,
2. $\ell_1(v) \subseteq \ell_2(\varphi(v))$ *for all* $v \in V_1$, *and*
3. $(\varphi(v), r, \varphi(w)) \in E_2$ *for all* $(v, r, w) \in E_1$.

This homomorphism is an embedding *iff the mapping $\varphi : V_1 \longrightarrow V_2$ is injective.*

For example, mapping y_i to x_i for $i = 1, \ldots, 6$ yields an embedding from the description tree on the right-hand side of Fig. 3 to the description tree on the left-hand side. If we changed the label of x_6 to $\{B\}$, then there would still exist a homomorphism between the two trees (mapping both y_5 and y_6 onto x_5), but not an embedding.

Theorem 5.2 *Let C, D be restricted $\mathcal{EL}^{(n)}$-concept descriptions and T_C, T_D the corresponding description trees. Then $C \sqsubseteq D$ iff there exists an embedding from T_D into T_C.*

The proof of this theorem is similar to the proof of the corresponding result for $\mathcal{EL}$ [5].

First, note that any interpretation can be viewed as a graph. An $\mathcal{EL}^{(n)}$*-graph* is of the form $G = (V, E, \ell)$, where V is a non-empty set, $E \subseteq V \times N_R \times V$, and $\ell : V \longrightarrow 2^{N_C}$.[5] A given interpretation $\mathcal{I}$ can be represented by an $\mathcal{EL}^{(n)}$-graph $G_\mathcal{I} = (V, E, \ell)$, where

- $V = \Delta^\mathcal{I}$;
- given a node $u \in \Delta^\mathcal{I}$, its label is

$$\ell(u) = \{A \mid A \text{ is a concept names such that } u \in A^\mathcal{I}\},$$

- and $E = \{(u, r, v) \mid (u, v) \in r^\mathcal{I}\}$.

Conversely, any $\mathcal{EL}^{(n)}$-graph G obviously represents an interpretation $\mathcal{I}_G$. For example, the $\mathcal{EL}^{(n)}$-graph depicted on the left-hand side of Fig. 3 represents the interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, where

- $\Delta^\mathcal{I} = \{x_1, \ldots, x_8\}$;
- $A^\mathcal{I} = \{x_1, x_2, x_5, x_6, x_8\}$ and $B^\mathcal{I} = \{x_3, x_6, x_7\}$;
- $r^\mathcal{I} = \{(x_1, x_2), (x_1, x_3), (x_1, x_4), (x_3, x_7), (x_3, x_8), (x_4, x_5), (x_4, x_6)\}$.

Definition 5.3 *Given two $\mathcal{EL}^{(n)}$-graphs $G_1 = (V_1, E_1, \ell_1)$ and $G_2 = (V_2, E_2, \ell_2)$, a mapping $\varphi : V_1 \longrightarrow V_2$ is an $\mathcal{EL}^{(n)}$-*homomorphism *of G_1 into G_2 iff it satisfies 2. and 3. of Definition 5.1, and the following* local injectivity *condition:*

$$(u, r, v) \in E \ \wedge \ (u, r, v') \in E \ \wedge \ v \neq v' \quad \Rightarrow \quad \varphi(v) \neq \varphi(v').$$

[5] Note that $\mathcal{EL}^{(n)}$-description trees are also $\mathcal{EL}^{(n)}$-graphs.

Obviously, any embedding between $\mathcal{EL}^{(n)}$-description trees is also an $\mathcal{EL}^{(n)}$-homomorphism. Conversely, if $\varphi : T_1 \longrightarrow T_2$ is an $\mathcal{EL}^{(n)}$-homomorphism between the $\mathcal{EL}^{(n)}$-description trees T_1, T_2 that maps the root of T_1 onto the root of T_2, then φ is an embedding between these trees. In addition, if $\varphi_1 : T_1 \longrightarrow T_2$ is an embedding between $\mathcal{EL}^{(n)}$-description trees and $\varphi_2 : T_2 \longrightarrow G$ is an $\mathcal{EL}^{(n)}$-homomorphism, then their composition $\varphi_1 \circ \varphi_2 : T_1 \longrightarrow G$ is also an $\mathcal{EL}^{(n)}$-homomorphism.

Lemma 5.4 *Let C be a restricted $\mathcal{EL}^{(n)}$-concept description, $\mathcal{I}$ an interpretation, and $d_0 \in \Delta^{\mathcal{I}}$. Then $d_0 \in C^{\mathcal{I}}$ iff there is an $\mathcal{EL}^{(n)}$-homomorphism $\varphi : T_C \longrightarrow G_{\mathcal{I}}$ that maps the root of T_C onto d_0.*

Proof. The restricted $\mathcal{EL}^{(n)}$-concept description C is of the form

$$A_1 \sqcap \ldots \sqcap A_n \sqcap \exists r_1.(B_{1,1}, \ldots, B_{1,k_1}) \sqcap \ldots \sqcap \exists r_m.(B_{m,1}, \ldots, B_{m,k_m}),$$

where $A_1, \ldots, A_n$ are concept names, $r_1, \ldots, r_m$ are distinct role names, and $B_{1,1}, \ldots, B_{m,k_m}$ are restricted $\mathcal{EL}^{(n)}$-concept descriptions. Thus, the corresponding $\mathcal{EL}^{(n)}$-description tree $T_C = (V, E, \ell)$ has the following form:

- it has a root v_0 with label $\ell(v_0) = \{A_1, \ldots, A_n\}$;
- for $1 \le i \le m$ and $1 \le j \le k_i$, this root has an r_i-successor $v_{i,j}$ that is the root of the $\mathcal{EL}^{(n)}$-description tree $T_{B_{i,j}}$ corresponding to $B_{i,j}$.

Let $G_{\mathcal{I}} = (\Delta^{\mathcal{I}}, E_{\mathcal{I}}, \ell_{\mathcal{I}})$.

First, assume that $d_0 \in C^{\mathcal{I}}$. Then $d_0 \in A_i^{\mathcal{I}}$ for $i = 1, \ldots, n$, which shows that $\ell(v_0) \subseteq \ell_{\mathcal{I}}(d_0)$. Thus, if we define $\varphi(v_0) = d_0$, then 2. of Definition 5.1 is satisfied. In addition, for $1 \le i \le m$ and $1 \le j \le k_i$ there are elements $d_{i,j} \in \Delta^{\mathcal{I}}$ such that

- $(d_0, d_{i,j}) \in r_i^{\mathcal{I}}$,
- $d_{i,j} \neq d_{i,j'}$ for $j \neq j'$, and
- $d_{i,j} \in B_{i,j}^{\mathcal{I}}$.

By induction, there are $\mathcal{EL}^{(n)}$-homomorphisms $\varphi_{i,j} : T_{B_{i,j}} \longrightarrow G_{\mathcal{I}}$ such that $\varphi_{i,j}(v_{i,j}) = d_{i,j}$. We define $\varphi : T_C \longrightarrow G_{\mathcal{I}}$ as follows:

$$\varphi(v) := \begin{cases} d_0 & \text{if } v = v_0, \\ \varphi_{i,j}(v) & \text{if } v \text{ is a node in } T_{B_{i,j}}. \end{cases}$$

It is easy to see that φ is indeed a well-defined $\mathcal{EL}^{(n)}$-homomorphism.

Second, assume that there is an $\mathcal{EL}^{(n)}$-homomorphism $\varphi : T_C \longrightarrow G_{\mathcal{I}}$ such that $\varphi(v_0) = d_0$. By 2. of Definition 5.1, $\ell(v_0) = \{A_1, \ldots, A_n\} \subseteq \ell_{\mathcal{I}}(d_0)$, which shows that $d_0 \in A_i^{\mathcal{I}}$ for $i = 1, \ldots, n$. To show $d_0 \in C^{\mathcal{I}}$, it remains to be shown that there are $d_{i,j} \in \Delta^{\mathcal{I}}$ (for $1 \le i \le m$ and $1 \le j \le k_i$) such that

- $(d_0, d_{i,j}) \in r_i^{\mathcal{I}}$,
- $d_{i,j} \neq d_{i,j'}$ for $j \neq j'$, and
- $d_{i,j} \in B_{i,j}^{\mathcal{I}}$.

