LTCS–Report

Aachen University of Technology
Research group for
Theoretical Computer Science

Computing Least Common Subsumer in Description Logics with Existential Restrictions

Franz Baader and Ralf Küsters and Ralf Molitor

LTCS-Report 98-09

An abridged version of this technical report is published in the Proceedings of IJCAI'99.

RWTH Aachen
LuFg Theoretische Informatik
http://www-lti.informatik.rwth-aachen.de

Ahornstr. 55
52074 Aachen
Germany

Computing Least Common Subsumer in Description Logics with Existential Restrictions

Franz Baader and Ralf Küsters and Ralf Molitor
LuFg Theoretical Computer Science, RWTH Aachen
email: {baader,kuesters,molitor}@informatik.rwth-aachen.de

Abstract

Computing the least common subsumer (lcs) is an inference task that can be used to support the "bottom-up" construction of knowledge bases for KR systems based on description logics. Previous work on how to compute the lcs has concentrated on description logics that allow for universal value restrictions, but not for existential restrictions. The main new contribution of this paper is the treatment of description logics with existential restrictions. More precisely, we show that, for the description logic $\mathcal{ALE}$ (which allows for conjunction, universal value restrictions, existential restrictions, negation of atomic concepts, as well as the top and the bottom concept), the lcs always exists and can effectively be computed.

Our approach for computing the lcs is based on an appropriate representation of concept descriptions by certain trees, and a characterization of subsumption by homomorphisms between these trees. The lcs operation then corresponds to the product operation on trees.

1 Introduction

Knowledge representation systems based on description logics (DL) can be used to describe the knowledge of an application domain in a structured and formally well-understood way. Traditionally, the knowledge base of a DL system is built in a "top-down" fashion by first formalizing the relevant concepts of the domain (its terminology) by *concept descriptions*, i.e., expressions that are built from atomic concepts (unary predicates) and atomic roles (binary predicates) using the concept constructors provided by the DL language. In a second step, the concept descriptions are used to specify properties of objects and individuals occurring in the domain. DL systems provide their users with inference services that support both steps: classification of concepts and of individuals. Classification of concepts determines subconcept/superconcept relationships (called subsumption relationships) between the concepts of a given terminology, and thus allows one to structure the terminology in the form of a subsumption hierarchy. Classification of individuals (or objects) determines whether a given individual is always an instance of a certain concept (i.e., whether this instance relationship is implied by the descriptions of the individual and the concept).

This traditional "top-down" approach for constructing a DL knowledge base is not always adequate, however. On the one hand, it need not be clear from the outset which are the relevant concepts in a particular application. On the other hand, even if it is clear which (intuitive) concepts should be introduced, it is in general not easy to come up with formal definitions of these concepts within the available description language. For example, in one of our applications in chemical process engineering [16, 5], the process engineers prefer to construct the knowledge base (which consists of descriptions of standard building blocks of process models, such as reactors) in the following "bottom-up" fashion: first, they introduce several "typical" examples of the standard building block as individuals, and then they generalize (the descriptions of) these individuals into a concept description that (i) has all the individuals as instances, and (ii) is the most specific description satisfying property (i). The task of computing a description satisfying (i) and (ii) can be split into two subtasks: computing the most specific concept of a single individual, and computing the least common subsumer of a given finite number of concepts. The *most specific concept* (msc) of an individual b (the *least common subsumer* (lcs) of concept descriptions $C_1, \ldots, C_n$) is the most specific concept description C (expressible in the given description language) that has b as an instance (that subsumes $C_1, \ldots, C_n$).

The present paper investigates the second subtask for the sub-language $\mathcal{ALE}$ of the DL employed in our process engineering application.[1] This language allows both for value restrictions and existential restrictions, but not for full negation and disjunction (since the lcs operation is trivial in the presence of disjunction, and thus does not provide useful information). It can, e.g., be used to introduce the concept of a reactor with cooling jacket by the description $\mathsf{Reactor} \sqcap \exists \mathsf{connected\text{-}to.Cooling\text{-}Jacket} \sqcap \forall \mathsf{functionality}.\neg \mathsf{Vaporize}$, where $\mathsf{Vaporize}$ is a primitive concept (i.e., not further defined).

Previous work on how to compute the lcs [10, 11, 13] has concentrated on sub-languages of the DL used by the system CLASSIC [7], which allows (among other constructors) for value restrictions, but not for existential restrictions. Thus, the main new contribution of the present paper is the treatment of existential restrictions.

For didactic reasons, we will start with showing how to compute the lcs in the small language $\mathcal{EL}$, which allows for conjunction and existential restrictions only, and then extend our treatment in two steps to $\mathcal{FLE}$ by adding value restrictions, and then to $\mathcal{ALE}$ by further adding primitive negation. For all three languages, we proceed in the following manner. First, we introduce an appropriate data structure for representing concept descriptions (so-called description trees), and show that subsumption can be characterized by the existence of homomorphisms between description trees. From this characterization we then deduce that the lcs operation on concept descriptions corresponds to the product operation on description trees, which can easily be computed. We will also comment on the

[1] See [10] for a description of other contexts in which the lcs operation is useful.

Construct name	Syntax	Semantics
primitive concept $P \in N_C$	P	$P^I \subseteq \Delta$
top-concept	$\top$	Δ
conjunction	$C \sqcap D$	$C^I \cap D^I$
existential restr. for $r \in N_R$	$\exists r.C$	$\{x \in \Delta \mid \exists y : (x,y) \in r^I \wedge y \in C^I\}$
value restr. for $r \in N_R$	$\forall r.C$	$\{x \in \Delta \mid \forall y : (x,y) \in r^I \rightarrow y \in C^I\}$
primitive negation, $P \in N_C$	$\neg P$	$\Delta \setminus P^I$
bottom-concept	$\bot$	$\emptyset$

Table 1: Syntax and semantics of concept descriptions.

complexity of subsumption and the lcs for the languages under consideration.

2 Preliminaries

Concept descriptions are inductively defined with the help of a set of *constructors*, starting with a set N_C of *primitive concepts* and a set N_R of *primitive roles*. The constructors determine the expressive power of the DL. In this work, we consider concept descriptions built from the constructors shown in Table 1. In the description logic $\mathcal{EL}$, concept descriptions are formed using the constructors top-concept ($\top$), conjunction ($C \sqcap D$) and existential restriction ($\exists r.C$). The description logic $\mathcal{FLE}$ additionally provides us with value restrictions ($\forall r.C$), and $\mathcal{ALE}$ allows for all the constructors shown in Table 1.

The semantics of a concept description is defined in terms of an *interpretation* $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$. The domain Δ of $\mathcal{I}$ is a non-empty set of individuals and the interpretation function $\cdot^{\mathcal{I}}$ maps each primitive concept $P \in N_C$ to a set $P^I \subseteq \Delta$ and each primitive role $r \in N_R$ to a binary relation $r^I \subseteq \Delta \times \Delta$. The extension of $\cdot^{\mathcal{I}}$ to arbitrary concept descriptions is inductively defined, as shown in the third column of Table 1.

One of the most important traditional inference services provided by DL systems is computing the subsumption hierarchy.

Definition 1 (Subsumption) *Let C, D be concept descriptions.*
D subsumes C *(for short $C \sqsubseteq D$) iff $C^I \subseteq D^I$ for all interpretations $\mathcal{I}$.*
C *is* equivalent *to D (for short $C \equiv D$) iff $C \sqsubseteq D$ and $D \sqsubseteq C$, i.e., $C^I = D^I$ for all interpretations $\mathcal{I}$.*

In this paper, we are interested in the non-standard inference task of computing the *least common subsumer* (lcs) of concept descriptions.

Definition 2 (Least Common Subsumer) *Let $C_1, \ldots, C_n$ and C be concept descriptions in a DL $\mathcal{L}$. The concept description C is a* least common subsumer *(lcs) of $C_1, \ldots, C_n$ (for short $C = \mathsf{lcs}(C_1, \ldots, C_n)$) iff*

1. $C_i \sqsubseteq C$ for all $1 \leq i \leq n$, and

$$\begin{tikzcd}
& E' \arrow[d, dashed, "! \sqsubseteq"] \arrow[ddl, bend right, "\sqsubseteq"'] \arrow[ddr, bend left, "\sqsubseteq"] & \\
& E \arrow[dl, "\sqsubseteq"'] \arrow[dr, "\sqsubseteq"] & \\
C & & D
\end{tikzcd}$$

Figure 1: Conditions for the lcs of two concept descriptions.

2. *C is the least concept description with this property, i.e., if C' is a concept description satisfying $C_i \sqsubseteq C'$ for all $1 \leq i \leq n$, then $C \sqsubseteq C'$.*

Depending on the DL under consideration, the lcs of two or more descriptions need not always exists. For example, if we only allow for concept descriptions built from primitive concepts and conjunction, then the lcs of two different primitive concepts P, Q does not exist in this language, because $\mathsf{lcs}(P, Q) = \top$. However, if for the DL under consideration the lcs always exists, then it is unique up to equivalence.

In the following sections, we will show that, for the DLs $\mathcal{EL}$, $\mathcal{FLE}$, and $\mathcal{ALE}$, the lcs always exists and can effectively be computed. We will mostly restrict the attention to the problem of computing the lcs of two concept descriptions, since the lcs of $n > 2$ descriptions can be obtained by iterated application of the binary lcs operation.

Least common subsumer and products

It should be noted that there is a tight relationship between least common subsumer and the notion of a *product* as known from category theory [14].

Let us consider the category in which the objects are concept descriptions and where there exists a morphism from an object C to an object D iff C subsumes D.

Now let C, D be two concept descriptions. According to Definition 2 the lcs E of C and D must satisfy the following conditions:

1. $C \sqsubseteq E$ and $D \sqsubseteq E$, and

2. for all E' with $C \sqsubseteq E'$ and $D \sqsubseteq E'$ it holds that $E \sqsubseteq E'$.

A graphical depiction of these conditions is shown in Figure 1. The resulting diagram is also known as *product diagram* in category theory [14].

In other words, the lcs of two concept descriptions is the same as the product of these objects in the corresponding category. Since computing the product in a category is an associative and commutative operation, the lcs of n concept descriptions can be computed by an iterated application of the binary lcs operation.

3 Getting started – the LCS for $\mathcal{EL}$

As mentioned in the introduction, our results on computing the lcs in DLs with existential restrictions are based on a representation of concept descriptions by so-called description trees and an appropriate characterization of subsumption by homomorphisms between these trees.

3.1 $\mathcal{EL}$-description trees

In the case of the very small DL $\mathcal{EL}$, concept descriptions are represented by $\mathcal{EL}$-description trees.

Definition 3 ($\mathcal{EL}$-description trees) *An* $\mathcal{EL}$-description tree *is a tree of the form* $\mathcal{G} = (V, E, v_0, \ell)$ *with root* v_0 *where*

- *the edges* $vrw \in E$ *are labeled with primitive roles* r *from* N_R*, and*
- *the nodes* $v \in V$ *are labeled with sets* $\ell(v)$ *of primitive concepts from* N_C*.*

The empty label corresponds to the top-concept.

In the sequel, $\mathcal{G}(v)$ where v is a node in $\mathcal{G}$ denotes the subtree of $\mathcal{G}$ with root v. Furthermore, the size $|\mathcal{G}|$ of $\mathcal{G}$ is defined by $|V| + |E| + \Sigma_{v \in V}|l(v)|$.

Intuitively, such an $\mathcal{EL}$-description tree is merely a graphical representation of the syntax of the concept description. In order to translate concept description into description trees, we need the following notion. The *role depth* of an ($\mathcal{EL}$-, $\mathcal{FLE}$-, $\mathcal{ALE}$-) concept description C (for short $\mathsf{depth}(C)$) is inductively defined by

- $depth(\bot) = depth(\top) = depth(P) = depth(\neg P) := 0$;
- $depth(C \sqcap D) := max(depth(C), depth(D))$;
- $depth(\forall r.C) = depth(\exists r.C) := depth(C) + 1$.

Every $\mathcal{EL}$-concept description C can be written (modulo equivalence) as $C \equiv P_1 \sqcap \ldots \sqcap P_n \sqcap \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_m.C_m$ with $P_i \in N_C \cup \{\top\}$. Such a concept description can now be translated into an $\mathcal{EL}$-description tree $\mathcal{G}_C = (V, E, v_0, \ell)$ by induction on the role depth of C as follows.

If $\mathsf{depth}(C) = 0$ **then** $V := \{v_0\}$, $E := \emptyset$, and $\ell(v_0) := \{P_1, \ldots, P_n\} \setminus \{\top\}$.

If $\mathsf{depth}(C) > 0$ **then** for $1 \leq i \leq m$, let $\mathcal{G}_i = (V_i, E_i, v_{0i}, \ell_i)$ be the inductively defined description tree corresponding to C_i where, w.l.o.g., the V_i are pairwise disjoint. Then

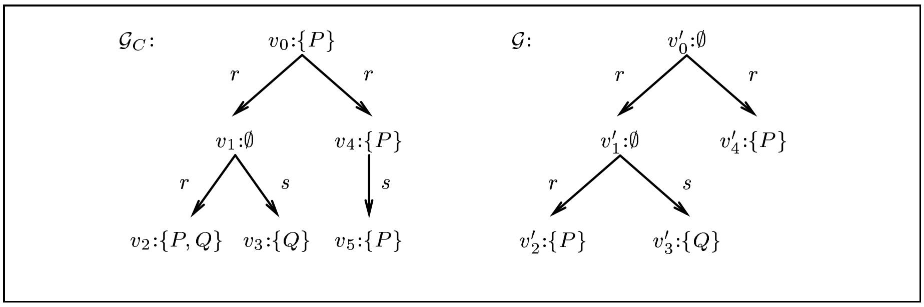

Figure 2: $\mathcal{EL}$-description trees.

- $V := \{v_0\} \cup \bigcup\limits_{1\le i\le m} V_i$,
- $E := \{v_0 r_i v_{0i} \mid 1 \le i \le m\} \cup \bigcup\limits_{1\le i\le m} E_i$,
- $\ell(v) := \begin{cases} \{P_1,\dots,P_n\} \setminus \{\top\}, & v = v_0 \\ \ell_i(v), & v \in V_i, 1 \le i \le m \end{cases}$

Example 4 The $\mathcal{EL}$-concept description

$$C := P \sqcap \exists r.(\exists r.(P \sqcap Q) \sqcap \exists s.Q) \sqcap \exists r.(P \sqcap \exists s.P)$$

yields the tree $\mathcal{G}_C$ depicted on the left-hand side of Figure 2.

Conversely, every $\mathcal{EL}$-description tree $\mathcal{G} = (V, E, v_0, \ell)$ can be translated into an $\mathcal{EL}$-concept description $C_\mathcal{G}$ by induction on the depth[2] of $\mathcal{G}$.

If $\mathsf{depth}(\mathcal{G}) = 0$ **then** $V = \{v_0\}$, $E = \emptyset$. If $\ell(v_0) = \emptyset$, then $C_\mathcal{G} := \top$; otherwise, we have $\ell(v_0) = \{P_1, \dots, P_n\}$, $n \ge 1$ and define $C_\mathcal{G} := P_1 \sqcap \dots \sqcap P_n$.

If $\mathsf{depth}(\mathcal{G}) > 0$ **then** let $\ell(v_0) = \{P_1, \dots, P_n\}$, $n \ge 0$, and let $\{v_1, \dots, v_m\}$ be the set of all successors of v_0 with $v_0 r_i v_i \in E$ for some $r_i \in N_R$, $1 \le i \le m$. Further, let $C_1, \dots, C_m$ denote the inductively defined $\mathcal{EL}$-concept descriptions corresponding to the subtrees of $\mathcal{G}$ with root v_i, $1 \le i \le m$. We define $C_\mathcal{G} := P_1 \sqcap \dots \sqcap P_n \sqcap \exists r_1.C_1 \sqcap \dots \sqcap \exists r_m.C_m$.

Note that only for a leaf $v \in V$ the empty label is translated into the top-concept. For a node $v \in V$ with $\ell(v) = \emptyset$ that is not a leaf, the empty set is not translated into the top-concept: the concept description corresponding to the subtree with root v only consists of existential restrictions induced by the direct successors of v.

[2] The *depth* of a description tree $\mathcal{G}$ is defined as the length of the longest path in $\mathcal{G}$. Since it directly corresponds to the depth of the corresponding concept description, it is also denoted by $\mathsf{depth}(\mathcal{G})$.

Example 5 The $\mathcal{EL}$-description tree $\mathcal{G}$ in Figure 2 yields the $\mathcal{EL}$-concept description

$$C_{\mathcal{G}} = \exists r.(\exists r.P \sqcap \exists s.Q) \sqcap \exists r.P.$$

The *semantics* of an $\mathcal{EL}$-description tree $\mathcal{G}$ is given by the semantics of the corresponding $\mathcal{EL}$-concept description, i.e., for an interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ we define $\mathcal{G}^{\mathcal{I}} := C_{\mathcal{G}}^{\mathcal{I}}$.

The translations of $\mathcal{EL}$-concept descriptions and $\mathcal{EL}$-description trees into one another preserve the semantics of concept descriptions in the sense that $C \equiv C_{\mathcal{G}_C}$.

3.2 Subsumption in $\mathcal{EL}$

In order to achieve a characterization of subsumption in $\mathcal{EL}$ that allows for characterizing the lcs by products of description trees, we need the following notion.

Definition 6 (Homomorphisms on $\mathcal{EL}$-description trees)
Let $\mathcal{G} = (V_G, E_G, v_0, \ell_G)$ and $\mathcal{H} = (V_H, E_H, w_0, \ell_H)$ be $\mathcal{EL}$-description trees. A mapping $\varphi : V_H \longrightarrow V_G$ is a homomorphism *from $\mathcal{H}$ to $\mathcal{G}$ iff the following conditions are satisfied:*

1. *$\varphi(w_0) = v_0$,*
2. *$\ell_H(v) \subseteq \ell_G(\varphi(v))$ for all $v \in V_H$, and*
3. *$\varphi(v)r\varphi(w) \in E_G$ for all $vrw \in E_H$.*

Now, subsumption between $\mathcal{EL}$-concept descriptions can be characterized in terms of homomorphisms between $\mathcal{EL}$-description trees.

Theorem 7 *Let C, D be $\mathcal{EL}$-concept descriptions and $\mathcal{G}_C, \mathcal{G}_D$ the corresponding description trees. Then $C \sqsubseteq D$ iff there exists a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$.*

We will prove a generalization of this theorem in Section 4.1.

Example 8 (Example 4 continued)
Consider the $\mathcal{EL}$-description trees depicted in Figure 2. We have $C \sqsubseteq C_{\mathcal{G}}$, because mapping v_i' onto v_i for $0 \leq i \leq 4$ yields a homomorphism from $\mathcal{G}$ to $\mathcal{G}_C$.

On the complexity of subsumption in $\mathcal{EL}$

For two $\mathcal{EL}$-concept descriptions C, D, subsumption $C \sqsubseteq D$ can be decided by (1) translating C, D into their corresponding $\mathcal{EL}$-description trees $\mathcal{G}_C, \mathcal{G}_D$ and (2) testing wether there exists a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$. In [15], a polynomial-time algorithm is introduced deciding wether there exists a homomorphism from a tree onto another tree. In [4] we have shown, that even for the DL $\mathcal{ELIRO}$[1] (which extends $\mathcal{EL}$ by *inverse roles*, *conjunction of roles*,

```
Input: Two $\mathcal{EL}$-description trees $\mathcal{H}$ and $\mathcal{G}$ normal form.
Output: "yes", if there exists a homomorphism from $\mathcal{H}$ to $\mathcal{G}$, "no", otherwise.
Let $\mathcal{H} = (V_H, E_H, w_0, \ell_H)$ and $\mathcal{G} = (V_G, E_G, v_0, \ell_G)$. Further, let
$\{v_1, \ldots, v_n\}$ be a post-order sequence of $V_H$, i.e., $v_1$ is a leaf and $v_n = w_0$.
Define a labeling $\delta : V_G \longrightarrow \mathcal{P}(V_H)$ as follows.
Initialize $\delta$ by $\delta(w) := \emptyset$ for all $w \in V_G$.
For $1 \leq i \leq n$ do
   For all $w \in V_G$ do
     If $\ell_H(v_i) \subseteq \ell_G(w)$ and
       for all $v_i r v \in E_H$ there is $w' \in V_G$ such that
         $v \in \delta(w')$ and $wrw' \in E_G$
     Then $\delta(w) := \delta(w) \cup \{v_i\}$
   od
od
If $w_0 \in \delta(v_0)$, then return "yes", else return "no".
```

Figure 3: Homomorphisms between $\mathcal{EL}$-description trees.

and *constants*) subsumption is tractable. For the DLs $\mathcal{FLE}$ and $\mathcal{ALE}$, however, subsumption is an NP-complete problem [12].

For the readers convenience, we now give a brief description of an algorithm deciding wether there exists a homomorphism from an $\mathcal{EL}$-description tree $\mathcal{H} = (V_H, E_H, w_0, \ell_H)$ onto an $\mathcal{EL}$-description tree $\mathcal{G} = (V_G, E_G, v_0, \ell_G)$. The idea behind is as follows. The algorithm defines a mapping $\delta : V_G \longrightarrow \mathcal{P}(V_H)$ that labels each node $v \in V_G$ with a set of nodes from V_H by once traversing the tree $\mathcal{H}$ from its leaves to its root w_0. If $w_0 \in \delta(v_0)$, then the algorithm answers "yes"; otherwise "no". If the algorithm answers "yes", then the mapping δ induces a homomorphism from $\mathcal{H}$ to $\mathcal{G}$.

The algorithm shown in Figure 3 is a restriction of the algorithm introduced in [4] deciding subsumption in $\mathcal{ELIRO}^1$. The algorithm terminates in time polynomial in the size of $\mathcal{G}$ and $\mathcal{H}$ [4]. In the restricted case of $\mathcal{EL}$-description trees, the mapping δ has the following properties: whenever we have $w \in \delta(v)$, then it holds that

- $\ell_H(w) \subseteq \ell_G(v)$ and
- for each r-successor w' of w in $\mathcal{H}$, there exists an r-successor v' of v in $\mathcal{G}$ with $w' \in \delta(v')$.

The proof of soundness and completeness of the algorithm makes heavy use of these properties. A full proof can be adapted from the proof of soundness and completeness for the extended algorithm given in [4].

A comparison with conjunctive queries and conceptual graphs

Theorem 7 is a special case of the characterization of subsumption between simple conceptual graphs [9], and of the characterization of containment of conjunctive queries [1].

Simple conceptual graphs

In [4] we have investigated the relation between DLs and the simple conceptual graphs as introduced in [17], Assumption 3.1.2, page 73 (see also [8]). We have shown that there is a 1-1 correspondence between the DL $\mathcal{ELIRO}^1$ and the class of simple conceptual graphs that are trees. More precisely, we have shown that $\mathcal{ELIRO}^1$-concept descriptions can be translated into simple conceptual graphs that are trees, and vice versa. Thereby, equivalence is defined w.r.t. FO formulae corresponding to simple conceptual graphs and $\mathcal{ELIRO}^1$-concept descriptions, respectively. Due to this equivalence, subsumption in $\mathcal{ELIRO}^1$ can be reduced to subsumption between simple conceptual graphs. Subsumption between simple conceptual graphs $G \sqsubseteq H$ can be characterized by the existence of a homomorphism from the subsumer H to the subsumee G where G must be in some normal form [9, 8]. As a consequence, for two $\mathcal{ELIRO}^1$-concept descriptions C, D, $C \sqsubseteq D$ can be decided by first translating C and D into their corresponding simple conceptual graphs G_C and G_D and then testing wether there exists a homomorphism from G_D to the normal form of G_C. In the more general setting of simple conceptual graphs, testing for such a homomorphism is an NP-complete problem. In the restricted case of $\mathcal{ELIRO}^1$, however, testing for a homomorphism is a tractable problem (see the previous paragraph for a description of an algorithm testing for the existence of a homomorphism between two $\mathcal{EL}$-description trees).

In [8] the authors pointed out that there is a 1-1 correspondence between simple conceptual graphs and conjunctive queries. Thus, subsumption in $\mathcal{EL}$ is also a special case of containment between conjunctive queries.

Conjunctive queries

A conjunctive query q is defined as a finite conjunction of atomic formulae of the form $R(x_1, \ldots, x_n)$ where x_i might be variables and constants [1]. Given a set $\{y_1, \ldots, y_m\}$ of distinguished variables occurring in q and an interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$, the semantics of q is defined as $\{(\delta_1, \ldots, \delta_m) \in \Delta^m \mid \mathcal{I} \models q[y_1/\delta_1, \ldots, y_m/\delta_m]\}$, where the free variables occurring in q are existentially quantified.

Now, every $\mathcal{EL}$-concept description C can be translated into an equivalent conjunctive query q_C where q_C contains no constants, only unary and binary predicates, and exactly one distinguished variable x_0. For example, we obtain the following conjunctive query q_C for the $\mathcal{EL}$-concept description C introduced in Example 4:

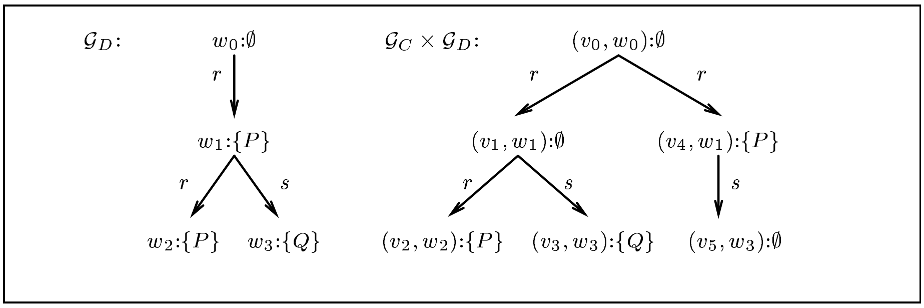

Figure 4: The product of $\mathcal{EL}$-description trees.

$$\begin{aligned}&P(x_0)\wedge\\&r(x_0,x_1)\wedge r(x_1,x_2)\wedge P(x_2)\wedge Q(x_2)\wedge s(x_1,x_3)\wedge Q(x_3)\\&\wedge r(x_0,x_4)\wedge P(x_4)\wedge s(x_4,x_5)\wedge P(x_5),\end{aligned}$$

where q_C is equivalent to C if we assume x_0 to be the single distinguished variable in q_C.

Containment of conjunctive queries $p \sqsubseteq q$ can be characterized in terms of homomorphisms between the variables and constants occurring in p and q, respectively [1]. In the restricted case of conjunctive queries corresponding to $\mathcal{EL}$-concept descriptions, the definition of homomorphisms between conjunctive queries coincides with the one given in Definition 6, and Theorem 7 is a direct consequence of Theorem 6.2.3 in [1].

3.3 Least common subsumer in $\mathcal{EL}$

The product of description trees is defined by induction on the depth of the trees.

Definition 9 (Product of $\mathcal{EL}$-description trees) *Let $\mathcal{G} = (V_G, E_G, v_0, \ell_G)$, $\mathcal{H} = (V_H, E_H, w_0, \ell_H)$ be $\mathcal{EL}$-description trees. Further, let $\mathcal{G}(v)$ denote the subtree of $\mathcal{G}$ with root v. The* product $\mathcal{G} \times \mathcal{H} = (V, E, (v_0, w_0), \ell)$ of $\mathcal{G}$ and $\mathcal{H}$ *is defined as follows. We define (v_0, w_0) to be the root of $\mathcal{G} \times \mathcal{H}$, labeled with $\ell_G(v_0) \cap \ell_H(w_0)$. For each r-successor v of v_0 in $\mathcal{G}$ and w of w_0 in $\mathcal{H}$, we obtain an r-successor (v, w) of (v_0, w_0) in $\mathcal{G} \times \mathcal{H}$ that is the root of the inductively defined product of $\mathcal{G}(v)$ and $\mathcal{H}(w)$.*

Example 10 (Example 4 continued)
Consider the $\mathcal{EL}$-description tree $\mathcal{G}_C$ depicted in Figure 2 and the $\mathcal{EL}$-description tree $\mathcal{G}_D$ depicted on the left hand side of Figure 4, where $\mathcal{G}_D$ corresponds to the $\mathcal{EL}$-concept description $D := \exists r.(P \sqcap \exists r.P \sqcap \exists s.Q)$. The product $\mathcal{G}_C \times \mathcal{G}_D$ is depicted on the right hand side of Figure 4.

We are now equipped to formalize the characterization of the lcs of two $\mathcal{EL}$-concept descriptions by the product of $\mathcal{EL}$-description trees.

Theorem 11 *Let C, D be $\mathcal{EL}$-concept descriptions and $\mathcal{G}_C, \mathcal{G}_D$ the corresponding description trees. Then, $C_{\mathcal{G}_C \times \mathcal{G}_D}$ is the lcs of C and D.*[3]

Proof: We have to show that

1. $C \sqsubseteq C_{\mathcal{G}_C \times \mathcal{G}_D}$,

2. $D \sqsubseteq C_{\mathcal{G}_C \times \mathcal{G}_D}$, and

3. for each C' with $C \sqsubseteq C'$ and $D \sqsubseteq C'$, we have $C_{\mathcal{G}_C \times \mathcal{G}_D} \sqsubseteq C'$.

The projection π_i with $\pi_i(v_1, v_2) := v_i$, $i = 1, 2$, yields a homomorphism from $\mathcal{G}_C \times \mathcal{G}_D = (V, E, (v_0, w_0), \ell)$ to $\mathcal{G}_C$ for $i = 1$ and to $\mathcal{G}_D$ for $i = 2$. By Theorem 7 this means that $C \sqsubseteq C_{\mathcal{G}_C \times \mathcal{G}_D}$ and $D \sqsubseteq C_{\mathcal{G}_C \times \mathcal{G}_D}$.

Now let C' be an arbitrary common subsumer of C and D and $\mathcal{G}' = (V', E', v_0', \ell')$ the corresponding $\mathcal{EL}$-description tree. Theorem 7 yields homomorphisms φ_1 from $\mathcal{G}_{C'}$ to $\mathcal{G}_C$ and φ_2 from $\mathcal{G}_{C'}$ to $\mathcal{G}_D$. Define a mapping $\varphi :=< \varphi_1, \varphi_2 >$ from $\mathcal{G}_{C'}$ to $\mathcal{G}_C \times \mathcal{G}_D$ as the product of φ_1 and φ_2, i.e., $\varphi(v') := (\varphi_1(v'), \varphi_2(v'))$ for all $v' \in V'$. We prove that (1) φ is well-defined, i.e., $\varphi(v') \in V$ for all $v' \in V'$, and (2) φ is a homomorphism according to Definition 6 from $\mathcal{G}_{C'}$ to $\mathcal{G}_C \times \mathcal{G}_D$.

The first point is shown by induction on the length $\delta(v')$ of the path in $\mathcal{G}_{C'}$ leading from v_0' to v'.[4]

$\delta(v') = 0$**:** Then we have $v' = v_0'$ and hence, $\varphi(v_0') = (\varphi_1(v_0'), \varphi_2(v_0')) = (v_0, w_0) \in V$.