If we define $d_{i,j} := \varphi(v_{i,j})$, then the fact that φ is an $\mathcal{EL}^{(n)}$-homomorphism implies that the first and the second point are satisfied. In addition, the restriction $\varphi_{i,j}$ of φ to $T_{B_{i,j}}$ is an $\mathcal{EL}^{(n)}$-homomorphism such that $\varphi_{i,j}(v_{i,j}) = d_{i,j}$. By induction, this shows that the third point is satisfied as well. $\square$

We are now ready to prove Theorem 5.2.

First assume that there is an embedding $\varphi : T_D \longrightarrow T_C$ such that $\varphi(v_0) = u_0$ where u_0 is the root of T_C and v_0 is the root of T_D. To show $C \sqsubseteq D$, let $\mathcal{I}$ be an interpretation, and assume that $d_0 \in C^{\mathcal{I}}$. By Lemma 5.4, there is an $\mathcal{EL}^{(n)}$-homomorphism $\varphi' : T_C \longrightarrow G_{\mathcal{I}}$ such that $\varphi'(u_0) = d_0$. But then $\varphi \circ \varphi' : T_D \longrightarrow G_{\mathcal{I}}$ is an $\mathcal{EL}^{(n)}$-homomorphism such that $\varphi \circ \varphi'(v_0) = \varphi'(\varphi(v_0)) = \varphi'(u_0) = d_0$. By Lemma 5.4, this implies $d_0 \in D^{\mathcal{I}}$.

Second, assume that $C \sqsubseteq D$. The $\mathcal{EL}^{(n)}$-description tree T_C is an $\mathcal{EL}^{(n)}$-graph, and thus represents an interpretation $\mathcal{I}$. Let u_0 be the root of T_C. Since the identity map is an $\mathcal{EL}^{(n)}$-homomorphism from T_C into T_C that maps u_0 onto u_0, Lemma 5.4 yields $u_0 \in C^{\mathcal{I}}$. But then $C \sqsubseteq D$ implies $u_0 \in D^{\mathcal{I}}$. By Lemma 5.4, this means that there is an $\mathcal{EL}^{(n)}$-homomorphism $\varphi : T_D \longrightarrow T_C$ such that $\varphi(v_0) = u_0$ where v_0 is the root of T_D. As noted above, such an $\mathcal{EL}^{(n)}$-homomorphism is actually an embedding. This completes the proof of Theorem 5.2.

5.2 Deciding the existence of an embedding

To show that subsumption between restricted $\mathcal{EL}^{(n)}$-concept descriptions is a polynomial-time problem, it remains to be shown that the existence of an embedding can be decided in polynomial time. First, let us recall the well-known bottom-up approach for testing for the existence of a homomorphism [22, 5].

Let $T_1 = (V_1, E_1, v_{0,1}, \ell_1)$ and $T_2 = (V_2, E_2, v_{0,2}, \ell_2)$ be two $\mathcal{EL}^{(n)}$-description trees, and assume that we want to check whether there is a homomorphism from T_1 to T_2. The idea underlying the polynomial time test is to compute, for each $v \in V_1$, the set $\delta(v)$ of all nodes $w \in V_2$ such that there is a homomorphism from the subtree of T_1 with root v to the subtree of T_2 with root w. Once these sets δ are computed for all nodes of T_1, we can simply check whether $v_{0,2}$ belongs to $\delta(v_{0,1})$. The sets $\delta(v)$ are computed in a bottom-up fashion, where a node is treated only after all its successor nodes have been considered:[6]

[6]For example, one can use a postorder tree walk [11] of the nodes of T_1 to realize this.

1. If v is a leaf of T_1, then $\delta(v)$ simply consists of all the nodes $w \in V_2$ such that $\ell_1(v) \subseteq \ell_2(w)$.

2. Let v be a node of T_1 and let $(v, r_1, v_1), \ldots, (v, r_k, v_k)$ be all the edges in E_1 with first component v. Since we work bottom up, we know that the sets $\delta(v_1), \ldots, \delta(v_k)$ have already been computed. The set $\delta(v)$ consists of all the nodes $w \in V_2$ such that

   (a) $\ell_1(v) \subseteq \ell_2(w)$ and

   (b) for each $i, 1 \leq i \leq k$ there exists a node $w_i \in \delta(v_i)$ such that $(w, r_i, w_i) \in E_2$.

It is easy to show that this indeed yields a polynomial-time algorithm for checking the existence of a *homomorphism* between two $\mathcal{EL}^{(n)}$-description trees.

If we want to test for the existence of an *embedding*, we must modify Step 2 of this algorithm. In fact, we must ensure that distinct r-successors of v can be mapped to distinct r-successors of w. This can be achieved as follows:

2′. Let v be a node of T_1, and for each role r let $(v, r, v_{1,r}), \ldots, (v, r, v_{k_r,r})$ be the edges in E_1 with first component v and label r. Since we work bottom up, we know that the sets $\delta(v_{1,r}), \ldots, \delta(v_{k_r,r})$ have already been computed. The set $\delta(v)$ consists of all the nodes $w \in V_2$ satisfying the following two properties:

   (a) $\ell_1(v) \subseteq \ell_2(w)$,

   (b) for all roles r, the family $F_r(w) := (S_{1,r}(w), \ldots, S_{k_r,r}(w))$ has an SDR, where the members of this family are defined as

$$S_{i,r}(w) := \{w' \in \delta(v_{i,r}) \mid (w, r, w') \in E_2\}.$$

Obviously, the existence of an SDR for $F_r(w)$ allows us to map the r-successors of v to *distinct* r-successors of w, and thus construct an embedding. For this algorithm to be polynomial, it remains to be shown that the existence of an SDR can be decided in polynomial time. Note that Hall's characterization of the existence of an SDR obviously does not yield a polynomial-time procedure. However, checking for the existence of an SDR is basically the same as solving the maximum bipartite matching problem, which can be done in polynomial time since it can be reduced to a network flow problem [11].

To be more precise, let $(L \cup R, E)$ be a bipartite graph, i.e., $L \cap R = \emptyset$ and $E \subseteq L \times R$. A *matching* is a subset M of E such that each node in $L \cup R$ occurs at most once in M. This matching is called *maximum* iff there is no other matching having a larger cardinality. As shown in [11], such a maximum matching can be computed in time polynomial in the cardinality of V and E.

Let $F = (S_1, \ldots, S_n)$ be a finite family of finite sets, and let $L := \{1, \ldots, n\}$ and $R = S_1 \cup \ldots \cup S_n$.[7] We define the set of edges of the bipartite graph $G_F = (L \cup R, E)$ as follows:

$$E := \{(i, s) \mid s \in S_i\}.$$

It is easy to see that the family F has an SDR iff the corresponding bipartite graph G_F has a maximum matching of cardinality n. In fact, $(1, s_1), \ldots, (n, s_n)$ is a maximum matching iff $s_1, \ldots, s_n$ is an SDR.

Thus, we have shown that the existence of an embedding can be decided in polynomial time. Together with Theorem 5.2, this yields the following tractability result:

Corollary 5.5 *Subsumption between restricted $\mathcal{EL}^{(n)}$-concept descriptions can be decided in polynomial time.*

A first implementation of this polynomial-time algorithm behaves much better than the translation approach on the example concept descriptions C, D from Section 3 and their obvious extensions to larger n. For small n, the subsumption relationship is found immediately (i.e., with no measurable run-time), and even for $n = 100$, the runtime (of our unoptimized implementation) is just 1 second. One could argue that the comparison of these results with the performance of RACER on the $\mathcal{ALCQ}$-translations of C, D and their extensions to larger n is unfair since the culprit is the exponential translation rather than RACER. However, this is the only known translation of $\mathcal{EL}^{(n)}$-concept descriptions into a DL that can be handled by RACER, and it is the one originally used in the process engineering application.

5.3 Acyclic TBoxes

The frame-like formalism employed in the process engineering application allows to inherit properties from other frames. To represent this feature within our DL approach, a TBox is needed. However, it is sufficent to consider only acyclic $\mathcal{EL}^{(n)}$-TBoxes that are restricted in a similar way as restricted $\mathcal{EL}^{(n)}$-concept descriptions. Formally, an acyclic $\mathcal{EL}^{(n)}$-TBox is called *restricted* iff its concept definitions are of the form

$$A \equiv P_1 \sqcap \ldots \sqcap P_n \sqcap \exists r_1.(A_{1,1}, \ldots, A_{1,\ell_1}) \sqcap \ldots \sqcap \exists r_m.(A_{m,1}, \ldots, A_{m,\ell_m}),$$

where $A, A_{1,1}, \ldots, A_{m,\ell_m}$ are defined concepts, $P_1, \ldots, P_n$ are primitive concepts, and $r_1, \ldots, r_m$ are *distinct* role names.