$\delta(v') > 0$**:** Let $v' \in V'$ with $\delta(v') > 0$. Then, since $\mathcal{G}_{C'}$ is a tree, there exists a unique predecessor $v'' \in V'$ of v', i.e., $v''rv' \in E'$ for some $r \in N_R$ or $v''\forall rv' \in E'$ for some $r \in N_R$. Assume $v''rv' \in E'$ for some $r \in N_R$. (The case $v''\forall rv' \in E'$ can be handled in the same way.) Obviously, we have $\delta(v'') = \delta(v') - 1$. By induction, we know $(\varphi_1(v''), \varphi_2(v'')) \in V$. Since φ_i, $i = 1, 2$, are homomorphisms, we have $\varphi_1(v'')r\varphi_1(v') \in E_C$ and $\varphi_2(v'')r\varphi_2(v') \in E_D$. Definition 9 yields $(\varphi_1(v'), \varphi_2(v'))$ as an r-successor of $(\varphi_1(v''), \varphi_2(v''))$ in $\mathcal{G}_C \times \mathcal{G}_D$ and hence, $(\varphi_1(v'), \varphi_2(v')) \in V$.

Now, the proof of (2) is rather simple.

1. We have $\varphi(v_0') = (\varphi_1(v_0'), \varphi_2(v_0')) = (v_0, w_0)$, because φ_1 (φ_2) is a homomorphism from $\mathcal{G}_{C'}$ to $\mathcal{G}_C$ ($\mathcal{G}_D$).

2. Since $\ell'(v') \subseteq \ell_C(\varphi_1(v'))$ and $\ell'(v') \subseteq \ell_D(\varphi_2(v'))$ for all $v' \in V'$, we have $\ell'(v') \subseteq \ell_C(\varphi_1(v')) \cap \ell_D(\varphi_2(v')) = \ell(\varphi_1(v'), \varphi_2(v'))$.

3. Let $v'rw' \in E'$. Then we have $\varphi_1(v')r\varphi_1(w') \in E_C$ and $\varphi_2(v')r\varphi_2(w') \in E_D$. Due to (1) we have $(\varphi_1(v'), \varphi_2(v')) \in V$ and then by Definition 9, it is $(\varphi_1(v'), \varphi_2(v'))r(\varphi_1(w'), \varphi_2(w')) \in E$.

[3] As already mentioned before, the lcs is by definition uniquely determined up to equivalence. Therefore, we say "the" lcs instead of "an" lcs.

[4] In $\mathcal{EL}$-description trees $\mathcal{G} = (V, E, v_0, \ell)$, for each node $v \in V$, there exists a *unique* path leading from the root v_0 to v. Hence, $\delta(v)$ is well-defined.

Now, Theorem 7 implies $C_{\mathcal{G}_C \times \mathcal{G}_D} \sqsubseteq C'$. This completes the proof of Theorem 11. □

Example 12 (Example 10 continued) For the concept descriptions C and D from Example 4 and Example 10, respectively, we obtain

$$\mathsf{lcs}(C, D) = C_{\mathcal{G}_C \times \mathcal{G}_D} = \exists r.(\exists r.P \sqcap \exists s.Q) \sqcap \exists r.(P \sqcap \exists s.\top).$$

On the complexity of least common subsumers in $\mathcal{EL}$

The size of the lcs of two $\mathcal{EL}$-concept descriptions C, D can be bounded by the size of $\mathcal{G}_C \times \mathcal{G}_D$, which is polynomial in the size of $\mathcal{G}_C$ and $\mathcal{G}_D$. Since the size of the description tree corresponding to a given description is linear in the size of the description, we obtain:

Proposition 13 *The size of the lcs of* two *$\mathcal{EL}$-concept descriptions C, D is polynomial in the size of C and D, and the lcs can be computed in polynomial time.*

In our process engineering application, however, we are interested in the lcs of $n > 2$ concept descriptions $C_1, \ldots, C_n$. This lcs can be obtained from the product $\mathcal{G}_{C_1} \times \cdots \times \mathcal{G}_{C_n}$ of their corresponding $\mathcal{EL}$-description trees. Therefore, the size of the lcs can be bounded by the size of this product. It has turned out that, even for the small DL $\mathcal{EL}$, this size cannot be polynomially bounded.

Example 14 We define for each $n \in \mathbb{N}$ a sequence $\{C_{1n}, \ldots, C_{nn}\}$ of $\mathcal{EL}$-concept descriptions such that each C_{in} has size linear in n and the lcs of $C_{1n}, \ldots, , C_{nn}$ has size exponential in n.

For $n \in \mathbb{N}$ and $1 < i < n$ let

- $D_i := \exists r.(P \sqcap Q \sqcap D_{i-1})$ and $D_1 := \exists r.(P \sqcap Q)$, and
- $C_{1n} := \exists r.(P \sqcap D_{n-1}) \sqcap \exists r.(Q \sqcap D_{n-1})$,
- $C_{in} := \underbrace{\exists r.(P \sqcap Q \sqcap \exists r.(\cdots \exists r.}_{\times(i-1)}(P \sqcap Q \sqcap \exists r.(P \sqcap D_{n-i}) \sqcap \exists r.(Q \sqcap D_{n-i}))\ \underbrace{)\cdots)}_{\times(i-1)}$,
- $C_{nn} := \underbrace{\exists r.(P \sqcap Q \sqcap \exists r.(\cdots \exists r.}_{\times(n-1)}(P \sqcap Q \exists r.P \sqcap \exists r.Q)\ \underbrace{)\cdots)}_{\times(n-1)}$.

Intuitively, the description tree corresponding to D_i is a single path of length i, each edge labeled r and each node (except the root) labeled with $\{P, Q\}$. Furthermore, the $\mathcal{EL}$-description tree $\mathcal{G}_{in} = (V_i, E_i, v_{0i}, \ell_i)$ corresponding to C_{in} is a tree of depth n of the following form:

1. the successors of the root v_{0i} reached via a path of length $j < i$ form a single path of length $i - 1$, where each node except the root is labeled with $\{P, Q\}$,

2. the $(i-1)$. r-successor has two r-successors, one labeled $\{P\}$, the other one labeled $\{Q\}$, and
3. each of these two successors is the root of a single path of length $n-i$, where again each node except the root is labeled with $\{P, Q\}$.

It is not hard to see, that each C_{in} is linear in the size of n.

As an example, the description trees corresponding to C_{13}, C_{23}, C_{33} are depicted in Figure 5. By Theorem 11, the description tree $\mathcal{G}_{13} \times \mathcal{G}_{23} \times \mathcal{G}_{33}$ corresponds to the lcs of C_{13}, C_{23}, C_{33}. The description tree $\mathcal{G}_{13} \times \mathcal{G}_{23} \times \mathcal{G}_{33}$ is depicted in Figure 6 where the nodes are called y_i instead of (v_i, w_j, x_k). Obviously, this is a full binary tree of depth 3.

The product $\mathcal{G}_{1n} \times \ldots \times \mathcal{G}_{nn}$ is isomorphic to a full binary tree $\mathcal{B}_n = (V_n, E_n, w_0, \ell_n)$ of depth n of the following form:

- each edge is labeled with r,
- the root is labeled with the empty set, and
- each node $v \in V_n$ that is not a leaf has exactly two r-successors, one labeled with $\{P\}$ and one labeled with $\{Q\}$.

Since a full binary tree of depth n has exponential many nodes w.r.t. n, and the size of $C_{\mathcal{B}_n}$ is linear in the size of $\mathcal{B}_n$, the size of $C_{\mathcal{B}_n}$ is exponential in n. By Theorem 11 we obtain that $C_{\mathcal{B}_n} \equiv \mathsf{lcs}(C_{1n}, \ldots, C_{nn})$. In order to prove that the size of any $\mathcal{EL}$-concept description equivalent to $\mathsf{lcs}(C_{1n}, \ldots, C_{nn})$ is exponential in the size of $C_{1n}, \ldots, C_{nn}$ it remains to be shown that there does not exist an $\mathcal{EL}$-concept description C such that $C \equiv C_{\mathcal{B}_n}$ and $|C| < |C_{\mathcal{B}_n}|$.

Assume that there exists an $\mathcal{EL}$-concept description and hence an $\mathcal{EL}$-description tree $\mathcal{G}_C = (V, E, v_0, l)$ such that

- there exist homomorphism φ_i from $\mathcal{G}_C$ to $\mathcal{G}_{in}$, i.e., for all $1 \leq i \leq n$ we have $C_{in} \sqsubseteq C$, and
- there exists a homomorphism φ from $\mathcal{B}_n$ to $\mathcal{G}_C$, i.e., $C \sqsubseteq C_{\mathcal{B}_n} \equiv \mathsf{lcs}(C_{1n}, \ldots, C_{nn})$, and
- $|\mathcal{G}_C| < |\mathcal{B}_n|$.

Due to the last item, we have to distinguish two cases, namely (1) $|V| < |V_n|$ and (2) $\sum_{v \in V} |\ell(v)| < \sum_{v \in V_n} |\ell_n(v)|$. Note that $|E| < |E_n|$ would imply $|V| < |V_n|$ since both G_C and $\mathcal{B}_n$ are trees. Now, in both cases we derive a contradiction to $C \sqsubseteq C_{\mathcal{G}_{1n} \times \ldots \times \mathcal{G}_{nn}}$.

First note that there exist no edges labeled with some $s \in N_R$, $s \neq r$, in $\mathcal{G}_C$, because otherwise, there cannot exist a homomorphism from $\mathcal{G}_C$ to any of the $\mathcal{EL}$-description trees $\mathcal{G}_{in}$, $1 \leq i \leq n$. This yields a contradiction to $C_{in} \sqsubseteq C$ for all $1 \leq i \leq n$. Further, each label of a node $v \in V$ is of the form $\emptyset$, $\{P\}$, or $\{Q\}$, because otherwise, there exists some $i \in \{1, \ldots, n\}$ such that there cannot exist a homomorphism from $\mathcal{G}_C$ to $\mathcal{G}_{in}$. Additionally, we have

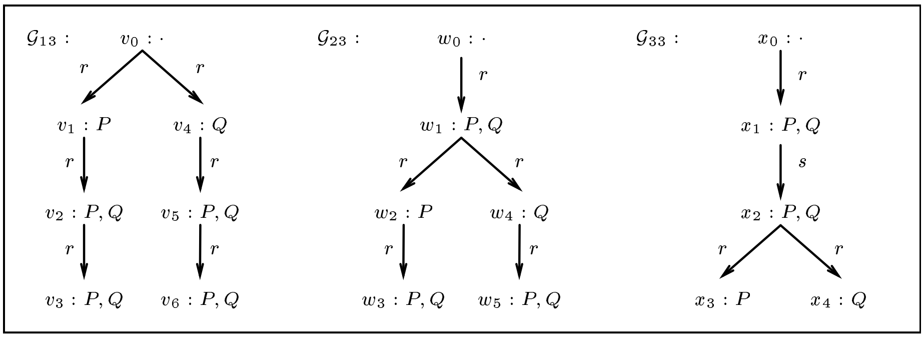

Figure 5: $\mathcal{EL}$-description trees for Example 14.

$\ell(v_0) = \emptyset$, because otherwise, the concept description corresponding to the same tree where the root is labeled with the empty set subsumes $C_\mathcal{G}$ as well as C_{in} for all $1 \leq i \leq n$ (because each root v_{0i} has empty label). Thus, $C_\mathcal{G}$ would be no lcs of $C_{1n}, \ldots, C_{nn}$.

Now assume $|\mathcal{G}_C| < |\mathcal{B}_n|$ and $|V| < |V_n|$. Then there exist $w_1 \neq w_2$ in $\mathcal{B}_n$, $w_1 \neq w_0$ or $w_2 \neq w_0$, with $\varphi(w_1) = \varphi(w_2) = v'$ for some v' in $\mathcal{G}_C$. W.l.o.g. we assume $w_1 \neq w_0$ and $\ell_n(w_1) = \{P\}$. This implies $\ell(v') = \{P\}$ and $\ell_n(w_2) \subseteq \{P\}$. Due to the properties of $\mathcal{B}_n$ it follows that the unique r-predecessors w_1' and w_2' of w_1 and w_2, respectively, are also different. Since φ is a homomorphism from $\mathcal{B}_n$ to $\mathcal{G}_C$ and $\mathcal{G}_C$ is an $\mathcal{EL}$-description tree, there exists a unique r-predecessor v'' of v' in $\mathcal{G}_C$ and we have $\varphi(w_1') = \varphi(w_2') = v''$. Let k be the length of the unique path from v_0 to v' in $\mathcal{G}_C$. By induction, we can easily show that when iterating the above argument k times, then we obtain predecessors w_1^k and w_2^k of w_1 and w_2 in $\mathcal{B}_n$ that still must be different. Further, the root v_0 is the predecessor of v' reached after k steps. Because $\mathcal{G}_C$ and $\mathcal{B}_n$ are trees, w_0 is the unique node in $\mathcal{B}_n$ mapped onto v_0 by φ. Thus, $w_1^k = w_2^k = w_0$ in contradiction to $w_1^k \neq w_2^k$.

In order to complete the proof, it remains to consider the case $|\mathcal{G}_C| < |\mathcal{B}_n|$ and $|V| = |V_n|$. Then we have $\sum_{v \in V} |\ell(v)| < \sum_{v \in V_n} |\ell_n(v)|$. Since each node except the root in $\mathcal{B}_n$ has a label of size 1, there exists a node $v \in V \setminus \{v_0\}$ such that $\ell(v) = \emptyset$. Further, there does not exist a node $w \in V_n \setminus \{w_0\}$ with $\varphi(w) = v$, because $\ell_n(w) \not\subseteq \ell(v)$ for all $w \in V_n \setminus \{w_0\}$. Thus, there exist $w_1 \neq w_2$ in $\mathcal{B}_n$, $w_1 \neq w_0$ or $w_2 \neq w_0$, with $\varphi(w_1) = \varphi(w_2) = v'$ for some $v' \in V$. Now, we obtain a contradiction in the same way as in the first case.

Due to Example 14 we obtain

Proposition 15 *The size of the lcs of n $\mathcal{EL}$-concept descriptions $C_1, \ldots, C_n$ of size linear in n may grow exponential in n.*

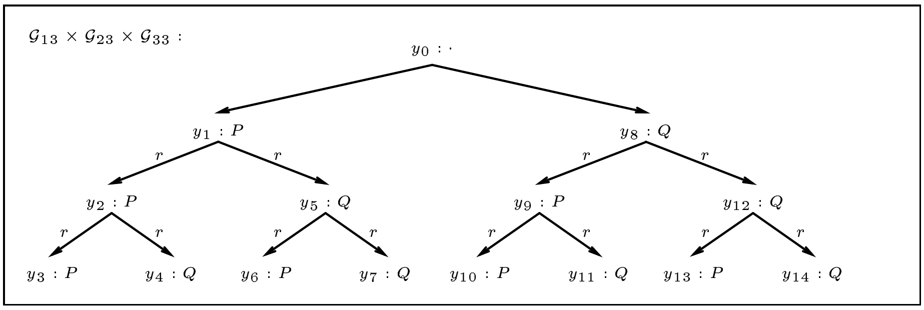

Figure 6: On the complexity of the lcs in $\mathcal{EL}$, continued.

4 Extending the results to $\mathcal{FLE}$

Our goal is to obtain a characterization of the lcs in $\mathcal{FLE}$ analogous to the one given in Theorem 11 for $\mathcal{EL}$. To achieve this goal, we first extend the characterization of subsumption by homomorphisms from $\mathcal{EL}$ to $\mathcal{FLE}$.

4.1 Subsumption in $\mathcal{FLE}$

In order to cope with value restrictions occurring in $\mathcal{FLE}$-concept descriptions, we allow in $\mathcal{FLE}$-description trees for two types of edges, namely those labeled with a primitive role $r \in N_R$ (corresponding to existential restrictions of the form $\exists r.C$) and those labeled with $\forall r$ for some $r \in N_R$ (corresponding to value restrictions of the form $\forall r.D$).

Definition 16 ($\mathcal{FLE}$-description trees) *An $\mathcal{FLE}$-*description tree *is a tree of the form $\mathcal{G} = (V, E, v_0, \ell)$ with root v_0 where*

- *the edges in E are labeled with primitive roles r from N_R or with $\forall r$ for some $r \in N_R$, and*
- *the nodes $v \in V$ are labeled with sets $\ell(v)$ of primitive concepts from N_C.*

The empty label corresponds to the top-concept.

Just as for $\mathcal{EL}$, there exists a 1-1 correspondence between $\mathcal{FLE}$-concept descriptions and $\mathcal{FLE}$-description trees. Every $\mathcal{FLE}$-concept description C can be written (modulo equivalence) as $C \equiv P_1 \sqcap \ldots \sqcap P_n \sqcap \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_m.C_m \sqcap \forall s_1.D_1 \sqcap \ldots \sqcap \forall s_k.D_k$ with $P_i \in N_C \cup \{\top\}$. Such a concept description can now be translated into an $\mathcal{FLE}$-description tree $\mathcal{G}(C) = (V, E, v_0, \ell)$ by induction on the role depth of C as follows.[5]

If $\mathsf{depth}(C) = 0$ **then** $V := \{v_0\}$, $E := \emptyset$, and $\ell(v_0) := \{P_1, \ldots, P_n\} \setminus \{\top\}$.

[5] We introduce the notion $\mathcal{G}(C)$ instead of $\mathcal{G}_C$, because we will use $\mathcal{G}_C$ later on for another purpose.

If $\mathsf{depth}(C) > 0$ **then** for $1 \leq i \leq m$, let $\mathcal{G}_i = (V_i, E_i, v_{0i}, \ell_i)$ be the inductively defined $\mathcal{FLE}$-description trees corresponding to C_i, and, for $1 \leq j \leq k$, let $\mathcal{G}'_j = (V'_j, E'_j, v'_{0j}, \ell'_j)$ be the inductively defined $\mathcal{FLE}$-description tree corresponding to D_j, where, w.l.o.g., the V_i and V'_j are pairwise disjoint. Then

- $V := \{v_0\} \cup \bigcup_{1 \leq i \leq m} V_i \cup \bigcup_{1 \leq j \leq k} V'_j$,
- $E := \{v_0 r_i v_{0i} \mid 1 \leq i \leq m\} \cup \bigcup_{1 \leq i \leq m} E_i \cup \bigcup_{1 \leq j \leq m} E'_j$,
- $\ell(v) := \begin{cases} \{P_1, \ldots, P_n\} \setminus \{\top\}, & v = v_0 \\ \ell_i(v), & v \in V_i, 1 \leq i \leq m \\ \ell'_j(v), & v \in V'_j, 1 \leq j \leq k \end{cases}$

Conversely, every $\mathcal{FLE}$-description tree $\mathcal{G} = (V, E, v_0, \ell)$ can again be translated into an $\mathcal{FLE}$-concept description $C_\mathcal{G}$ by induction on the depth of $\mathcal{G}$.

If $\mathsf{depth}(\mathcal{G}) = 0$ **then** $V = \{v_0\}$, $E = \emptyset$. If $\ell(v_0) = \emptyset$, then $C_\mathcal{G} := \top$; otherwise, we have $\ell(v_0) = \{P_1, \ldots, P_n\}$, $n \geq 1$ and define $C_\mathcal{G} := P_1 \sqcap \ldots \sqcap P_n$.

If $\mathsf{depth}(\mathcal{G}) > 0$ **then** let $\ell(v_0) = \{P_1, \ldots, P_n\}$, $n \geq 0$, and let $\{v_1, \ldots, v_m\}$ be the set of all successors of v_0 with $v_0 r_i v_i \in E$ for some $r_i \in N_R$, $1 \leq i \leq m$, and let $\{w_1, \ldots, w_k\}$ be the set of all successors of v_0 with $v_0 \forall s_i w_i \in E$ for some $s_i \in N_R$, $1 \leq i \leq k$. Further, let $C_1, \ldots, C_m$ $(D_1, \ldots, D_k)$ denote the inductively defined $\mathcal{FLE}$-concept descriptions corresponding to the subtrees of $\mathcal{G}$ with root v_i, $1 \leq i \leq m$ $(w_i, 1 \leq i \leq m)$. We define $C_\mathcal{G} := P_1 \sqcap \ldots \sqcap P_n \sqcap \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_m.C_m \sqcap \forall s_1.D_1 \sqcap \ldots \sqcap \forall s_k.D_k$.

The semantics of an $\mathcal{FLE}$-description tree $\mathcal{G}$ is given by the semantics of its corresponding $\mathcal{FLE}$-concept description $C_\mathcal{G}$.

The notion of a homomorphism also extends to $\mathcal{FLE}$-description trees in a natural way.

Definition 17 (Homomorphisms between $\mathcal{FLE}$-description trees)
Let $\mathcal{G} = (V_G, E_G, v_0, \ell_G)$ and $\mathcal{H} = (V_H, E_H, w_0, \ell_H)$ be $\mathcal{FLE}$-description trees. A mapping $\varphi : V_H \longrightarrow V_G$ is a homomorphism *from $\mathcal{H}$ to $\mathcal{G}$ iff the following conditions are satisfied:*

1. *$\varphi(w_0) = v_0$,*
2. *$\ell_D(v) \subseteq \ell_C(\varphi(v))$ for all $v \in V_H$,*
3. *$\varphi(v) r \varphi(w) \in E_G$ for all $vrw \in E_H$, and*
4. *$\varphi(v) \forall r \varphi(w) \in E_C$ for all $v \forall r w \in E_H$.*

However, these straightforward extensions are not sufficient to obtain a sound and complete characterization of subsumption in $\mathcal{FLE}$ based on homomorphisms between $\mathcal{FLE}$-description trees.

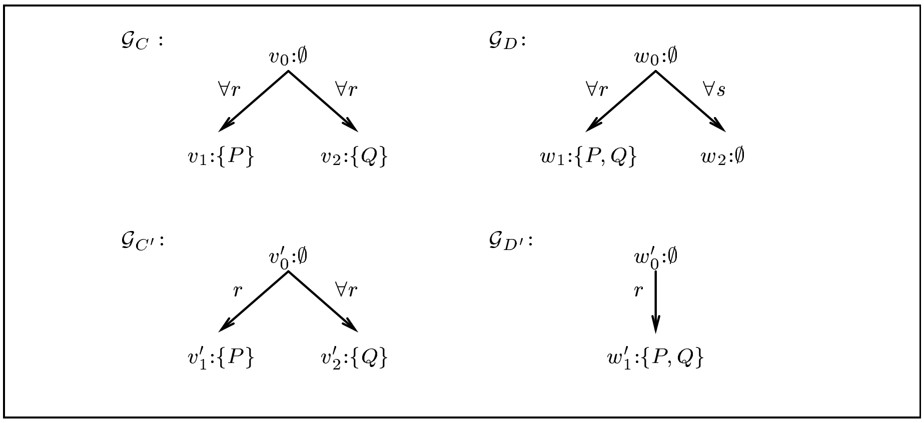

Figure 7: $\mathcal{FLE}$-description trees.

Example 18 Consider the following $\mathcal{FLE}$-concept descriptions:

$$\begin{array}{rcl} C & := & \forall r.P \sqcap \forall r.Q \\ D & := & \forall r.(P \sqcap Q) \sqcap \forall s.\top \\ C' & := & \exists r.P \sqcap \forall r.Q \\ D' & := & \exists r.(P \sqcap Q) \end{array}$$

The corresponding $\mathcal{FLE}$-description trees are depicted in Figure 7. It is easy to see that $C \sqsubseteq D$ and $C' \sqsubseteq D'$, but there exists neither a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$ nor from $\mathcal{G}_{D'}$ to $\mathcal{G}_{C'}$.

To avoid these problems, we must normalize the $\mathcal{FLE}$-concept descriptions before translating them into $\mathcal{FLE}$-description trees. The *normal form* of an $\mathcal{FLE}$-concept description C is obtained from C by exhaustively applying the following normalization rules. We denote the $\mathcal{FLE}$-description tree of the normal form of C by $\mathcal{G}_C$.

Definition 19 ($\mathcal{FLE}$ normalization rules) *Let E, F be two $\mathcal{FLE}$-concept descriptions and $r \in N_R$ a primitive role. The $\mathcal{FLE}$-*normalization rules *are defined as follows*

$$\begin{array}{rcl} \forall r.E \sqcap \forall r.F & \longrightarrow & \forall r.(E \sqcap F) \\ \forall r.E \sqcap \exists r.F & \longrightarrow & \forall r.E \sqcap \exists r.(E \sqcap F) \\ \forall r.\top & \longrightarrow & \top \\ E \sqcap \top & \longrightarrow & E \end{array}$$

The rules should be read modulo commutativity of conjunction; e.g., $\exists r.E \sqcap \forall r.F$ is also normalized to $\exists r.(E \sqcap F) \sqcap \forall r.F$. Since each normalization rule preserves equivalence, the resulting normalized $\mathcal{FLE}$-concept description is equivalent to the original one.

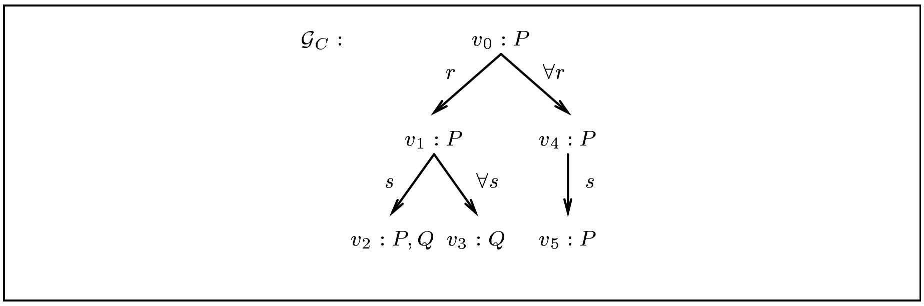

Figure 8: An $\mathcal{FLE}$-description tree.

In the case of $\mathcal{EL}$-concept descriptions, we have $C = C_{\mathcal{G}_C}$ (up to commutativity and associativity of conjunction). In the case of $\mathcal{FLE}$-concept descriptions we still have $C = C_{\mathcal{G}(C)}$ where $\mathcal{G}(C)$ denotes the $\mathcal{FLE}$-description tree obtained from C without first normalizing C. However, for the $\mathcal{FLE}$-description tree $\mathcal{G}_C$ we get $C \equiv C_{\mathcal{G}_C}$ (because the normalization rules preserve equivalence), but not necessarily $C = C_{\mathcal{G}_C}$.

Example 20 Consider the $\mathcal{FLE}$-concept description

$$C = P \sqcap \exists r.\forall s.Q \sqcap \forall r.P \sqcap \forall r.\exists s.P.$$

The $\mathcal{FLE}$-description tree $\mathcal{G}_C$ corresponding to C is depicted in Figure 8. We obtain the following $\mathcal{FLE}$-concept description from $\mathcal{G}_C$, which is obviously equivalent but not equal to C:

$$P \sqcap \exists r.(P \sqcap \exists s.(P \sqcap Q) \sqcap \forall s.Q) \sqcap \forall r.(P \sqcap \exists s.P).$$

The following lemma formalizes some important properties of $\mathcal{FLE}$-description trees that are obtained from $\mathcal{FLE}$-concept descriptions in normal form.

Lemma 21 *Let C be an $\mathcal{FLE}$-concept description and $\mathcal{G}_C = (V, E, v_0, \ell)$ its corresponding $\mathcal{FLE}$-description tree.*

1. *If C is in normal form, then it holds that*
   - *$C = C_{\mathcal{G}_C}$ up to commutativity and associativity of conjunction, and*
   - *$\mathcal{G}_C = \mathcal{G}_{C_{\mathcal{G}_C}}$ up to renaming nodes.*
2. *For each node $v \in V$ and each primitive role $r \in N_R$, v has at most one outgoing edge labeled $\forall r$.*
3. *Let $\{vrw, v\forall rw'\} \subseteq E$, and let C denote the $\mathcal{FLE}$-concept description corresponding to the subtree of $\mathcal{G}$ with root w, and C' the one corresponding to the subtree of $\mathcal{G}$ with root w'. Then $C \sqsubseteq C'$.*

4. *Leaves in $\mathcal{G}$ labeled with the empty set cannot be reached via an edge labeled $\forall r$ for some $r \in N_R$, i.e., $C_\mathcal{G}$ does not contain a subconcept of the form $\forall r.\top$.*

Proof: The first item can be easily shown by induction on $\mathsf{depth}(C)$ and by induction on $\mathsf{depth}(\mathcal{G})$, respectively. Note that translating an $\mathcal{FLE}$-description tree $\mathcal{G}$ that is obtained from an $\mathcal{FLE}$-concept description in normal form always yields an $\mathcal{FLE}$-concept description in normal form.

The second item is a direct consequence from the exhaustive application of the first normalization rule ($\forall r.E \sqcap \forall r.F \longrightarrow \forall r.(E \sqcap F)$). The third item follows from the exhaustive application of the second normalization rule ($\forall r.E \sqcap \exists r.F \longrightarrow \forall r.E \sqcap \exists r.(E \sqcap F)$). Finally, the last item is a consequence of the application of the normalization rules $\forall r.\top \longrightarrow \top$ and $E \sqcap \top \longrightarrow \top$. □

In order to prove soundness and completeness of the characterization of subsumption in $\mathcal{FLE}$, we will use the notion of the *canonical interpretation* $\mathcal{I}_C$ of C.

Definition 22 (Canonical interpretation) *Let C be an $\mathcal{FLE}$-concept description and $\mathcal{G}_C = (V, E, v_0, \ell)$ the corresponding $\mathcal{FLE}$-description tree of C. The* canonical interpretation *of C is defined by $\mathcal{I}_C := (\Delta_C, \cdot^{\mathcal{I}_C})$ where*

- $\Delta_C := V$,
- *for $P \in \mathcal{C}$, $P^{\mathcal{I}_C} := \{v \in V \mid P \in \ell(v)\}$, and*
- *for $r \in \mathcal{R}$, $r^{\mathcal{I}_C} := \{(v, w) \mid vrw \in E\}$.*

Lemma 23 *Let C be an $\mathcal{FLE}$-concept description and $\mathcal{I}_C$ its canonical interpretation. Then we have $v_0 \in C^{\mathcal{I}_C}$ where v_0 denotes the root of $\mathcal{G}_C$.*

Proof: In Section 5.1, we will introduce an extension of Lemma 23 to $\mathcal{ALE}$. The proof of the extended lemma also yields a proof of Lemma 23. □

We are now ready to prove the characterization of subsumption in $\mathcal{FLE}$ based on homomorphisms between $\mathcal{FLE}$-description trees.

Theorem 24 *Let C, D be $\mathcal{FLE}$-concept descriptions and $\mathcal{G}_C, \mathcal{G}_D$ their corresponding $\mathcal{FLE}$-description trees. Then $C \sqsubseteq D$ iff there exists a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$.*

Proof: Let $\mathcal{G}_C = (V_C, E_C, v_0, \ell_C)$ and $\mathcal{G}_D = (V_D, E_D, w_0, \ell_D)$.

We first prove the if-direction. Let φ be a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$ and $x_0 \in C^\mathcal{I}$ for an interpretation $\mathcal{I} = (\Delta, \cdot^\mathcal{I})$. We prove $x_0 \in D^\mathcal{I}$ by induction on the number $|V_D|$ of nodes in $\mathcal{G}_D$.

$|V_D| = 1$**:** Let $V_D := \{w_0\}$. Since there exists a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$, we have $\ell_D(w_0) \subseteq \ell_C(v_0)$. Let $x_0 \in C^\mathcal{I}$. Since D is the conjunction of the atomic concepts in $\ell_D(w_0)$ and each primitive concept $P \in \ell_C(v_0)$ occurs in the conjunction on top-level of C, it follows $x_0 \in D^\mathcal{I}$.