[7]Without loss of generality we can assume that $L \cap R = \emptyset$.

In the presence of TBoxes, it is obviously sufficient to have an algorithm that decides subsumption between defined concepts. In principle, subsumption between the defined concepts A and B w.r.t. an acyclic and restricted TBox can be decided by first expanding A and B into $\mathcal{EL}^{(n)}$-concept descriptions by replacing defined concept names by their definitions until no more defined concepts occur. Then, subsumption between the expanded concept descriptions can be decided without reference to a TBox. The definition of a restricted $\mathcal{EL}^{(n)}$-TBox makes sure that the expanded $\mathcal{EL}^{(n)}$-concept descriptions are actually restricted, and thus one can use the algorithm described in Section 5.2 to decide subsumption between them. However, it is well-know that the expansion process may lead to an exponential blow-up of the concept descriptions it is applied to [20]. Thus, the approach described above yields a subsumption algorithm that may need exponential time.

In this section we show how to obtain a polynomial-time subsumption algorithm in the presence of restricted acyclic $\mathcal{EL}^{(n)}$-TBoxes. To formulate this algorithm, it is convenient to assume that TBoxes are in a certain form: an $\mathcal{EL}^{(n)}$-TBox $\mathcal{T}$ is in *normal form* if it is acyclic, restricted, and, for all concept definitions $A \doteq C \in \mathcal{T}$, each defined concept name occurs at most once in C. It is not hard to see that every restricted acyclic $\mathcal{EL}^{(n)}$-TBox can be converted into normal form by introducing additional defined concept names. For example, the TBox

$$\begin{array}{rcl} A_1 & \doteq & \exists r.A_2 \sqcap \exists s.(A_2, A_3) \\ A_2 & \doteq & C \\ A_3 & \doteq & D \end{array}$$

can be rewritten into

$$\begin{array}{rcl} A_1 & \doteq & \exists r.A_2 \sqcap \exists s.(A_2', A_3) \\ A_2 & \doteq & C \\ A_2' & \doteq & C \\ A_3 & \doteq & D. \end{array}$$

This conversion can be carried out in polynomial time, and it causes an at most quadratic blowup in size. In the following we assume that all TBoxes are in normal form.

Similar to our representation of restricted $\mathcal{EL}^{(n)}$-concept descriptions as trees, we represent $\mathcal{EL}^{(n)}$-TBoxes in normal form as $\mathcal{EL}^{(n)}$*-directed acyclic graphs (DAGs)*, where the nodes (which are the defined concept names) are labelled with sets of primitive concept names, and the edges are labelled with role names. Formally, an $\mathcal{EL}^{(n)}$-DAG is given by a tuple $G_\mathcal{T} = (V, E, \ell)$, where V is a set of nodes, $E \subseteq V \times N_R \times V$ is a set of N_R-labeled edges that form a directed acyclic graph, and $\ell : V \longrightarrow 2^{N_C}$ is the node labelling function. A given TBox $\mathcal{T}$ in normal form can be translated into the following $\mathcal{EL}^{(n)}$-DAG $G_\mathcal{T} = (V_\mathcal{T}, E_\mathcal{T}, \ell_\mathcal{T})$:

- $V_\mathcal{T}$ is the set of defined concept names in $\mathcal{T}$;

- if $A \equiv P_1 \sqcap \ldots \sqcap P_n \sqcap \exists r_1.(A_{1,1}, \ldots, A_{1,\ell_1}) \sqcap \ldots \sqcap \exists r_m.(A_{m,1}, \ldots, A_{m,\ell_m})$ is in $\mathcal{T}$, then $\ell_{\mathcal{T}}(A) = \{P_1, \ldots, P_n\}$ and A is the source of the edges

$$(A, r_1, A_{1,1}), \ldots, (A, r_1, A_{1,\ell_1}), \ldots, (A, r_m, A_{m,1}), \ldots, (A, r_m, A_{m,\ell_m}) \in E_{\mathcal{T}}.$$

Note that $\mathcal{EL}^{(n)}$-DAGs are a special kind of $\mathcal{EL}$-graphs as introduced for cyclic $\mathcal{EL}$-TBoxes in [1]. The fact that the TBox $\mathcal{T}$ is assumed to be in normal form makes sure that, for every node A of $G_{\mathcal{T}}$, its successor nodes are distinct defined concepts of $\mathcal{T}$. For a node v in an $\mathcal{EL}^{(n)}$-DAG $G = (V, E, \ell)$ we write $S_G(v)$ to denote the set $\{u \mid (v, r, u) \in E \text{ for some } r \in N_R\}$ of its successor nodes.

Definition 5.6 *Let $G = (V, E, \ell)$ be an $\mathcal{EL}^{(n)}$-DAG. For $v, v' \in V$, we say that* v is embeddable into v' in G *if*

1. *$\ell(v) \subseteq \ell(v')$ and*
2. *there exists an injection $\varphi : S_G(v) \to S_G(v')$ such that, for all $u \in S_G(v)$,*
   (a) $(v, r, u) \in E$ implies $(v', r, \varphi(u)) \in E$;
   (b) u is embeddable into $\varphi(u)$.

It is easily seen that being embeddable is well-defined as the recursive use of "embeddable" in the definition refers only to nodes for which the maximum length of a path to a sink (i.e., a node without successor nodes) is strictly smaller.

Theorem 5.7 *Let $\mathcal{T}$ be an $\mathcal{EL}^{(n)}$-TBox in normal form and A, B defined concepts in $\mathcal{T}$. Then $A \sqsubseteq_{\mathcal{T}} B$ iff B is embeddable into A in $G_{\mathcal{T}}$.*

Proof. Let $\widehat{A}$ and $\widehat{B}$ be the results of expanding A and B w.r.t. $\mathcal{T}$. It is well-known that $A \sqsubseteq_{\mathcal{T}} B$ iff $\widehat{A} \sqsubseteq \widehat{B}$. By Theorem 5.2, the latter holds iff there exists an embedding from $T_{\widehat{B}}$ into $T_{\widehat{A}}$. For proving Theorem 5.7, it thus suffices to show that there exists an embedding from $T_{\widehat{B}}$ into $T_{\widehat{A}}$ iff B is embeddable into A in $G_{\mathcal{T}}$. This is proved in what follows. Let $G_{\mathcal{T}} = (V_{\mathcal{T}}, E_{\mathcal{T}}, \ell_{\mathcal{T}})$, $T_{\widehat{A}} = (V_A, E_A, v_A, \ell_A)$, and $T_{\widehat{B}} = (V_B, E_B, u_B, \ell_B)$.

The *proof of the if-direction* is by induction on the depth of $T_{\widehat{B}}$. Assume that B is embeddable into A in $G_{\mathcal{T}}$, and let φ be the injection witnessing Property 2 in the definition of embeddable.

For the induction start, let the depth of $T_{\widehat{B}}$ be zero. Then B is a defined concept name with definition

$$B \equiv P_1 \sqcap \ldots \sqcap P_n,$$

where $P_1, \ldots, P_n$ are primitive concepts. By Property 1 of embeddable and by the construction of $T_{\widehat{B}}$ and $T_{\widehat{A}}$, we have $\ell_B(u_B) = \ell_{\mathcal{T}}(B) \subseteq \ell_{\mathcal{T}}(A) = \ell_A(v_A)$. Thus, the mapping $\psi := \{u_B \mapsto v_A\}$ is an embedding from $T_{\widehat{B}}$ to $T_{\widehat{A}}$.