$|V_D| > 1$: For an $\mathcal{FLE}$-description tree $\mathcal{G}$ and a node w in $\mathcal{G}$, let $\mathcal{G}(w)$ denote the subtree of $\mathcal{G}$ with root w. We have

$$\begin{aligned} C \equiv C_{\mathcal{G}_C} &= \bigsqcap_{P \in \ell_C(v_0)} P \sqcap \bigsqcap_{v_0 r v \in E_C} \exists r.C_{\mathcal{G}_C(v)} \sqcap \bigsqcap_{v_0 \forall r v \in E_C} \forall r.C_{\mathcal{G}_C(v)}, \\ D \equiv C_{\mathcal{G}_D} &= \bigsqcap_{P \in \ell_D(w_0)} P \sqcap \bigsqcap_{w_0 r w \in E_D} \exists r.C_{\mathcal{G}_D(w)} \sqcap \bigsqcap_{w_0 \forall r w \in E_D} \forall r.C_{\mathcal{G}_D(w)}. \end{aligned}$$

- Since $x_0 \in C^{\mathcal{I}} = C^{\mathcal{I}}_{\mathcal{G}_C}$, $\ell_D(w_0) \subseteq \ell_C(v_0)$ implies that $x_0 \in P^{\mathcal{I}}$ for all $P \in \ell_D(w_0)$.
- Let $w_0 r w \in E_D$. Then we have $v_0 r \varphi(w) \in E_C$ and the homomorphism φ restricted to the nodes in $\mathcal{G}_D(w)$ is a homomorphism from $\mathcal{G}_D(w)$ to $\mathcal{G}_C(\varphi(w))$. By induction we get $C_{\mathcal{G}_C(\varphi(w))} \sqsubseteq C_{\mathcal{G}_D(w)}$. Since $v_0 r \varphi(w) \in E_C$, the concept description $\exists r.C_{\mathcal{G}_C(\varphi(w))}$ occurs in the conjunction on top-level of $C_{\mathcal{G}_C}$. Thus, there exists $y \in \Delta$ with $(x_0, y) \in r^{\mathcal{I}}$ and $y \in C^{\mathcal{I}}_{\mathcal{G}_C(\varphi(w))}$. It follows that $y \in C^{\mathcal{I}}_{\mathcal{G}_D(w)}$, and hence, $x_0 \in \exists r.C^{\mathcal{I}}_{\mathcal{G}_D(w)}$.
- Let $w_0 \forall r w \in E_D$. We have $v_0 \forall r \varphi(w) \in E_C$ and, again by induction, $C_{\mathcal{G}_C(\varphi(w))} \sqsubseteq C_{\mathcal{G}_D(w)}$. As before, the concept description $\forall r.C_{\mathcal{G}_C(\varphi(w))}$ is a conjunct in the conjunction on top-level of $C_{\mathcal{G}_C}$. Thus, $(x_0, y) \in r^{\mathcal{I}}$ implies $y \in C^{\mathcal{I}}_{\mathcal{G}_C(\varphi(w))} \subseteq C^{\mathcal{I}}_{\mathcal{G}_D(w)}$, and hence, $x_0 \in \forall r.C^{\mathcal{I}}_{\mathcal{G}_D(w)}$.

Thus, x_0 is an instance of each conjunct occurring in the conjunction on top-level of $C_{\mathcal{G}_D}$, which means that $x_0 \in C^{\mathcal{I}}_{\mathcal{G}_D} = D^{\mathcal{I}}$.

We prove the only-if-direction by induction on *depth*(D). First, we give an outline of the proof. We assume that $C \sqsubseteq D$. In order to construct a homomorphism from $\mathcal{G}_D$ into $\mathcal{G}_C$, we first show that $\ell_D(w_0) \subseteq \ell_C(v_0)$. Then we prove, for each successor w of w_0, that there exists a successor v of v_0, reached via an edge with the same label, such that $C_{\mathcal{G}_C(v)} \sqsubseteq C_{\mathcal{G}_D(w)}$. By induction, we obtain a homomorphism φ'_w from $\mathcal{G}_{C_{\mathcal{G}_D(w)}}$ to $\mathcal{G}_{C_{\mathcal{G}_C(v)}}$. Since $C_{\mathcal{G}_C(v)}$ and $C_{\mathcal{G}_D(w)}$ are in normal form, we get $\mathcal{G}_C(v) = \mathcal{G}_{C_{\mathcal{G}_C(v)}}$ and $\mathcal{G}_D(w) = \mathcal{G}_{C_{\mathcal{G}_D(w)}}$ up to renaming nodes (see Lemma 21.1), and hence, φ'_w also yields a homomorphism φ_w from $\mathcal{G}_D(w)$ to $\mathcal{G}_{C(v)}$ by just renaming the nodes appropriately. Using these homomorphisms φ_w we can define a homomorphism φ from $\mathcal{G}_D$ to $\mathcal{G}_C$.

Base case: $\mathsf{depth}(D) = 0$.
Then it is $D = P_1 \sqcap \ldots \sqcap P_n$ and $\mathcal{G}_D = (\{w_0\}, \emptyset, w_0, l_D)$ with $l_D(w_0) = \{P_1, \ldots, P_n\}$. Now, $C \sqsubseteq D$ and $v_0 \in C^{\mathcal{I}_C}$ (see Lemma 23) imply $v_0 \in D^{\mathcal{I}_C}$. By definition of $\mathcal{I}_C$ this means that $P_i \in \ell_C(v_0)$ for $1 \leq i \leq n$. Thus, $\varphi : \{w_0 \longmapsto v_0\}$ is a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$.

Induction step: $\mathsf{depth}(D) > 0$.
Let $D = P_1 \sqcap \ldots \sqcap P_n \sqcap \exists r_1.D_1 \sqcap \ldots \sqcap \exists r_m.D_m \sqcap \forall s_1.F_1 \sqcap \ldots \sqcap \forall s_k.F_k$ and $\mathcal{G}_D = (V_D, E_D, w_0, l_D)$. In order to define a homomorphism φ from $\mathcal{G}_D$ to $\mathcal{G}_C$,

we first consider the condition on the roots v_0 and w_0 and then the conditions on r- and $\forall r$-successors of w_0 in $\mathcal{G}_D$.

As before, $C \sqsubseteq D$ and $v_0 \in C^{I_C}$ imply $\ell_D(w_0) \subseteq \ell_C(v_0)$.

Let $w_0 r w \in E_D$. We have to show that there exists $v \in V_C$ with $v_0 r v \in E_C$ and $C_{\mathcal{G}_C(v)} \sqsubseteq C_{\mathcal{G}_D(w)}$. This subsumption relationship is necessary for applying the induction hypothesis. We have $v_0 \in C^{\mathcal{I}_C}$ and hence $v_0 \in D^{\mathcal{I}_C}$. Thus, there exists $v \in \Delta_C = V_C$ with $(v_0, v) \in r^{\mathcal{I}_C}$ and $v \in C^{\mathcal{I}_C}_{\mathcal{G}_D(w)}$. Consequently, the set $\{v_1, \ldots, v_\nu\}$ of all r-successors of v_0 in $\mathcal{G}_C$ is not empty. Assume that $C_{\mathcal{G}_C(v_i)} \not\sqsubseteq C_{\mathcal{G}_D(w)}$ for all $1 \leq i \leq \nu$. Then there exist pairwise disjoint interpretations $\mathcal{I}_i = (\Delta_{\mathcal{I}_i}, \cdot^{\mathcal{I}_i})$ and $x_i \in \Delta_{\mathcal{I}_i}$ such that $x_i \in C^{\mathcal{I}_i}_{\mathcal{G}_C(v_i)}$, $x_i \notin C^{\mathcal{I}_i}_{\mathcal{G}_D(w)}$. We define an interpretation $\mathcal{J}$ in such a way that we can derive a contradiction to $C \sqsubseteq D$. We define $\mathcal{J} = (\Delta_J, \cdot^J)$ by

- $\Delta_J := \left(\Delta_{\mathcal{I}_C} \cup \bigcup_{1 \leq i \leq \nu} \Delta_{\mathcal{I}_i}\right) \setminus \{v_1, \ldots, v_\nu\}$,
- $P^J := P^{\mathcal{I}_C} \setminus \{v_1, \ldots, v_\nu\} \cup \bigcup_{1 \leq i \leq \nu} P^{\mathcal{I}_i}$,
- $s^J := s^{\mathcal{I}_C} \cup \bigcup_{1 \leq i \leq \nu} s^{\mathcal{I}_i}$ for $s \neq r$,
  $r^J := (r^{\mathcal{I}_C} \setminus \{(v_0, v_i) \mid 1 \leq i \leq \nu\}) \cup \bigcup_{1 \leq i \leq \nu} r^{\mathcal{I}_i} \cup \{(v_0, x_i) \mid 1 \leq i \leq \nu\}$.

We first show that $v_0 \in C^J$. Therefore, it is sufficient to show that $v_0 \in C^J_{\mathcal{G}_C}$, because $C \equiv C_{\mathcal{G}_C}$. We show that v_0 is an instance of each conjunct occurring in the conjunction on top-level of $C_{\mathcal{G}_C}$.

By definition of $\mathcal{J}$, we have $v_0 \in P^J$ for all $P \in \ell_C(v_0)$.

For each r-successor v_i of v_0 in $\mathcal{G}_C$, i.e., for each existential restriction of the form $\exists r.C_i$ in the conjunction on top-level of C, there exists x_j with $(v_0, x_j) \in r^J$ and $x_j \in C_i^J$.

Let $\forall r.C'$ be the value restriction on r in the conjunction on top-level of $C_{\mathcal{G}_C}$. (If there exists no value restriction of this form on top-level of C, nothing more has to be shown.) By definition of $\mathcal{J}$, $\{x_1, \ldots, x_\nu\}$ is the set of all r-successors of v_0 in $\mathcal{J}$. For each $1 \leq i \leq \nu$, there exists an r-successor v of v_0 in $\mathcal{G}_C$ such that $x_i \in C^J_{\mathcal{G}_C(v)}$. By Lemma 21.3 we know $C_{\mathcal{G}_C(v)} \sqsubseteq C'$ and hence $x_i \in C'^J$. Thus, $v_0 \in (\forall r.C')^J$.

By definition of $\mathcal{J}$, all existential restrictions of the form $\exists s.C'$, $s \neq r$, and all value restrictions of the form $\forall s.C'$, $s \neq r$, occurring in the conjunction on top-level of C are also satisfied by v_0. Hence, we have $v_0 \in C^J_{\mathcal{G}_C} = C^J$.

It remains to show that $v_0 \notin D^J$. By construction of $\mathcal{J}$, none of the r-successors x_j of v_0 satisfies $x_j \in C^J_{\mathcal{G}_D(w)}$. So we have $v_0 \notin D^J$. Consequently, $C^J \not\subseteq D^J$ which is a contradiction to $C \sqsubseteq D$ and we have shown that there exists $v \in V_C$ with $v_0 r v \in E_C$ and $C_{\mathcal{G}_C(v)} \sqsubseteq C_{\mathcal{G}_D(w)}$. By the induction hypothesis, we can define a homomorphism φ_w from $\mathcal{G}_D(w)$ to $\mathcal{G}_C(v)$ as described in the outline of the proof of the only-if-direction.

In the last step, we have to consider edges of the form $w_0 \forall s w \in E_D$. We have to show that there exists $v \in V_C$ such that $v_0 \forall s v \in E_C$ and $C_{\mathcal{G}_C(v)} \sqsubseteq C_{\mathcal{G}_D(w)}$. Therefore, we first assume that there does not exist a node $v \in V_C$ such that $v_0 \forall s v \in E_C$.

We know that $v_0 \in D^{\mathcal{I}_C}$ and $C_{\mathcal{G}_D(w)} \not\equiv \top$ (see Lemma 21.4). Thus, there exists an interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ and $x \in \Delta$ such that $x \notin C^{\mathcal{I}}_{\mathcal{G}_D(w)}$. W.l.o.g. we may assume that $\mathcal{I}_C$ and $\mathcal{I}$ are disjoint. We define an interpretation $\mathcal{J} = (\Delta_J, \cdot^J)$ such that $v_0 \in \Delta_J$ and $v_0 \in D^J \setminus C^J$:

- $\Delta_J := \Delta_{\mathcal{I}_C} \cup \Delta$,
- $P^J := P^{\mathcal{I}_C} \cup P^{\mathcal{I}}$,
- $r^J := r^{\mathcal{I}_C} \cup r^{\mathcal{I}}$ for $r \neq s$ and
  $s^J := s^{\mathcal{I}_C} \cup s^{\mathcal{I}} \cup \{(v_0, x)\}$.

Since $x \notin C^J_{\mathcal{G}_D(w)}$ and $(v_0, x) \in s^J$, it is $v_0 \notin D^J$. By assumption, there is no $\forall s$-successor in $\mathcal{G}_C$. Thus there is no value restriction of the form $\forall s.C'$ in the conjunction on top-level of C and hence, it is $v_0 \in C^J$. This implies $C^J \not\subseteq D^J$ which is a contradiction to $C \sqsubseteq D$. Thus we have shown that there exists a node $v \in V_C$ with $v_0 \forall s v \in E_C$. Note that v is unique since each node in $\mathcal{G}_C$ has at most one $\forall s$-successor (see Lemma 21.2).

It remains to show that $C_{\mathcal{G}_C(v)} \sqsubseteq C_{\mathcal{G}_D(w)}$. For this purpose, let us assume that $C_{\mathcal{G}_C(v)} \not\sqsubseteq C_{\mathcal{G}_D(w)}$. Then there exists an interpretation $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ and $x \in \Delta$ such that $x \in C^{\mathcal{I}}_{\mathcal{G}_C(v)}$ and $x \notin C^{\mathcal{I}}_{\mathcal{G}_D(w)}$. W.l.o.g. let $\mathcal{I}$ and $\mathcal{I}_C$ be disjoint. As before, we will derive a contradiction to $C \sqsubseteq D$. Let $\mathcal{J} = (\Delta_J, \cdot^J)$ be defined as above. We have $x \notin C^J_{\mathcal{G}_D(w)}$ and hence $v_0 \notin D^J$. Furthermore, we know that $C \equiv C_{\mathcal{G}_C}$ and that $\forall s.C_{\mathcal{G}_C(v)}$ is the only value restriction for s in the conjunction on top level of $C_{\mathcal{G}_C}$. Since $x \in C^J_{\mathcal{G}_C(v)}$ it is now easy to see that $v_0 \in C^J$. This implies $C^J \not\subseteq D^J$ which is a contradiction to $C \sqsubseteq D$.

To sum up, we have shown that for each $w \in V_D$ with $w_0 \forall s w \in E_D$ there exists a node $v \in V_C$ such that $v_0 \forall s v \in E_C$ and $C_{\mathcal{G}_C(v)} \sqsubseteq C_{\mathcal{G}_D(w)}$. By the induction hypothesis, we can again define a homomorphism φ_w from $\mathcal{G}_D(w)$ to $\mathcal{G}_C(v)$ as described in the outline of the proof of the only-if-direction.

Now we can define a homomorphism φ from $\mathcal{G}_D$ to $\mathcal{G}_C$ by

$$\varphi := \{w_0 \mapsto v_0\} \cup \bigcup_{w_0 r w \in E_D} \varphi_w \cup \bigcup_{w_0 \forall r w \in E_D} \varphi_w.$$

By construction, φ is a well-defined homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$.

This completes the proof of Theorem 24. □

Remark 25 *In the proof of the only-if-direction of Theorem 24 we made use of the fact that the $\mathcal{FLE}$-concept descriptions are normalized before they are translated into their corresponding $\mathcal{FLE}$-description trees. For example, it is necessary to collect and represent all value restrictions on a primitive role $r \in N_R$ occurring in a conjunction by only one subtree whose root v is reached via an edge labeled with $\forall r$ in order to derive $v_0 \in C^J$ and hence $C \not\sqsubseteq D$ in the last step of the proof.*

A comparison with tableaux-based algorithms

It should be noted that there is a close relationship between the normalization rules introduced above and some of the so-called *propagation rules* employed by tableaux-based subsumption algorithms, as e.g. introduced in [12]. The main idea underlying our second normalization rule and the propagation rule treating value restrictions is to make the knowledge implicitly given by a conjunction of the form $\forall r.E \sqcap \exists r.F$ explicit by propagating E onto the existential restriction according to the equivalence $\forall r.E \sqcap \exists r.F \equiv \forall r.E \sqcap \exists r.(E \sqcap F)$. As shown in [12], this propagation rule may lead to an exponential blow-up of the tableau, and the same is true for our normalization rule. More precisely, applying the normalization rules introduced above to an $\mathcal{FLE}$-concept description C may lead to a normalized concept description, and hence a corresponding $\mathcal{FLE}$-description tree $\mathcal{G}_C$, of size exponential in the size of C.

Example 26 We define a sequence of $\mathcal{FLE}$-concept descriptions $\{C_1, C_2, C_3, \ldots\}$ such that

- C_n has role depth n,
- $|C_n|$ is polynomial in n, and
- the size of the description trees $\mathcal{G}_n$ corresponding to C_n is exponential in n.

We define C_n, $n \geq 1$ inductively by

- $C_1 := \exists r.P \sqcap \exists r.Q$ and
- $C_n := \exists r.P \sqcap \exists r.Q \sqcap \forall r.C_{n-1}$.

As an example, the $\mathcal{FLE}$-description tree $\mathcal{G}_2$ corresponding to $C_2 = \exists r.P \sqcap \exists r.Q \sqcap \forall r.\exists r.P \sqcap \exists r.Q$ is depicted in Figure 9.

We first show by induction on n that $|C_n|$ is linear in n.

Base case: $n = 1$. Obvious.

Induction step: $n \longrightarrow n+1$
We have $C_{n+1} := \exists r.P \sqcap \exists r.Q \sqcap \forall r.C_n$. By induction, $|C_n|$ is linear in n. Further, we have $|C_{n+1}| = |C_n| + 10$, which is still linear in n.

We now want to show that the size of $\mathcal{G}_n$ is exponential in n. Therefore, we show by induction on n that the normal form of C_n obtained from C_n by exhaustively applying the normalization rules introduced on page 17 has size greater than 2^n. Since $\mathcal{G}_n$ is obtained from the normal form of C_n, and its size is linear in the size of the normal form of C_n, this implies that $|\mathcal{G}_n| > 2^n$.

Base case: $n = 1$
Obviously, C_1 is already in normal form. We have $|C_1| = 7 > 2^1$.

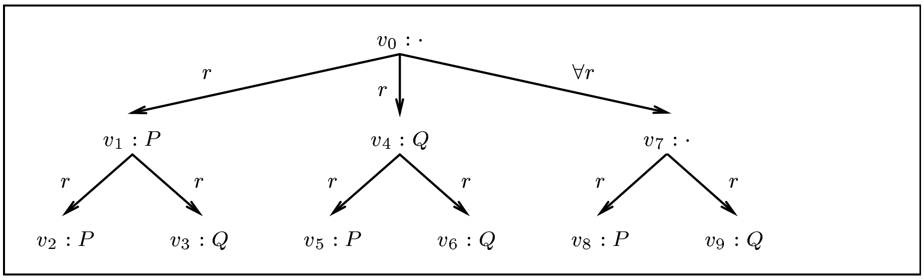

Figure 9: The $\mathcal{FLE}$-description tree G_2.

Induction step: $n \longrightarrow n+1$

We have $C_{n+1} := \exists r.P \sqcap \exists r.Q \sqcap \forall r.C_n$. Let C'_n denote the normal form of C_n. Then the normal form of C_{n+1} is given by $C'_{n+1} = \exists r.(P \sqcap C'_n) \sqcap \exists r.(Q \sqcap C'_n) \sqcap \forall r.C'_n$. By induction we have $|C'_n| > 2^n$. Hence, $|C_{n+1}| > 2 \times |C'_n| > 2 \times 2^n = 2^{n+1}$.

Note that the exponential blow-up cannot be avoided since (i) as for $\mathcal{EL}$, existence of a homomorphism between $\mathcal{FLE}$-description trees can be tested in polynomial time; and (ii) subsumption in $\mathcal{FLE}$ is an NP-complete problem [12].

4.2 Least Common Subsumer for $\mathcal{FLE}$

Just as for $\mathcal{EL}$, we can now use the characterization of subsumption in $\mathcal{FLE}$ by homomorphisms to characterize the lcs of two $\mathcal{FLE}$-concept descriptions by the product of $\mathcal{FLE}$-description trees.

Definition 27 (Product of $\mathcal{FLE}$-description trees)
The product $\mathcal{G} \times \mathcal{H}$ *of two $\mathcal{FLE}$-description trees $\mathcal{G} = (V_G, E_G, v_0, \ell_G)$ and $\mathcal{H} = (V_H, E_H, w_0, \ell_H)$ is defined by induction on the depth of the trees. The node (v_0, w_0) labeled with $\ell_G(v_0) \cap \ell_H(w_0)$ is the root of $\mathcal{G} \times \mathcal{H}$. For each r-successor v of v_0 in $\mathcal{G}$ and w of w_0 in $\mathcal{H}$, we obtain an r-successor (v, w) of (v_0, w_0) in $\mathcal{G} \times \mathcal{H}$ that is the root of the inductively defined product of $\mathcal{G}(v)$ and $\mathcal{H}(w)$. Additionally, for each $\forall r$-successor v of v_0 in $\mathcal{G}$ and w of w_0 in $\mathcal{H}$, we obtain a $\forall r$-successor (v, w) of (v_0, w_0) in $\mathcal{G} \times \mathcal{H}$ that is the root of the inductively defined product of $\mathcal{G}(v)$ and $\mathcal{H}(w)$.*

Obviously, the product of two $\mathcal{FLE}$-description trees is again an $\mathcal{FLE}$-description tree.

Theorem 28 *Let C, D be $\mathcal{FLE}$-concept descriptions and $\mathcal{G}_C, \mathcal{G}_D$ the corresponding description trees. Then, $C_{\mathcal{G}_C \times \mathcal{G}_D}$ is the lcs of C and D.*

Proof: The proof of Theorem 28 can be adapted from the proof of Theorem 11 by (1) replacing $\mathcal{EL}$-concept descriptions and $\mathcal{EL}$-description trees by

$\mathcal{FLE}$-concept descriptions and $\mathcal{FLE}$-description trees, respectively, and (2) handling edges labeled with $\forall r$ for some $r \in N_R$ in the same way as those labeled with $r \in N_R$. □

Remark 29 *Each node in $\mathcal{G}_C \times \mathcal{G}_D$ has at most one $\forall r$-successor for each $r \in N_R$, because each node in $\mathcal{G}_C$ and $\mathcal{G}_D$, respectively, has at most one $\forall r$-successor (see Lemma 21.2). However, the $\mathcal{FLE}$-concept description $C_{\mathcal{G}_C \times \mathcal{G}_D}$ need not to be in normal form. For example, the lcs of $C = \forall r.P$ and $D = \forall r.Q$ is given by $C_{\mathcal{G}_C \times \mathcal{G}_D} = \forall r.\top$ which is obviously not in normal form, since the normalization rule $\forall r.\top \longrightarrow \top$ is applicable to $C_{\mathcal{G}_C \times \mathcal{G}_D}$.*

On the complexity of least common subsumer in $\mathcal{FLE}$

As mentioned above, $\mathcal{FLE}$ differs from $\mathcal{EL}$ in that the $\mathcal{FLE}$-description tree corresponding to an $\mathcal{FLE}$-concept description C may be of size exponential in the size of C. Therefore, even for *two* $\mathcal{FLE}$-concept descriptions C, D, the size of the lcs cannot be polynomially bounded.

Example 30 Let C_n, $n \geq 1$, be defined as in Example 26 and D_n, $n \geq 1$, defined as in Example 14. By Theorem 24, the lcs of C_n and D_n is given by $C_{\mathcal{G}_{C_n} \times \mathcal{G}_{D_n}}$. It is not hard to see that the product of $\mathcal{G}_{C_n}$ and $\mathcal{G}_{D_n}$ again yields the full binary tree $\mathcal{B}_n$ of depth n as defined in Example 14.

Furthermore, we can show analogously to the proof in Example 14 that $C_{\mathcal{B}_n}$ is minimal, i.e., there exists no $\mathcal{FLE}$-concept description C' such that (1) $C_n \sqsubseteq C'$ and $D_n \sqsubseteq C'$, (2) $C' \sqsubseteq C_{\mathcal{B}_n}$, and (3) $|C'| < |C_{\mathcal{B}_n}|$.

Both C_n and D_n are linear in n (see Example 26), but $\mathcal{B}_n$ and hence $\mathsf{lcs}(C_n, D_n)$ have size exponential in n (see Example 14).

Due to Example 30 we obtain:

Proposition 31 *The size of the lcs of* two *$\mathcal{FLE}$-concept descriptions C, D may be exponential in the size of C and D.*

It should be noted, that the size of the lcs of n $\mathcal{FLE}$-concept descriptions can be exponentially bounded. By induction on the role depth of C, it is not hard to show that the size $|\mathcal{G}_C|$ of the $\mathcal{FLE}$-description tree $\mathcal{G}_C$ corresponding to an $\mathcal{FLE}$-concept description C can be bounded by an exponential function $2^{c \cdot |C|}$ where c is a non-negative integer. Furthermore, the size of the lcs of n $\mathcal{FLE}$-concept descriptions $C_1, \ldots, C_n$ can be bounded by the size of the product $\mathcal{G}_{C_1} \times \ldots \times \mathcal{G}_{C_n}$. Since each $|\mathcal{G}_{C_i}|$ is bounded by $2^{c \cdot |C_i|}$, it is $|\mathcal{G}_{C_1} \times \ldots \times \mathcal{G}_{C_n}| \leq 2^{c \cdot M \cdot n}$ where $\mathsf{max}\{|C_1|, \ldots, |C_n|\}$. Thus, we obtain:

Proposition 32 *The size of the lcs of n $\mathcal{FLE}$-concept descriptions $C_1, \ldots, C_n$ can be bounded by an exponential function with $M := \mathsf{max}\{|C_1|, \ldots, |C_n|\}$ and n as its inputs.*

5 Existential Restrictions and Inconsistencies

In order to characterize the lcs of two $\mathcal{ALE}$-concept descriptions by the product of description trees, we must adapt the notions description tree, homomorphism, and product appropriately, taking into account the additional constructors primitive negation and bottom-concept.

5.1 Subsumption in $\mathcal{ALE}$

First, we extend the definition of an $\mathcal{FLE}$-description tree to $\mathcal{ALE}$ by additionally allowing for negated atomic concepts $\neg P$ as well as $\bot$ in the labels of nodes.

Definition 33 ($\mathcal{ALE}$-description trees) *An $\mathcal{ALE}$-*description tree *is a tree of the form $\mathcal{G} = (V, E, v_0, \ell)$ with root v_0 where*

- *the edges in E are labeled with primitive roles r from N_R or with $\forall r$ for some $r \in N_R$, and*
- *the nodes $v \in V$ are labeled with sets $\ell(v) = \{P_1, \dots, P_n\}$ where each P_i, $1 \leq i \leq n$, is of one of the following forms: $P_i \in N_C$, $P_i = \neg P$ for some $P \in N_C$, or $P_i = \bot$.*

The empty label corresponds to the top-concept.

Just as for $\mathcal{EL}$ and $\mathcal{FLE}$, there exists a 1-1 correspondence between $\mathcal{ALE}$-concept descriptions and $\mathcal{ALE}$-description trees. Every $\mathcal{ALE}$-concept description C can be written (modulo equivalence) as $C \equiv P_1 \sqcap \ldots \sqcap P_n \sqcap \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_m.C_m \sqcap \forall s_1.D_1 \sqcap \ldots \sqcap \forall s_k.D_k$ with $P_i \in N_C \cup \{\neg P \mid P \in N_C\} \cup \{\top, \bot\}$. Such a concept description can now be translated into an $\mathcal{ALE}$-description tree $\mathcal{G}(C) = (V, E, v_0, \ell)$ by induction on the role depth of C analogous to the translation of $\mathcal{FLE}$-concept descriptions into $\mathcal{FLE}$-description trees.

Further, every $\mathcal{ALE}$-description tree $\mathcal{G} = (V, E, v_0, \ell)$ can be translated into an $\mathcal{ALE}$-concept description $C_{\mathcal{G}}$ analogous to the translation of $\mathcal{FLE}$-description trees into $\mathcal{FLE}$-concept descriptions, and the semantics of an $\mathcal{ALE}$-description tree $\mathcal{G}$ is given by the semantics of its corresponding $\mathcal{ALE}$-concept description $C_{\mathcal{G}}$.

Since $\mathcal{ALE}$ is an extension of $\mathcal{FLE}$, and since we are again interested in a characterization of subsumption by homomorphisms, we must normalize $\mathcal{ALE}$-concept descriptions before translating them into their corresponding $\mathcal{ALE}$-description trees. In addition to the normalization rules for $\mathcal{FLE}$, we need three more rules, which deal with the fact that $\mathcal{ALE}$-concept descriptions may contain inconsistent sub-descriptions (e.g., $\bot$ and $P \sqcap \neg P$ for $P \in N_C$):

Definition 34 ($\mathcal{ALE}$ normalization rules) *Let E, F be two $\mathcal{ALE}$-concept descriptions and $r \in N_R$ a primitive role. The $\mathcal{ALE}$-*normalization rules *are defined as follows*

$$\begin{array}{rcl}
\forall r.E \sqcap \forall r.F & \longrightarrow & \forall r.(E \sqcap F) \\
\forall r.E \sqcap \exists r.F & \longrightarrow & \forall r.E \sqcap \exists r.(E \sqcap F) \\
\forall r.\top & \longrightarrow & \top \\
E \sqcap \top & \longrightarrow & E \\
P \sqcap \neg P & \longrightarrow & \bot, \textit{ for each } P \in N_C \\
\exists r.\bot & \longrightarrow & \bot \\
E \sqcap \bot & \longrightarrow & \bot
\end{array}$$

Starting with an $\mathcal{ALE}$-concept description C, the exhaustive application of these rules yields an equivalent $\mathcal{ALE}$-concept description in normal form, which is used to construct the *$\mathcal{ALE}$-description tree corresponding to C*.

In addition to the conditions for $\mathcal{FLE}$-description trees (see Lemma 21), the $\mathcal{ALE}$-description trees obtained this way satisfy the following condition.

Lemma 35 *Let C be an $\mathcal{ALE}$-concept description and $\mathcal{G}_C = (V, E, v_0, \ell)$ its corresponding $\mathcal{ALE}$-description tree. If the label of a node $v \in V$ contains $\bot$, then we have $\ell(v) = \{\bot\}$ and it is either $v = v_0$ or v is a leaf that is reached from its predecessor by an edge with label $\forall r$ for some $r \in N_R$.*

Proof: For an $\mathcal{ALE}$-concept description C in normal form we have either $C = \bot$ or $\bot$ only occurs in a sub-description of C of the form $\forall r.\bot$ for some $r \in N_R$ (see the $\mathcal{ALE}$-normalization rules). Hence, the $\mathcal{ALE}$-description tree corresponding to C has the property formalized in Lemma 35. □

Besides the notion of the canonical interpretation $\mathcal{I}_C$ we also need the notion of the *extended canonical interpretation* $\mathcal{J}_C$ of C in order to prove soundness and completeness of the characterization of subsumption in $\mathcal{ALE}$ by homomorphisms between $\mathcal{ALE}$-description trees.

Definition 36 ((Extended) Canonical interpretation)
Let C be an $\mathcal{ALE}$-concept description and $\mathcal{G}_C = (V, E, v_0, \ell)$ the corresponding $\mathcal{ALE}$-description tree. The canonical interpretation $\mathcal{I}_C = (\Delta, \cdot^{\mathcal{I}_C})$ *of C is defined as*

- $\Delta := V$,
- $P^{\mathcal{I}_C} := \{v \in V \mid P \in \ell(v)\}$ *for all* $P \in N_C$, *and*
- $r^{\mathcal{I}_C} := \{(v, w) \mid (vrw \in E)\}$ *for* $r \in N_R$.