For the induction step, let $A \equiv C$ and $B \equiv D$ be the definitions of A and B in $\mathcal{T}$. Since B is embeddable into A in $G_\mathcal{T}$, by Property 2a of embeddable, and by construction of $G_\mathcal{T}$, every role occurring in an existential restriction in D also occurs in an existential restriction in C. Thus, C and D can be written as

$$\begin{array}{rcl} C & = & P_1 \sqcap \ldots \sqcap P_n \sqcap \exists r_1.(A_{1,1}, \ldots, A_{1,\ell_1}) \sqcap \ldots \sqcap \exists r_m.(A_{m,1}, \ldots, A_{m,\ell_m}), \\ D & = & Q_1 \sqcap \ldots \sqcap Q_{n'} \sqcap \exists r_1.(B_{1,1}, \ldots, B_{1,\ell'_1}) \sqcap \ldots \sqcap \exists r_{m'}.(B_{m',1}, \ldots, B_{m',\ell'_{m'}}), \end{array}$$

with $m' \leq m$. Let

$$\begin{array}{rcll} I_C & := & \{(i,j) \mid 1 \leq i \leq m \text{ and } 1 \leq j \leq \ell_i\} & \text{and} \\ I_D & := & \{(i,j) \mid 1 \leq i \leq m' \text{ and } 1 \leq j \leq \ell'_i\}. & \end{array}$$

Because $\mathcal{T}$ is assumed to be in normal form, the following holds for $G_\mathcal{T}$:

(a) For $(i,j) \in I_C$, the node A is connected (only) via r_i to $A_{i,j}$. Moreover, mapping (i,j) to $A_{i,j}$ yields a bijection between I_C and $\{A_{i,j} \mid (i,j) \in I_C\}$.

(b) For $(i,j) \in I_D$, the node B is connected (only) via r_i to $B_{i,j}$. Moreover, mapping (i,j) to $B_{i,j}$ yields a bijection between I_D and $\{B_{i,j} \mid (i,j) \in I_D\}$.

Similar properties are satisfied in $T_{\widehat{A}}$ and $T_{\widehat{B}}$:

(c) The root v_A of $T_{\widehat{A}}$ has exactly one successor $v_{i,j}$ for each $(i,j) \in I_C$. Moreover, v_A is connected to $v_{i,j}$ (only) via r_i, and $v_{i,j}$ is the root of the $\mathcal{EL}^{(n)}$-tree $T_{\widehat{A_{i,j}}}$ obtained by expanding $A_{i,j}$.

(d) The root u_B of $T_{\widehat{B}}$ has exactly one successor $u_{i,j}$ for each $(i,j) \in I_D$. Moreover, u_B is connected to $u_{i,j}$ (only) via r_i, and $u_{i,j}$ is the root of the $\mathcal{EL}^{(n)}$-tree $T_{\widehat{B_{i,j}}}$ obtained by expanding $B_{i,j}$.

Let $(i,j) \in I_D$. By Property 2a of embeddable and due to the first part of (a) and (b) above, $\varphi(B_{i,j}) = A_{i,k}$ for some $(i,k) \in I_C$. By Property 2b of embeddable, $B_{i,j}$ is embeddable into $A_{i,k}$ in $G_\mathcal{T}$. The induction hypothesis thus yields embeddings $\psi_{i,j}$ from $T_{\widehat{B_{i,j}}}$ into $T_{\widehat{A_{i,k}}}$.

Now define the mapping $\psi : V_B \to V_A$ by setting $\psi(u_B) := v_A$ and taking the union of all the mappings $\psi_{i,j}$. We claim that ψ is an embedding from $T_{\widehat{B}}$ into $T_{\widehat{A}}$. As the $\psi_{i,j}$ are embeddings and their domains are disjoint, it suffices to consider u_B and its successors. Property 1 of homomorphisms (mapping of root to root) is clearly satisfied. For Property 2 (inclusion of node labels), we can show as in the induction start that $\ell_B(u_B) \subseteq \ell_A(v_A)$. Concerning Property 3 (edge labels), let $(u_B, r_i, u_{i,j}) \in E_B$. Then $\psi_{i,j}(u_{i,j}) = v_{i,k}$ for some k with $1 \leq k \leq \ell_i$. By (c) above, we have $(v_A, r_i, v_{i,k}) \in E_A$ as required.

It remains to be shown that ψ is an embedding. Thus, let $u_{i,j}, u_{i',j'}$ be distinct successors of u_A. We must show that $v_{i,k} := \psi_{i,j}(u_{i,j}) \neq \psi_{i',j'}(u_{i',j'}) =: v_{i',k'}$. Since $u_{i,j} \neq u_{i',j'}$, the index pairs $(i,j), (i',j')$ are distinct elements of I_D, and thus $B_{i,j}, B_{i',j'}$ are distinct defined concepts. By the definition of the mappings $\psi_{i,j}$ and $\psi_{i',j'}$, we have $\varphi(B_{i,j}) = A_{i,k}$ and $\varphi(B_{i',j'}) = A_{i',k'}$, and thus $A_{i,k}, A_{i',k'}$ are distinct defined concepts by the definition of embeddable. This shows that the index pairs $(i,k), (i',k')$ are distinct elements of I_C, and thus $v_{i,k} \neq v_{i',k'}$ by Property (c) above. Note that this also implies that the codomains of $\psi_{i,j}$ and $\psi_{i',j'}$ are disjoint.

The *proof of the only-if-direction* is again by induction on the depth of $T_{\widehat{B}}$. Let ψ be an embedding from $T_{\widehat{B}}$ to $T_{\widehat{A}}$.

For the induction start, let the depth of $T_{\widehat{B}}$ be zero. Since $\psi(u_B) = v_A$ and by construction of $T_{\widehat{B}}$ and $T_{\widehat{A}}$, we have $\ell_{\mathcal{T}}(B) = \ell_B(u_B) \subseteq \ell_A(v_A) = \ell_{\mathcal{T}}(A)$. As the depth of $T_{\widehat{B}}$ is zero, B does not have any outgoing edges in $G_{\mathcal{T}}$. Thus, B is embeddable into A in $G_{\mathcal{T}}$.

For the induction step, let $A \equiv C$ and $B \equiv D$ be the definitions of A and B in $\mathcal{T}$. Since $\psi(u_B) = v_A$, by Property 3 of homomorphisms, and by construction of $T_{\widehat{A}}$ and $T_{\widehat{B}}$, every role name occurring in an existential restriction in D also occurs in an existential restriction in C. Thus, C and D can be written exactly as in the proof of the if-direction. Let I_C and I_D be defined as above. The successors of A and B in $G_{\mathcal{T}}$, of v_A in $T_{\widehat{A}}$, and of u_B in $T_{\widehat{B}}$ also satisfy the properties (a)–(d) stated in the proof of the if-direction.

We must show that B is embeddable into A in $G_{\mathcal{T}}$. For Property 1 of embeddable, we can show as in the induction start that $\ell_{\mathcal{T}}(B) \subseteq \ell_{\mathcal{T}}(A)$. For Property 2, define the mapping $\varphi : S_G(B) \to S_G(A)$ by setting, for each $(i,j) \in I_D$, $\varphi(B_{i,j}) = A_{i',j'}$ if $\psi(u_{i,j}) = v_{i',j'}$.

First, we show that φ is an injection. If $B_{i_1,j_1} \neq B_{i_2,j_2}$, then $(i_1,j_1) \neq (i_2,j_2)$, and thus $u_{i_1,j_1} \neq u_{i_2,j_2}$. Since ψ is an embedding, this implies that $v_{i'_1,j'_1} := \psi(u_{i_1,j_1}) \neq \psi(u_{i_2,j_2}) =: v_{i'_2,j'_2}$. Finally, this implies $A_{i_1,j_1} \neq A_{i'_2,j'_2}$ by Property (a) above.

It thus remains to be shown that φ satisfies Properties 2a and 2b of embeddable. For Property 2a, let $(i,j) \in I_D$. By Property (b), $(B, r, B_{i,j}) \in E_{\mathcal{T}}$ implies $r = r_i$. Due to Property 3 of homomorphisms, $\psi(u_{i,j}) = v_{i,k}$ for some $(i,k) \in I_C$. Thus, $\varphi(B_{i,j}) = A_{i,k}$. By Property (a) above, we have $(A, r_i, A_{i,k}) \in E_{\mathcal{T}}$ as required. To show Property 3a, consider again a tuple $(i,j) \in I_D$. We must show that $B_{i,j}$ is embeddable into $\varphi(B_{i,j})$ in $G_{\mathcal{T}}$. Let $\psi(u_{i,j}) = v_{i,k}$. Then $\varphi(B_{i,j}) = A_{i,k}$. Clearly, ψ is an embedding from $T_{\widehat{B_{i,j}}}$ (the subtree of T_B with root $u_{i,j}$) into $T_{\widehat{A_{i,k}}}$ (the subtree of T_A with root $v_{i,k}$). It follows by the induction hypothesis that $B_{i,j}$ is embeddable into $A_{i,k}$ in $G_{\mathcal{T}}$, as required by Property 3a. $\square$

It remains to note that, to deciding whether B is embeddable into A in $G_{\mathcal{T}}$, we can use the marking algorithm for testing for the existence of an embedding between description trees presented in Section 5.2: the bottom-up labelling strategy of "treating a node only after all its successor nodes have been considered" works also on DAGs, and it is not hard to verify that B is embeddable into A in $G_{\mathcal{T}}$ iff A occurs in the marking $\delta(B)$ of B.

Corollary 5.8 *Subsumption in $\mathcal{EL}^{(n)}$ w.r.t. a restricted acyclic TBox can be decided in polynomial time.*

6 Unrestricted $\mathcal{EL}^{(n)}$-concept descriptions

In such concept descriptions, several n-ary existential restrictions for the same role r can occur in a conjunction, such as in the description

$$C_u := A \sqcap \exists r.(A, B) \sqcap \exists r.(\exists r.A \sqcap \exists r.A).$$

If we translate this unrestricted $\mathcal{EL}^{(n)}$-concept description into a description tree, then we obtain the tree on the right-hand side of Fig. 3, which is also obtained as a translation of the restricted $\mathcal{EL}^{(n)}$-concept description

$$C_r := A \sqcap \exists r.(A, B, \exists r.(A, A)).$$

To distinguish between these two descriptions, we introduce *distinctness classes*: for each node x in the tree and each role r, the r-successors of x are partitioned into such classes. For example, in the tree corresponding to C_u, the r-successors of y_1 are partitioned into the sets $\{y_2, y_3\}$, $\{y_4\}$, whereas there is only one distinctness class $\{y_2, y_3, y_4\}$ for these nodes in the tree corresponding to C_r.

The notion of an *embedding* must take these distinctness classes into account. Instead of requiring that the homomorphism φ is injective, we require that it is injective on distinctness classes.

Definition 6.1 *Given two $\mathcal{EL}^{(n)}$-description trees T_1, T_2 that are equipped with distinctness classes, a homomorphism $\varphi : T_1 \longrightarrow T_2$ is called an* embedding *iff for each node x in T_1 and each distinctness class $\{x_1, \ldots, x_k\}$ of the r-successors of x, the nodes $\varphi(x_1), \ldots, \varphi(x_k)$ are* distinct *r-successors of $\varphi(x)$.*

However, if we just change the notion of an embedding in this way, then Theorem 5.2 obviously does not hold for unrestricted $\mathcal{EL}^{(n)}$-concept descriptions. In fact, if $\varphi(x_1), \ldots, \varphi(x_k)$ do not belong to the same distinctness class in T_2, then we cannot be sure that they really represent distinct individuals. For example, if $C = \exists r.A \sqcap \exists r.B$ and $D = \exists r.(A, B)$, then there is an embedding from T_D into T_C, but D does not subsume C.

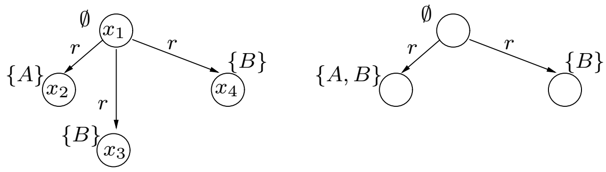

Figure 4: Identification of $\mathcal{EL}^{(n)}$-description trees.

Thus, an obvious conjecture could be that the embedding must *respect distinctness classes*, i.e., we must require $\varphi(x_1), \dots, \varphi(x_k)$ to belong to the same distinctness class. However, the following example shows that this requirement is too strong. Let $C = \exists r.A \sqcap \exists r.(B, B)$ and $D = \exists r.(A, B)$. There is no embedding from T_D to T_C that respects distinctness classes, but it is easy to see that D subsumes C.

Before we can formulate a correct characterization of subsumption between unrestricted $\mathcal{EL}^{(n)}$-concept descriptions, we must introduce some notation.

Definition 6.2 *Given an $\mathcal{EL}^{(n)}$- description tree $T = (V, E, v_0, \ell)$ where role successors are partitioned into distinctness classes, an* identification on T *is an equivalence relation $\sim$ on V such that $v_1 \sim v_2$ implies that*

- *there are $u_1, u_2 \in V$ and a role r such that v_1 is an r-successor of u_1, v_2 is an r-successor of u_2, and $u_1 \sim u_2$;*
- *if $v_1 \neq v_2$, then v_1, v_2 do not belong to the same distinctness class.*

Any identification $\sim$ on T induces a description tree $T/\sim$ whose nodes are the $\sim$-equivalence classes $[v]_\sim := \{u \in V \mid u \sim v\}$, whose root is $[v_0]_\sim$, and whose edges and node labels are defined as follows:

$$E_\sim := \{([u]_\sim, r, [v]_\sim) \mid \text{there is } u' \in [u]_\sim, v' \in [v]_\sim \text{ such that } (u', r, v') \in E\},$$
$$\ell_\sim([u]_\sim) := \bigcup_{u' \in [u]_\sim} \ell(u').$$

Note that the first condition on identifications in the above definition ensures that the graph defined this way is indeed a tree with root $[v_0]_\sim$.

For example, the $\mathcal{EL}^{(n)}$-description tree T_C corresponding to $C = \exists r.A \sqcap \exists r.(B, B)$ is depicted on the left-hand side of Fig. 4, where the r-successors of x_1 are partitioned into the distinctness classes $\{x_2\}, \{x_3, x_4\}$. There are three different identifications: the identity relation, the relation where in addition $x_2 \sim x_3$, and the relation where in addition $x_2 \sim x_4$. The $\mathcal{EL}^{(n)}$-description tree induced by the identity relation is T_C itself, whereas the trees induced by the other two identifications are isomorphic to the tree depicted on the right-hand side of Fig. 4. Obviously, there is an embedding of the $\mathcal{EL}^{(n)}$-description tree T_D corresponding to $D = \exists r.(A, B)$ into each of these two trees.

Theorem 6.3 *Let C, D be (unrestricted) $\mathcal{EL}^{(n)}$-concept descriptions and T_C, T_D the corresponding description trees. Then $C \sqsubseteq D$ iff for every identification $\sim$ on T_C there exists an embedding from T_D into $T_C/\!\sim$.*

Before proving this theorem, let us point out that it yields an NP-algorithm for testing *non*-subsumption of unrestricted $\mathcal{EL}^{(n)}$-concept descriptions: guess in non-deterministic polynomial time an identification $\sim$ of T_C, and then check in polynomial time (by a simple adaptation of the algorithm described in Section 5.2) whether there is an embedding from T_D into $T_C/\!\sim$.

Corollary 6.4 *The subsumption problem for (unrestricted) $\mathcal{EL}^{(n)}$-concept descriptions is in coNP.*

Before we can prove Theorem 6.3, we must first show that the auxiliary definitions and results from Section 5.1 can be adapted to the case of unrestricted $\mathcal{EL}^{(n)}$-concept descriptions.

Definition 6.5 *Let $T_1 = (V_1, E_1, v_{0,1}, \ell_1)$ be an $\mathcal{EL}^{(n)}$-description tree that is equipped with distinctness classes, and let $G_2 = (V_2, E_2, \ell_2)$ be an $\mathcal{EL}^{(n)}$-graph. The mapping $\varphi : V_1 \longrightarrow V_2$ is an $\mathcal{EL}^{(n)}$-*homomorphism *iff it satisfies 2. and 3. of Definition 5.1, and is* injective on the distinctness classes *of T_1, i.e.,*

if $v \neq v'$ belong to the same distinctness class of T_1, then $\varphi(v) \neq \varphi(v')$.

If G_2 is also an $\mathcal{EL}^{(n)}$-description tree and φ maps the root of T_1 onto the root of G_2, then it is easy to see that φ is an embedding in the sense of Definition 6.1.

With this adapted notion of an $\mathcal{EL}^{(n)}$-homomorphism, the following analogon of Lemma 5.4 can easily be proved.