The extended canonical interpretation *of C is defined by* $\mathcal{J}_C := (\Delta_{\mathcal{J}_C}, \cdot^{\mathcal{J}_C})$ *with*

- $\Delta_{\mathcal{J}_C} := V$,
- $P^{\mathcal{J}_C} := \{v \in V \mid P \in \ell(v)\}$ *for all* $P \in N_C$, *and*
- $r^{\mathcal{J}_C} := \{(v, w) \mid (vrw \in E) \text{ or } (v\forall rw \in E \text{ and } \ell(w) \neq \{\bot\})\}$ *for all* $r \in N_R$.

The extended canonical interpretation of C differs from the canonical interpretation of C only in that the primitive roles $r \in N_R$ are interpreted not only by edges in $\mathcal{G}_C$ of the form vrw but also by those of the form $v\forall rw$.

Lemma 37 *Let C be an $\mathcal{ALE}$-concept description, $\mathcal{G}_C = (V, E, v_0, \ell)$ the corresponding $\mathcal{ALE}$-description tree, $\mathcal{I}_C = (\Delta, \cdot^{\mathcal{I}_C})$ its canonical interpretation, and $\mathcal{J}_C = (\Delta_{\mathcal{J}_C}, \cdot^{\mathcal{J}_C})$ the extended canonical interpretation of C. If $\ell(v_0) \neq \{\bot\}$, then $v_0 \in C^{\mathcal{I}_C}$ and $v_0 \in C^{\mathcal{J}_C}$.*

Proof: Let C be an $\mathcal{ALE}$-concept description and $\mathcal{G}_C = (V, E, v_0, \ell)$ the corresponding description tree where $\ell(v_0) \neq \{\bot\}$. In order to show that $v_0 \in C^{\mathcal{I}_C}$ and $v_0 \in C^{\mathcal{J}_C}$ it is sufficient to show that $v_0 \in C_{\mathcal{G}_C}^{\mathcal{I}_C}$ and $v_0 \in C_{\mathcal{G}_C}^{\mathcal{J}_C}$. Therefore, we show by induction on the depth of $\mathcal{G}_C$ that $v \in C_{\mathcal{G}_C(v)}^{\mathcal{I}_C}$ and $v \in C_{\mathcal{G}_C(v)}^{\mathcal{J}_C}$ for each $v \in V$ with $\ell(v) \neq \{\bot\}$. This implies $v_0 \in C_{\mathcal{G}_C}^{\mathcal{I}_C} = C^{\mathcal{I}_C}$ and $v_0 \in C_{\mathcal{G}_C}^{\mathcal{J}_C} = C^{\mathcal{J}_C}$.

Base case: $\mathsf{depth}(\mathcal{G}_C) = 0$
We have $\mathcal{G}_C = (V, \emptyset, v_0, \ell)$. If $\ell(v_0) = \emptyset$, then $C_{\mathcal{G}_C} = \top \equiv C$ and $v_0 \in C^{\mathcal{I}_C} = \Delta = \Delta_{\mathcal{J}_C} = \{v_0\}$. Now assume $\ell(v_0) = \{P_1, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m\} \neq \emptyset$. Since $\ell(v_0) \neq \{\bot\}$, by Lemma 35 we know that $P_i \neq \bot$ for all $1 \leq i \leq n$ and that there is no i, j such that $P_i = Q_j$. Thus, by the definition of $\mathcal{I}_C$ and $\mathcal{J}_C$ it is $v_0 \in P_i^{\mathcal{I}_C} = P_i^{\mathcal{J}_C}$ for all $1 \leq i \leq n$ and $v_0 \notin Q_i^{\mathcal{I}_C} = Q_i^{\mathcal{J}_C}$ for all $1 \leq i \leq m$. This shows $v_0 \in C_{\mathcal{G}_C}^{\mathcal{I}_C} = C^{\mathcal{I}_C}$ and $v_0 \in C_{\mathcal{G}_C}^{\mathcal{J}_C} = C^{\mathcal{J}_C}$.

Induction step: $\mathsf{depth}(\mathcal{G}_C) > 0$
Let $\mathcal{G}_C = (V, E, v_0, \ell)$ and $C_{\mathcal{G}_C} = P_1 \sqcap \ldots \sqcap P_n \sqcap \neg Q_1 \sqcap \ldots \sqcap \neg Q_m \sqcap \exists r_1.C_1 \sqcap \ldots \sqcap \exists r_l.C_l \sqcap \forall s_1.D_1 \sqcap \ldots \sqcap \forall s_k.D_k$. We show that $v_0 \in C'^{\mathcal{I}_C}$ and $v_0 \in C'^{\mathcal{J}_C}$ for each conjunct C' occurring in the conjunction on top-level of $C_{\mathcal{G}_C}$.

Let $C' = P_1 \sqcap \ldots \sqcap P_n \sqcap \neg Q_1 \sqcap \ldots \sqcap \neg Q_m$. As in the base case we can show $v_0 \in C'^{\mathcal{I}_C}$ and $v_0 \notin C'^{\mathcal{J}_C}$.

Now, consider $\exists r_i.C_i$ for some $1 \leq i \leq l$. By definition of $C_{\mathcal{G}_C}$, there exists $v_i \in V$ such that $v_0 r_i v_i \in E$ and $C_{\mathcal{G}_C(v_i)} = C_i$. Due to Lemma 35 we get $\ell(v_i) \neq \{\bot\}$. By induction it is $v_i \in C_{\mathcal{G}_C(v_i)}^{\mathcal{I}_C}$ and $v_i \in C_{\mathcal{G}_C(v_i)}^{\mathcal{J}_C}$. Hence, $v_0 \in (\exists r_i.C_i)^{\mathcal{I}_C}$ and $v_0 \in (\exists r_i.C_i)^{\mathcal{J}_C}$.

Consider $\forall s_j.D_j$ for some $1 \leq j \leq k$. By definition of $C_{\mathcal{G}_C}$ there exists $v_j \in V$ such that $v_0 \forall s_j v_j \in E$ and $C_{\mathcal{G}_C(v_j)} = D_j$. We distinguish two cases:

1. $\ell(v_j) = \{\bot\}$. Then it is $s_j \notin \{r_1, \ldots, r_m\}$, because otherwise, the application of the $\mathcal{ALE}$-normalization rules would yield $\bot$ as the normal form of C and hence, we would have $\ell(v_0) = \{\bot\}$, which contradicts the assumption $\ell(v_0 \neq \{\bot\}$. Now, by definition of $\mathcal{I}_C$ and $\mathcal{J}_C$, there exists no $v \in V$ such that $(v_0, v) \in s_j^{\mathcal{I}_C}$ or $(v_0, v) \in s_j^{\mathcal{J}_C}$. Consequently, $v_0 \in (\forall s_j.\bot)^{\mathcal{I}_C}$ and $v_0 \in (\forall s_j.\bot)^{\mathcal{J}_C}$.

2. $\ell(v) \neq \{\bot\}$. First, assume that $(v_0, v) \in s_j^{\mathcal{I}_C}$. By definition of $\mathcal{I}_C$, it follows that $v_0 s_j v \in E$. Lemma 21.3[6] yields $C_{\mathcal{G}_C(v)} \sqsubseteq C_{\mathcal{G}_C(v_j)}$. By induction we have $v \in C_{\mathcal{G}_C(v)}^{\mathcal{I}_C} \subseteq C_{\mathcal{G}_C(v_j)}^{\mathcal{I}_C}$, and thus we have shown $v_0 \in (\forall s_j.D_j)^{\mathcal{I}_C}$.

[6] More precisely, the extension of Lemma 21 to $\mathcal{ALE}$.

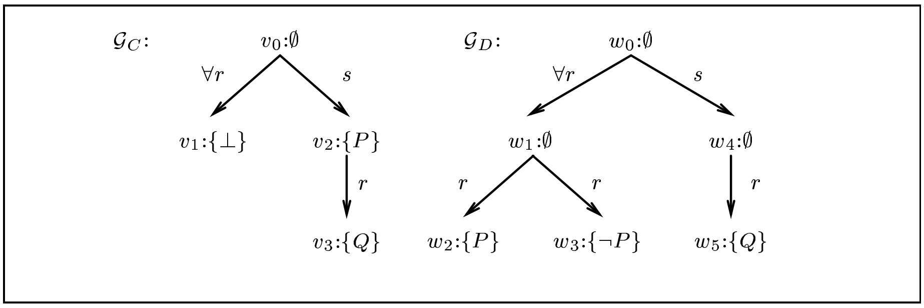

Figure 10: $\mathcal{ALE}$-description trees.

Second, assume that $(v_0, v) \in s_j^{\mathcal{J}_C}$. By definition of $\mathcal{J}_C$, $v_0 s_j v \in E$ or $v_0 \forall s_j v \in E$. Assume $v_0 s_j v \in E$. Analogously to the proof for $\mathcal{I}_C$ it follows that $v \in D_j^{\mathcal{J}_C}$. Assume $v_0 \forall s_j v \in E$, i.e., $v = v_j$ (see Lemma 21.2). Since $D_j = C_{\mathcal{G}_C(v_j)}$, the induction hypothesis yields $v_j \in D_j^{\mathcal{J}_C}$ and thus we have again shown $v_0 \in (\forall s_j.D_j)^{\mathcal{J}_C}$.

This completes the proof of Lemma 37. □

In the next step, we must adapt the notion of a homomorphism appropriately. Unfortunately, the straightforward adaptation of the notion of a homomorphism from $\mathcal{FLE}$-description trees to $\mathcal{ALE}$-description trees does not yield a sound and complete characterization of subsumption in $\mathcal{ALE}$.

Example 38 Consider the following $\mathcal{ALE}$-concept descriptions:

$$\begin{aligned} C &:= (\forall r.\exists r.(P \sqcap \neg P)) \sqcap (\exists s.(P \sqcap \exists r.Q)), \\ D &:= (\forall r.(\exists r.P \sqcap \exists r.\neg P)) \sqcap (\exists s.\exists r.Q). \end{aligned}$$

The description D is already in normal form, and the normal form of C is $C' := \forall r.\bot \sqcap \exists s.(P \sqcap \exists r.Q)$. The corresponding $\mathcal{ALE}$-description trees $\mathcal{G}_C$ and $\mathcal{G}_D$ are depicted in Figure 10.

It is easy to see that there does not exist a homomorphism (in the sense of Section 4) from $\mathcal{G}_D$ to $\mathcal{G}_C$, although we have $C \sqsubseteq D$. In particular, the $\mathcal{ALE}$-concept description $\exists r.P \sqcap \exists r.\neg P$ corresponding to the subtree with root w_1 of $\mathcal{G}_D$ subsumes $\bot$, which is the concept description corresponding to the subtree with root v_1 in $\mathcal{G}_C$. Therefore, a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$ should be allowed to map the whole tree corresponding to $\exists r.P \sqcap \exists r.\neg P$, i.e., the nodes w_1, w_2, w_3, onto the tree corresponding to $\bot$, i.e., onto v_1.

In general, if a node w in $\mathcal{H}$ is mapped onto v in $\mathcal{G}$ with $\ell_G(v) = \{\bot\}$, then all nodes in the subtree with root w in $\mathcal{H}$ must also be mapped onto the node v in $\mathcal{G}$.

Definition 39 (Homomorphisms between $\mathcal{ALE}$-description trees)
A homomorphism *from an $\mathcal{ALE}$-description tree $\mathcal{H} = (V_H, E_H, w_0, \ell_H)$ to an $\mathcal{ALE}$-description tree $\mathcal{G} = (V_G, E_G, v_0, \ell_G)$ is a mapping $\varphi : V_H \longrightarrow V_G$ such that*

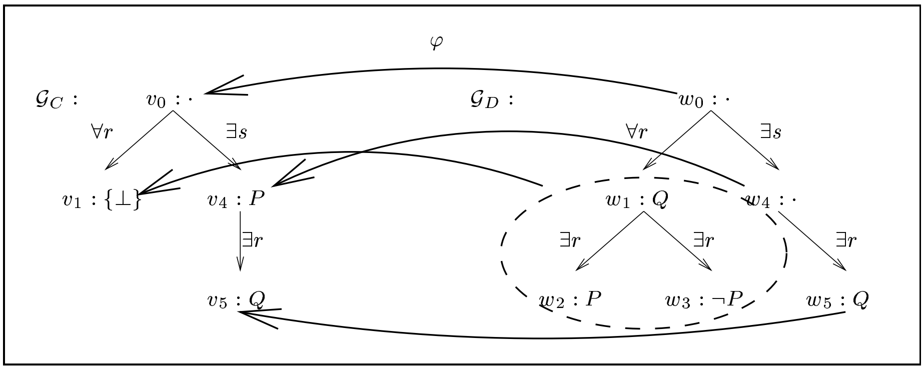

Figure 11: Subsumption for $\mathcal{ALE}$.

1. *$\varphi(w_0) = v_0$,*

2. *for all $v \in V_H$ we have $\ell_H(v) \subseteq \ell_G(\varphi(v))$ or $\ell_G(\varphi(v)) = \{\bot\}$,*

3. *for all $vrw \in E_H$, either $\varphi(v)r\varphi(w) \in E_G$, or $\varphi(v) = \varphi(w)$ and $\ell_G(\varphi(v)) = \{\bot\}$, and*

4. *for all $v\forall rw \in E_H$, either $\varphi(v)\forall r\varphi(w) \in E_G$, or $\varphi(v) = \varphi(w)$ and $\ell_G(\varphi(v)) = \{\bot\}$.*

Example 40 (Example 38 continued)
Consider the $\mathcal{ALE}$-concept descriptions C and D from Example 38 again. As already mentioned, it holds that $C \sqsubseteq D$. A depiction of a homomorphism φ from $\mathcal{G}_D$ to $\mathcal{G}_C$ is given in Figure 11. Note that each node of the subtree $\mathcal{G}_D(w_1)$ is mapped onto v_1.

Theorem 41 *Let C, D be $\mathcal{ALE}$-concept descriptions and $\mathcal{G}_C, \mathcal{G}_D$ the corresponding $\mathcal{ALE}$-description trees. Then $C \sqsubseteq D$ iff there exists a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$.*

Proof: The proof is very similar to the proof of Theorem 24. Again, we first prove the if-direction by induction on the number $|V_D|$ of nodes in $\mathcal{G}_D$.

Let $\mathcal{G}_C = (V_C, E_C, v_0, \ell_C)$, $\mathcal{G}_D = (V_D, E_D, w_0, \ell_D)$, φ a homomorphism from $\mathcal{G}_D$ into $\mathcal{G}_C$, and $\mathcal{I} = (\Delta, \cdot^{\mathcal{I}})$ an arbitrary interpretation. We show that $x_0 \in C^{\mathcal{I}}$ implies $x_0 \in D^{\mathcal{I}}$. Assume $x_0 \in C^{\mathcal{I}}$. Thus, $C \not\equiv \bot$ and hence $\ell_C(v_0) \neq \{\bot\}$ (see Definition 34 and Lemma 35).

$|V_D| = 1$: In this case, $V_D = \{w_0\}$ and $C_{\mathcal{G}_D} = \bigsqcap\limits_{P \in \ell_D(w_0)} P$. Since $C \sqsubseteq D$, we have $\ell_D(w_0) \subseteq \ell_C(v_0)$. This means $x_0 \in P^I$ for all $P \in l_C(v_0)$ and $x_0 \notin Q^I$ for all $\neg Q \in l_C(v_0)$. Hence, $x_0 \in P^I$ for all $P \in l_D(w_0)$ and $x_0 \notin Q^I$ for all $\neg Q \in l_D(w_0)$. Thus, $x_0 \in (\bigsqcap\limits_{P \in \ell_D(w_0)} P)^{\mathcal{I}} = C^{\mathcal{I}}_{\mathcal{G}_D} = D^{\mathcal{I}}$.