Lemma 6.6 *Let C be an (unrestricted) $\mathcal{EL}^{(n)}$-concept description, $\mathcal{I}$ an interpretation, and $d_0 \in \Delta^{\mathcal{I}}$. Then $d_0 \in C^{\mathcal{I}}$ iff there is an $\mathcal{EL}^{(n)}$-homomorphism $\varphi : T_C \longrightarrow G_{\mathcal{I}}$ that maps the root of T_C onto d_0.*

In order to prove *"$\Rightarrow$" of Theorem 6.3*, we assume that $C \sqsubseteq D$. Let u_0 be the root of T_C, and $\sim$ an identification on T_C. The $\mathcal{EL}^{(n)}$-description tree $T_C/\!\sim$ represents an interpretation $\mathcal{I}$. It is easy to see that the mapping

$$\theta : T_C \longrightarrow T_C/\!\sim\, : u \mapsto [u]_\sim$$

is an $\mathcal{EL}^{(n)}$-homomorphism with $\theta(u_0) = [u_0]_\sim$. By Lemma 6.6, this implies $[u_0]_\sim \in C^{\mathcal{I}}$, and thus $[u_0]_\sim \in D^{\mathcal{I}}$. But then Lemma 6.6 also implies that there is an $\mathcal{EL}^{(n)}$-homomorphism $\varphi : T_D \longrightarrow T_C/\!\sim$ such that $\varphi(v_0) = [u_0]_\sim$, where v_0 is the root of T_D. As noted above, this homomorphism is in fact an embedding.

In order to prove *"$\Leftarrow$" of Theorem 6.3*, we assume that for every identification $\sim$ on T_C there exists an embedding from T_D into $T_C/\!\sim$. To show that this implies $C \sqsubseteq D$, let $\mathcal{I}$ be an interpretation, and assume that $d_0 \in C^\mathcal{I}$. By Lemma 6.6, this implies that there is an $\mathcal{EL}^{(n)}$-homomorphism $\varphi : T_C \longrightarrow G_\mathcal{I}$ such that $\varphi(u_0) = d_0$, where u_0 is the root of T_C. This homomorphism induces a binary relation $\sim_\varphi$ on the nodes of T_C, which we define by induction on the depth of nodes:

- The root u_0 of T is the only node on depth 0, and we have $u_0 \sim_\varphi u_0$.
- Assume that $\sim_\varphi$ is already defined on nodes of depth n for $n \geq 0$. If v_1, v_2 are nodes on depth $n+1$, then

  $v_1 \sim_\varphi v_2$ iff there are nodes $u_1 \sim_\varphi u_2$ on depth n and a role r such that v_1 is an r-successor of u_1, v_2 is an r-successor of u_2, and $\varphi(v_1) = \varphi(v_2)$.

It is easy to see that $\sim_\varphi$ is an identification on T_C.

The $\mathcal{EL}^{(n)}$-homomorphism $\varphi : T_C \longrightarrow G_\mathcal{I}$ induces the following mapping from the nodes of $T_C/\!\sim_\varphi$ to the nodes of $G_\mathcal{I}$:

$$\widehat{\varphi}([u]_{\sim_\varphi}) := \varphi(u).$$

Note that the definition of $\sim_\varphi$ implies that $\widehat{\varphi}$ is well-defined.

By our assumption, there is an embedding $\psi : T_D \longrightarrow T_C/\!\sim_\varphi$. We claim that the composition $\psi \circ \widehat{\varphi}$ is an $\mathcal{EL}^{(n)}$-homomorphism from T_D into $G_\mathcal{I}$ such that $\psi \circ \widehat{\varphi}(v_0) = d_0$, where v_0 is the root of T_D. By Lemma 6.6, this implies $d_0 \in D^\mathcal{I}$, which completes the proof of Theorem 6.3.

To prove the claim, first note that $\psi \circ \widehat{\varphi}(v_0) = \widehat{\varphi}(\psi(v_0)) = \widehat{\varphi}([u_0]_{\sim_\varphi}) = \varphi(u_0) = d_0$.

Second, let v be a node of T_D and assume that A belongs to its label in T_D. Since ψ is an embedding, this implies that A belongs to the label of $[u]_{\sim_\varphi} := \psi(v)$. Thus, there is $u' \sim_\varphi u$ such that A belongs to the label of u' in T_C. Since φ is an $\mathcal{EL}^{(n)}$-homomorphism, this implies that A belongs to the label of $\varphi(u')$ in $G_\mathcal{I}$. However, since $u' \sim_\varphi u$ we know that $\varphi(u') = \varphi(u) = \widehat{\varphi}([u]_{\sim_\varphi}) = \widehat{\varphi}(\psi(v))$. This shows that A belongs to the label of $\psi \circ \widehat{\varphi}(v)$ in $G_\mathcal{I}$.

Third, assume that (v_1, r, v_2) is an edge in T_D. Let $[u_1]_{\sim_\varphi} := \psi(v_1)$ and $[u_2]_{\sim_\varphi} := \psi(v_2)$. Since ψ is an embedding, $([u_1]_{\sim_\varphi}, r, [u_2]_{\sim_\varphi})$ is an edge in $T_C/\!\sim_\varphi$. By the definition of $T_C/\!\sim_\varphi$, this means that there are $u'_1 \sim_\varphi u_1$ and $u'_2 \sim_\varphi u_2$ such that (u'_1, r, u'_2) is an edge in T_C. Since φ is an $\mathcal{EL}^{(n)}$-homomorphism, this implies that $(\varphi(u'_1), r, \varphi(u'_2))$ is an edge in $G_\mathcal{I}$. However, since $u'_1 \sim_\varphi u_1$ and $u'_2 \sim_\varphi u_2$ we have, for $i = 1, 2$, that $\varphi(u'_i) = \varphi(u_i) = \widehat{\varphi}([u_i]_{\sim_\varphi}) = \widehat{\varphi}(\psi(v_i))$. This shows that $(\psi \circ \widehat{\varphi}(v_1), r, \psi \circ \widehat{\varphi}(v_2))$ is an edge in $G_{\mathcal{I}}$.

Finally, assume that v_1, v_2 are distinct r-successors of a common parent node v in T_D that belong to the same distinctness class. Let $[u]_{\sim_\varphi} := \psi(v)$, $[u_1]_{\sim_\varphi} := \psi(v_1)$, and $[u_2]_{\sim_\varphi} := \psi(v_2)$. Since ψ is an embedding, we know that $\psi(v_1) \neq \psi(v_2)$, and thus $u_1 \not\sim_\varphi u_2$. If we can show that this implies $\varphi(u_1) \neq \varphi(u_2)$, then we are done: since, for $i = 1, 2$, we have $\varphi(u_i) = \widehat{\varphi}([u_i]_{\sim_\varphi}) = \widehat{\varphi}(\psi(v_i))$, we then also have $\psi \circ \widehat{\varphi}(v_1) \neq \psi \circ \widehat{\varphi}(v_2)$.

To show that $u_1 \not\sim_\varphi u_2$ implies $\varphi(u_1) \neq \varphi(u_2)$, it is sufficient to show that, in T_C, there are nodes $w_1 \sim_\varphi w_2$ such that u_1 is an r-successor of w_1 and u_2 is an r-successor of w_2. Since ψ is an embedding, we know that, in $T_C/{\sim_\varphi}$, the node $[u_i]_{\sim_\varphi} = \psi(v_i)$ is an r-successor of $[u]_{\sim_\varphi} = \psi(v)$, for $i = 1, 2$. By the definition of $T_C/{\sim_\varphi}$, this implies that there are nodes w'_i, u'_i such that $w'_i \sim_\varphi u$ and $u'_i \sim_\varphi u_i$, and u'_i is an r-successor of w'_i. By the definition of $\sim_\varphi$, $u'_i \sim_\varphi u_i$ implies that there is a node $w_i \sim_\varphi w'_i$ such that u_i is an r-successor of w_i. Transitivity of $\sim_\varphi$ yields $w_1 \sim_\varphi w_2$.

This finishes the proof of Theorem 6.3.

7 Adding disjointness statements

In the chemical process engineering application motivating this paper, the real-world concepts expressed by concept names are often disjoint. For example, an object cannot be both an apparatus and a plant. Disjointness statements of the form $\mathit{dis}(P, Q)$, where P, Q are concept names, allow us to express such additional knowledge. An interpretation $\mathcal{I}$ is a model of a set of disjointness statements $\mathcal{D}$ iff $P^{\mathcal{I}} \cap Q^{\mathcal{I}} = \emptyset$ for all statements $\mathit{dis}(P, Q)$ in $\mathcal{D}$. Satisfiability and subsumption w.r.t. $\mathcal{D}$ are defined in the usual way: C is satisfiable w.r.t. $\mathcal{D}$ iff there is a model $\mathcal{I}$ of $\mathcal{D}$ such that $C^{\mathcal{I}} \neq \emptyset$; and C is subsumed by D w.r.t. $\mathcal{D}$ ($C \sqsubseteq_{\mathcal{D}} D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{D}$.

The following lemma shows that (un)satisfiability w.r.t. a set of disjointness statements is easy to decide.

Lemma 7.1 *The $\mathcal{EL}^{(n)}$-concept description C is unsatisfiable w.r.t. the set of disjointness statements $\mathcal{D}$ iff there is a disjointness statement $\mathit{dis}(P, Q)$ in $\mathcal{D}$ and a node v in T_C whose label contains P and Q.*

Proof. If there is a disjointness statement $\mathit{dis}(P, Q)$ in $\mathcal{D}$ and a node v in T_C containing P and Q, then C is obviously unsatisfiable w.r.t. $\mathcal{D}$.

Conversely, assume that, for all disjointness statement $\mathit{dis}(P, Q)$ in $\mathcal{D}$ and all nodes v in T_C, the label of v does not contain both P and Q. The $\mathcal{EL}^{(n)}$-

description tree T_C represents an interpretation $\mathcal{I}$. Because of our assumption, this interpretation is in fact a model of $\mathcal{D}$. The identity map is an $\mathcal{EL}^{(n)}$-homomorphism from T_C into T_C that maps the root u_0 of T_C onto u_0. By Lemma 6.6, this implies that $u_0 \in C^{\mathcal{I}}$, and thus C is satisfiable. $\square$

How does adding disjointness statements influence the complexity of the subsumption problem? Both for restricted and for unrestricted $\mathcal{EL}^{(n)}$-concept descriptions, the characterization of the subsumption problem (Theorem 5.2 and Theorem 6.3) can easily be extended to deal with disjointness statements.

For restricted $\mathcal{EL}^{(n)}$-concept descriptions, the only effect that disjointness statements have is that they can make concepts unsatisfiable.

Theorem 7.2 *Let C, D be* restricted *$\mathcal{EL}^{(n)}$-concept descriptions. Then $C \sqsubseteq_{\mathcal{D}} D$ iff*

1. *either C is unsatisfiable w.r.t. $\mathcal{D}$, or*
2. *both C and D are satisfiable w.r.t. $\mathcal{D}$ and $C \sqsubseteq D$.*

Proof. The "if" direction of the theorem is trivial.

To show the "only-if" direction, assume that $C \sqsubseteq_{\mathcal{D}} D$ and that C is satisfiable w.r.t. $\mathcal{D}$. The $\mathcal{EL}^{(n)}$-description tree T_C represents an interpretation $\mathcal{I}$, and the assumption that C is satisfiable w.r.t. $\mathcal{D}$ implies that $\mathcal{I}$ is a model of $\mathcal{D}$ (see the proof of Lemma 7.1). In addition, as shown in the proof of Lemma 7.1, the root u_0 of T_C satisfies $u_0 \in C^{\mathcal{I}}$. Thus, $C \sqsubseteq_{\mathcal{D}} D$ yields that $u_0 \in D^{\mathcal{I}}$. By Lemma 5.4, this implies that there is an $\mathcal{EL}^{(n)}$-homomorphism $\varphi : T_D \longrightarrow G_{\mathcal{I}} = T_C$ such that $\varphi(v_0) = u_0$. As noted in Section 5.1, such a homomorphism is actually an embedding. By Theorem 5.2, this implies that $C \sqsubseteq D$. $\square$

Corollary 7.3 *For restricted $\mathcal{EL}^{(n)}$-concept descriptions, subsumption w.r.t. disjointness statements can be decided in polynomial time.*

The effect of disjointness statements is less trivial if we consider unrestricted $\mathcal{EL}^{(n)}$-concept descriptions. The reason is that disjointness statements can enforce r-successors to be interpreted by distinct objects even though they do not belong to the same distinctness class. This problem does not occur for restricted $\mathcal{EL}^{(n)}$-concept descriptions since there all r-successors of a given node belong to the same distinctness class.

Before we can formulate a characterization of subsumption w.r.t. disjointness statements in the unrestricted case, we must modify the definition of an identification such that it takes disjointness statements into account.

Definition 7.4 *Let $\mathcal{D}$ be a set of disjointness statements and $T = (V, E, v_0, \ell)$ an $\mathcal{EL}^{(n)}$-description tree where role successors are partitioned into distinctness classes. The identification $\sim$ on T is* compatible with $\mathcal{D}$ *iff $u \sim v$ implies $\{P, Q\} \not\subseteq \ell(u) \cup \ell(v)$ for all $dis(P, Q)$ in $\mathcal{D}$.*

Let C be an (unrestricted) $\mathcal{EL}^{(n)}$-concept description. If the identification $\sim$ on T_C is compatible with $\mathcal{D}$, then the interpretation $\mathcal{I}$ represented by the tree $T_C/\sim$ is a model of $\mathcal{D}$. In particular, the identity relation is compatible with $\mathcal{D}$ iff C is satisfiable w.r.t. $\mathcal{D}$. If C is unsatisfiable w.r.t. $\mathcal{D}$, then no identification on T_C is compatible with $\mathcal{D}$.

Theorem 7.5 *Let $\mathcal{D}$ be a set of disjointness statements, C, D (unrestricted) $\mathcal{EL}^{(n)}$-concept descriptions, and T_C, T_D the corresponding description trees. Then $C \sqsubseteq_{\mathcal{D}} D$ iff for every identification $\sim$ on T_C that is compatible with $\mathcal{D}$ there exists an embedding from T_D into $T_C/\sim$.*

Proof. The proof of "$\Rightarrow$" is basically identical to the proof of "$\Rightarrow$" of Theorem 6.3. The only additional fact to note is that the compatibility of $\sim$ with $\mathcal{D}$ implies that the interpretation $\mathcal{I}$ represented by the tree $T_C/\sim$ is a model of $\mathcal{D}$.

The proof of "$\Leftarrow$" is also very similar to the proof of "$\Leftarrow$" of Theorem 6.3. Here the only additional thing to note is that the fact that $\mathcal{I}$ is a model of $\mathcal{D}$ implies that $\sim_\varphi$ is compatible with $\mathcal{D}$. In fact, assume to the contrary that there are nodes $v_1 \sim_\varphi v_2$ in T_C and a disjointness statement $dis(P, Q)$ in $\mathcal{D}$ such that $\{P, Q\} \subseteq \ell(u) \cup \ell(v)$. But then the label of $\varphi(v_1) = \varphi(v_2)$ in $G_{\mathcal{I}}$ contains both P and Q, which shows that $\mathcal{I}$ does not satisfy the disjointness statement $dis(P, Q)$. $\square$

Again, this theorem yields an NP-algorithm for non-subsumption, and thus the subsumption problem is in coNP. In the presence of disjointness constraints, we can also prove the matching lower bound.[8]

Corollary 7.6 *The subsumption problem for (unrestricted) $\mathcal{EL}^{(n)}$-concept descriptions w.r.t. disjointness statements is coNP-complete.*

Proof. The coNP-upper bound can be show as in the case of unrestricted $\mathcal{EL}^{(n)}$-concept descriptions without disjointness statements.

We show coNP-hardness by a reduction of *graph 3-colorability* to non-subsumption. A given undirected graph $G = (V, E)$ is 3-colorable iff there is a mapping $f : V \longrightarrow \{1, 2, 3\}$ such that $\{u, v\} \in E$ implies $f(u) \neq f(v)$. It is well-known (see

[8]The idea for this reduction is due to an anonymous referee.

[13]) that the 3-colorability problem, i.e., the question whether a given graph is 3-colorable, is NP-complete.

Let $G = (V, E)$ be an undirected graph with n vertices, i.e., $V = \{v_1, \ldots, v_n\}$. Without loss of generality we assume that this graph has no loops, i.e., $\{u, v\} \in E$ implies $u \neq v$. Let $A_1, \ldots, A_n$ be concept names. The graph $G = (V, E)$ is represented by the set of disjointness statements

$$\mathcal{D}_G := \{\mathit{dis}(A_i, A_j) \mid \{v_i, v_j\} \in E\}.$$

Let $C := \exists r.A_1 \sqcap \ldots \sqcap \exists r.A_n$ and $D := \exists r.(\top, \top, \top, \top)$. We claim that $C \not\sqsubseteq_{\mathcal{D}_G} D$ iff G is 3-colorable.

Without loss of generality we may assume that the $\mathcal{EL}^{(n)}$-description tree T_C corresponding to C has nodes $v_0, v_1, \ldots, v_n$ where v_0 is the root, and $v_1, \ldots, v_n$ are the r-successors of v_0 such that v_i has label $\{A_i\}$. Note that every node v_i belongs to a singleton distinctness class.

First, assume that G is 3-colorable, and let $f : V \longrightarrow \{1, 2, 3\}$ be the corresponding mapping. The binary relation

$$\sim_f := \{(v_i, v_j) \mid f(v_i) = f(v_j)\}$$

is an identification on T_C. In addition, since $\{v_i, v_j\} \in E$ implies $f(v_i) \neq f(v_j)$, it is compatible with $\mathcal{D}_G$. Since in $T_C/{\sim_f}$ the root has at most 3 different r-successors, there cannot be an embedding from T_D into $T_C/{\sim_f}$. By Theorem 7.5, this implies $C \not\sqsubseteq_{\mathcal{D}_G} D$.

Conversely, assume that $C \not\sqsubseteq_{\mathcal{D}_G} D$. Then there is an identification $\sim$ on T_C such that

- $\sim$ is compatible with $\mathcal{D}_G$; and
- there is no embedding from T_D into $T_C/{\sim}$.

The second fact implies that the root of $T_C/{\sim}$ has at most 3 r-successors. In the following, we treat the case where it has exactly 3 r-successors. (The other two cases can be treated similarly.) Thus, the root $[v_0]_\sim$ of $T_C/{\sim}$ has three r-successors u_1, u_2, u_3. These r-successors are $\sim$-equivalence classes, which partition the r-successors $v_1, \ldots, v_n$ of v_0 in T_C. We define

$$f_\sim : \{v_1, \ldots, v_n\} \longrightarrow \{1, 2, 3\} : v_i \mapsto \nu \quad \text{where } \nu \text{ is such that } u_\nu = [v_i]_\sim.$$

Let $\{v_i, v_j\} \in E$. Then $\mathit{dis}(A_i, A_j)$ belongs to $\mathcal{D}_G$, and thus the compatibility of $\sim$ with $\mathcal{D}_G$ implies that $v_i \not\sim v_j$. Consequently, $f(v_i) \neq f(v_j)$, which shows that G is 3-colorable. $\square$

8 Related and future work

Polynomiality of the subsumption problem in $\mathcal{EL}$ was shown in [5] as a by-product of the characterization of subsumption via the existence of homomomorphisms between the corresponding description trees. This result can also be obtained as a consequence of the fact that the containment problem $Q_1 \subseteq Q_2$ for conjunctive queries is polynomial if Q_2 is acyclic [26, 21]. Since it is easy to see that $\mathcal{EL}^{(n)}$-concept descriptions can be expressed by acyclic conjunctive queries with disequations [18], one might conjecture that polynomiality of subsumption in $\mathcal{EL}^{(n)}$ follows from the corresponding result for acyclic conjunctive queries with disequations. This is not true, however. In fact, the containment problem for conjunctive queries becomes considerably harder if disequations (i.e., atoms of the form $x \neq y$ for variables x, y) are allowed to occur in the conjunctive queries. For general conjunctive queries with disequations, the containment problem is Π_2^p-complete rather than NP-complete as in the case of conjunctive queries without disequations. Surprisingly, the problem remains Π_2^p-complete if Q_2 is restricted to being acyclic [18]. And even if both queries contain only disequations (and no database predicates), it is not hard to show by a reduction of the complement of the graph homomorphism problem that the containment problem is coNP-hard. Thus, the polynomiality result shown in the present paper does *not* follow from known results for containment of conjunctive queries with disequations.

In [9], it was shown that subsumption in $\mathcal{EL}$ remains polynomial even in the presence of GCIs, and this result was recently extended to a DL extending $\mathcal{EL}$ by several other interesting constructors [2]. Unfortunately, the results in [2] imply that subsumption in $\mathcal{EL}^{(n)}$ becomes ExpTime-hard in the presence of GCIs.

The most interesting topics for future research are, on the one hand, to show that the exponential translation from $\mathcal{EL}^{(n)}\mathcal{C}$ into $\mathcal{ALCQ}$ given in Section 3 is optimal, i.e., to prove that there is no polynomial translation. On the other hand, the exact complexity of subsumption between *unrestricted* $\mathcal{EL}^{(n)}$-concept descriptions is not yet known. The best complexity upper-bound that we currently have is coNP (see Corollary 6.4). We conjecture that the problem is coNP-hard, but have not yet found an appropriate reduction from a coNP-complete problem.

References

[1] Franz Baader. Terminological cycles in a description logic with existential restrictions. In *Proc. 18th Int. Joint Conf. on Artificial Intelligence*, 2003.

[2] Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$-envelope. In *Proc. 19th Int. Joint Conf. on Artificial Intelligence*, 2005. To appear.

[3] 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.

[4] Franz Baader and Philipp Hanschke. Extensions of concept languages for a mechanical engineering application. In *Proc. 16th German Workshop on Artificial Intelligence*, volume 671 of *LNCS*, 1992. Springer-Verlag.

[5] Franz Baader, Ralf Küsters, and Ralf Molitor. Computing least common subsumers in description logics with existential restrictions. In *Proc. 16th Int. Joint Conf. on Artificial Intelligence*, 1999.

[6] Franz Baader, Carsten Lutz, Eldar Karabaev,and Manfred Theißen. A new n-ary existential quantifier in description logics LTCS-Report 05-08, Theoretical Computer Science, TU Dresden, Germany, 2005. See http://lat.inf.tu-dresden.de/research/reports.html.

[7] Franz Baader, Maja Milicic, Carsten Lutz, Ulrike Sattler, and Frank Wolter. Integrating description logics and action formalisms for reasoning about web services LTCS-Report 05-02, Theoretical Computer Science, TU Dresden, Germany, 2005. See http://lat.inf.tu-dresden.de/research/reports.html.

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

[9] Sebastian Brandt. Polynomial time reasoning in a description logic with existential restrictions, GCI axioms, and—what else? In *Proc. 16th Eur. Conf. on Artificial Intelligence*, 2004.

[10] Diego Calvanese, Maurizio Lenzerini, and Daniele Nardi. A unified framework for class based representation formalisms. In *Proc. 4th Int. Conf. on the Principles of Knowledge Representation and Reasoning*, 1994.

[11] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press, 1990.

[12] Francesco Donini. Complexity of reasoning. In *[3]*. 2003.

[13] Michael R. Garey and David S. Johnson. *Computers and Intractability — A guide to NP-completeness*. W. H. Freeman and Company, San Francisco, 1979.

[14] Volker Haarslev and Ralf Möller. RACER system description. In *Proc. Int. Joint Conf. on Automated Reasoning*, 2001.

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

[16] Ian Horrocks. Using an expressive description logic: FaCT or fiction? In *Proc. 6th Int. Conf. on Principles of Knowledge Representation and Reasoning*, 1998.

[17] Ian Horrocks, Ulrike Sattler, and Stefan Tobies. Practical reasoning for very expressive description logics. *J. of the Interest Group in Pure and Applied Logic*, 8(3):239–264, 2000.

[18] Phokion G. Kolaitis, David M. Martin, and Madhukar N. Thakur. On the complexity of the containment problem for conjunctive queries with built-in predicates. In *Proc. 17th ACM Symp. on Principles of Database Systems*, 1998.

[19] Carsten Lutz. Complexity of terminological reasoning revisited. In *Proc. 6th Int. Conf. on Logic for Programming and Automated Reasoning*, volume 1705 of *LNAI*. Springer-Verlag, 1999.

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

[21] Xiaolei Qian. Query folding. In *Proc. 12th IEEE Int. Conf. on Data Engineering*, 1996.

[22] Steven W. Reyner. An analysis of a good algorithm for the subtree problem. *SIAM J. on Computing*, 6(4):730–732, 1977.

[23] Ulrike Sattler. A concept language extended with different kinds of transitive roles. In *Proc. 20th German Annual Conf. on Artificial Intelligence*, volume 1137 of *LNAI*. Springer-Verlag, 1996.

[24] Manfred Theißen and Lars von Wedel. The need for an n-ary existential quantifier in description logics. In *Proc. KI-04 Workshop on Applications of Description Logics*. CEUR Electronic Workshop Proceedings, http://CEUR-WS.org/Vol-115/, 2004.

[25] Stephan Tobies. *Complexity Results and Practical Algorithms for Logics in Knowledge Representation*. PhD thesis, Computer Science Department, RWTH Aachen, Germany, 2001.

[26] Mihalis Yannakakis. Algorithms for acyclic database schemes. In *Proc. 7th Int. Conf. on Very Large Data Bases*, 1981.