Note that $\bot \notin l_D(w_0)$, since otherwise $\bot \in l_C(v_0)$, which is a contradiction to $x_0 \in C^{\mathcal{I}}$.

$|V_D| > 1$: In this case, D is of the form $D \equiv C_{\mathcal{G}_D} = \bigsqcap_{P \in \ell_D(w_0)} P \sqcap \bigsqcap_{\neg Q \in \ell_D(w_0)} \neg Q \sqcap \bigsqcap_{w_0 r w \in E_D} \exists r.C_{\mathcal{G}_D(w)} \sqcap \bigsqcap_{w_0 \forall r w \in E_D} \forall r.C_{\mathcal{G}_D(w)}$.

- As in the case $|V_D| = 1$, it follows that $x_0 \in P^I$ for all $P \in \ell_D(w_0)$ and $x_0 \notin Q^I$ for all $\neg Q \in \ell_D(w_0)$. Again, we know that $\bot \notin \ell_D(w_0)$ since $\ell_C(v_0) \neq \{\bot\}$.
- Let $w_0 r w$ in E_D. Then it is $v_0 r \varphi(w) \in E_C$ and the homomorphism φ restricted to the nodes in $\mathcal{G}_D(w)$ is a homomorphism from $\mathcal{G}_D(w)$ to $\mathcal{G}_C(\varphi(w))$. By induction we know that $C_{\mathcal{G}_C(\varphi(w))} \sqsubseteq C_{\mathcal{G}_D(w)}$. Since $v_0 r \varphi(w) \in E_C$, there is a concept description of the form $\exists r.C_{\mathcal{G}_C(\varphi(w))}$ in the conjunction on top-level of $C_{\mathcal{G}_C}$. Thus, there exists $y \in \Delta$ with $(x_0, y) \in r^I$ and $y \in \mathcal{G}_C^I(\varphi(w))$. It follows that $y \in C^I_{\mathcal{G}_D(w)}$, and $x_0 \in (\exists r.C_{\mathcal{G}_D(w)})^I$.
- Let $w_0 \forall r w \in E_D$. As before, we know $v_0 \forall r \varphi(w) \in E_C$ and by induction $C_{\mathcal{G}_C(\varphi(w))} \sqsubseteq C_{\mathcal{G}_D(w)}$. Furthermore, $\forall r.C_{\mathcal{G}_C(\varphi(w))}$ is a subconcept occurring in the conjunction on top-level of $C_{\mathcal{G}_C}$. Thus, $(x_0, y) \in r^{\mathcal{I}}$ implies $y \in C^{\mathcal{I}}_{\mathcal{G}_C(\varphi(w))} \subseteq C^{\mathcal{I}}_{\mathcal{G}_D(w)}$ and hence $x_0 \in \forall r.C^{\mathcal{I}}_{\mathcal{G}_D(w)}$.

Thus, x_0 is an instance of each conjunct occurring in the conjunction on top-level of $C_{\mathcal{G}_D}$ and hence $x_0 \in C^{\mathcal{I}}_{\mathcal{G}_D} = D^{\mathcal{I}}$.

Next we show the only-if-direction of Theorem 41, i.e., that $C \sqsubseteq D$ implies that there exists a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$. Thus, we assume $C \sqsubseteq D$. Again, we distinguish two cases, namely, $C \equiv \bot$ and $C \not\equiv \bot$.

Let $C \equiv \bot$, C' the normal form of C, and $\mathcal{G}_C = (\{v_0\}, \emptyset, v_0, \ell_C)$. Then we have $C' = \bot$, since otherwise, we obtain a contradiction to $C \equiv \bot$ as follows. Assume $C' \neq \bot$. Then Lemma 35 yields $\ell_C(v_0) \neq \{\bot\}$. By Lemma 37 we can deduce $v_0 \in C^{\mathcal{I}_C}$ and hence a contradiction to $C \equiv \bot$.

Thus, we have $\ell_C(v_0) = \{\bot\}$. Obviously, the mapping φ defined by $\varphi(w) := v_0$ for all $w \in V_D$ yields a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$.

Now, let $C \not\equiv \bot$. This implies $D \not\equiv \bot$, and thus, $\ell_D(w_0) \neq \{\bot\}$. We will construct a homomorphism from $\mathcal{G}_D$ into $\mathcal{G}_C$ by induction on the depth of D.

Base case: $\mathsf{depth}(D) = 0$
Then $D = P_1 \sqcap \ldots \sqcap P_n \sqcap \neg Q_1 \sqcap \ldots \sqcap \neg Q_m$ with $P_i \in N_C$ for all $1 \leq i \leq n$ and $Q_j \in N_C$ for all $1 \leq j \leq m$. Further, $\mathcal{G}_D = (\{w_0\}, \emptyset, w_0, \ell_D)$ with $\ell_D(w_0) = \{P_1, \ldots, P_n, \neg Q_1 \ldots, \neg Q_t\}$. From $C \sqsubseteq D$ and $v_0 \in C^{\mathcal{I}_C}$ (Lemma 37) we can conclude $v_0 \in D^{\mathcal{I}_C}$. The definition of $\mathcal{I}_C$ yields $P_i \in \ell_C(v_0)$ for all $1 \leq i \leq n$. Assume that $\neg Q_j \notin \ell_C(v_0)$ for some $1 \leq j \leq m$. We define an extension of $\mathcal{I}_C = (\Delta, \cdot^{\mathcal{I}_C})$ as follows: $\mathcal{J} := (\Delta_{\mathcal{J}}, \cdot^{\mathcal{J}})$ with $\Delta_{\mathcal{J}} := \Delta$, $r^{\mathcal{J}} := r^{\mathcal{I}_C}$, $P^{\mathcal{J}} := P^{\mathcal{I}_C}$ for $P \neq Q_j$ and $Q_j^{\mathcal{J}} := Q^{\mathcal{I}_C} \cup \{v_0\}$. We still have $v_0 \in C^{\mathcal{J}}$, but $v_0 \notin D^{\mathcal{J}}$ which yields a contradiction to $C \sqsubseteq D$. Consequently, $\ell_D(w_0) \subseteq \ell_C(v_0)$, and thus $\varphi : \{w_0 \longmapsto v_0\}$ is a homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$.

Induction step: $\mathsf{depth}(D) > 0$
Let $D = P_1 \sqcap \ldots \sqcap P_n \sqcap \neg Q_1 \sqcap \ldots \sqcap \neg Q_t \sqcap \exists r_1.D_1 \sqcap \ldots \sqcap \exists r_m.D_m \sqcap \forall s_1.F_1 \sqcap \ldots \sqcap \forall s_k.F_k$ and $\mathcal{G}_D = (V_D, E_D, w_0, \ell_D)$. In order to define a homomorphism φ from $\mathcal{G}_D$ to $\mathcal{G}_C$, we first consider the condition on the roots v_0 and w_0 and then the conditions on r- and $\forall r$-successors of w_0 in $\mathcal{G}_D$.

As in the case $\mathsf{depth}(D) = 0$ we get $\ell_D(w_0) \subseteq \ell_C(v_0)$.

Let $w_0 r w \in E_D$. We must show that there exists a node $v \in V_C$ with $v_0 r v \in E_C$ and $C_{\mathcal{G}_C(v)} \sqsubseteq C_{\mathcal{G}_D(w)}$. Lemma 35 yields $\ell_D(w) \neq \{\bot\}$ as well as $\ell_C(v) \neq \{\bot\}$ for all r-successors v of v_0 in $\mathcal{G}_C$. As in the proof of the only-if-direction of Theorem 24 we obtain a node $v \in V_C$ such that $v_0 r v \in E_C$ and $C_{\mathcal{G}_C(v)} \sqsubseteq C_{\mathcal{G}_D(w)}$. By induction and the extension of Lemma 21 to $\mathcal{ALE}$, there exists a homomorphism φ_w from $\mathcal{G}_D(w)$ to $\mathcal{G}_C(v)$.

Now, let $w_0 \forall s w \in E_D$. We must show that there exists a node $v \in V_C$ such that $v_0 \forall s v \in E_C$ and $C_{\mathcal{G}_C(v)} \sqsubseteq C_{\mathcal{G}_D(w)}$. We distinguish two cases.

1. Assume that $\ell_D(w) = \{\bot\}$. We show that there exists a node $v \in V_C$ with $v_0 \forall s v \in E_C$ and $\ell_C(v) = \{\bot\}$.

   - Assume that there exists no $v \in V_C$ with $v_0 \forall s v \in E_C$. We define an extension $\mathcal{J} := (\Delta_J, \cdot^J)$ of $\mathcal{I}_C = (\Delta, \cdot^{\mathcal{I}_C})$ with $\Delta_J := \Delta \cup \{x\}$ for an $x \notin \Delta$, $P^{\mathcal{J}} := P^{\mathcal{I}_C}$, $r^{\mathcal{J}} := r^{\mathcal{I}_C}$ for $r \neq s$, and $s^{\mathcal{J}} := s^{\mathcal{I}_C} \cup \{(v_0, x)\}$. Then we still have $v_0 \in C^{\mathcal{J}}$, but $v_0 \notin D^{\mathcal{J}}$ which is a contradiction to $C \sqsubseteq D$.
   - Assume that there exists a node $v \in V_C$ such that $v_0 \forall s v \in E_C$ and $\ell_C(v) \neq \{\bot\}$. Then the extended canonical interpretation $\mathcal{J}_C$ satisfies that $v_0 \in C^{\mathcal{J}_C}$ (Lemma 37), but $v_0 \notin D^{\mathcal{J}_C}$ since $(v_0, v) \in s^{\mathcal{J}_C}$. This, again, contradicts the assumption that $C \sqsubseteq D$.

   Consequently, there exists a node $v \in V_C$ with $v_0 \forall s v \in E_C$ and $\ell_C(v) = \{\bot\}$. By Lemma 35 we get that there neither exist successors of v in $\mathcal{G}_C$ nor of w in $\mathcal{G}_D$. Thus, $\varphi_w : \{w \longmapsto v\}$ is a homomorphism from $\mathcal{G}_D(w)$ to $\mathcal{G}_C(v)$.

2. Assume that $\ell_D(w) \neq \{\bot\}$.

   - Assume there exists a node $v \in V_C$ such that $v_0 \forall s v \in E_C$ and $\ell_C(v) = \{\bot\}$. Then φ_w defined by $\varphi_w(w') := v$ for all nodes w' in $\mathcal{G}_D(w)$, is a homomorphism from $\mathcal{G}_D(w)$ to $\mathcal{G}_C(v)$.
   - Assume there exists no $v \in V_C$ such that $v_0 \forall s v \in E_C$ and $\ell_C(v) = \{\bot\}$. Then we have to show that there exists a node $v \in V_C$ such that $v_0 \forall s v \in E_C$ and $C_{\mathcal{G}_C(v)} \sqsubseteq C_{\mathcal{G}_D(w)}$.

     Just as for $\mathcal{FLE}$, we can first show that there exists a node $v \in V_C$ with $v_0 \forall s v \in E_C$. If $\ell_C(v) = \{\bot\}$, then we are in the case already described before. Thus, assume $\ell(v) \neq \{\bot\}$. Now, as in the Proof of Theorem 24, the assumption $C_{\mathcal{G}_C(v)} \not\sqsubseteq C_{\mathcal{G}_D(w)}$ again leads to a contradiction to $C \sqsubseteq D$ by constructing an extension $\mathcal{J}$ of $\mathcal{I}_C$ with

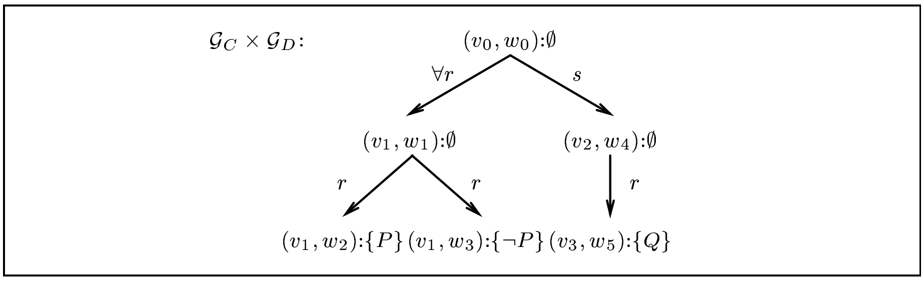

Figure 12: The product of $\mathcal{ALE}$-description trees.

$v_0 \in C^J$ and $v_0 \notin D^J$. Consequently, there exists a node $v \in V_C$ such that $v_0 \forall sv \in E_C$ and $C_{\mathcal{G}_C(v)} \sqsubseteq C_{\mathcal{G}_D(w)}$. By induction and the extension of Lemma 21 to $\mathcal{ALE}$, there exists a homomorphism φ_w from $\mathcal{G}_D(w)$ to $\mathcal{G}_C(v)$.

Finally, we compose the homomorphisms φ_w to a homomorphism φ from $\mathcal{G}_D$ to $\mathcal{G}_C$:

$$\varphi := \{w_0 \mapsto v_0\} \cup \bigcup_{w_0 r w \in E_D} \varphi_w \cup \bigcup_{w_0 \forall r w \in E_D} \varphi_w$$

By construction and since we are concerned with trees, φ is a well-defined homomorphism from $\mathcal{G}_D$ to $\mathcal{G}_C$.

This completes the proof of Theorem 41. □

5.2 The LCS for $\mathcal{ALE}$

The definition of the product of $\mathcal{ALE}$-description trees must be adapted to the modified notion of a homomorphism. In particular, this definition must treat leaves with label $\{\bot\}$ in a special manner. In fact, such a leaf corresponds to the bottom-concept, and since $\bot \sqsubseteq C$ for all $\mathcal{ALE}$-concept descriptions C, we have $\mathsf{lcs}(\bot, C) \equiv C$. Thus, our product operation should be defined such that $C_{\mathcal{G}_\bot \times \mathcal{G}_C} \equiv C$.

Definition 42 (Product of $\mathcal{ALE}$-description trees)
The product $\mathcal{G} \times \mathcal{H}$ of two $\mathcal{ALE}$-description trees $\mathcal{G} = (V_G, E_G, v_0, \ell_G)$ *and* $\mathcal{H} = (V_H, E_H, w_0, \ell_H)$ *is defined as follows. If* $\ell_G(v_0) = \{\bot\}$*, then we define* $\mathcal{G} \times \mathcal{H}$ *by replacing each node* w *in* $\mathcal{H}$ *by* (v_0, w)*, and if* $\ell_H(w_0) = \{\bot\}$*, then we define* $\mathcal{G} \times \mathcal{H}$ *by replacing each node* v *in* $\mathcal{G}$ *by* (v, w_0)*. Otherwise, we define* $\mathcal{G} \times \mathcal{H}$ *by induction on the depth of the trees analogous to the definition of the product of* $\mathcal{FLE}$*-description trees (see Definition 27).*

As an example, the product of the $\mathcal{ALE}$-description trees $\mathcal{G}_C$ and $\mathcal{G}_D$ (see Figure 10) is depicted in Figure 12

Theorem 43 *Let* C, D *be* $\mathcal{ALE}$*-concept descriptions and* $\mathcal{G}_C, \mathcal{G}_D$ *the corresponding* $\mathcal{ALE}$*-description trees. Then,* $C_{\mathcal{G}_C \times \mathcal{G}_D}$ *is the lcs of* C *and* D*.*

Proof: The proof is very similar to the proof of Theorem 11. Again, we have to show that

1. $C \sqsubseteq C_{\mathcal{G}_C \times \mathcal{G}_D}$,
2. $D \sqsubseteq C_{\mathcal{G}_C \times \mathcal{G}_D}$, and
3. for each C' with $C \sqsubseteq C'$ and $D \sqsubseteq C'$, we have $C_{\mathcal{G}_C \times \mathcal{G}_D} \sqsubseteq C'$.

Just as for $\mathcal{EL}$, the projection π_i with $\pi_i(v_1, v_2) := v_i$, $i = 1, 2$, yields a homomorphism from $\mathcal{G}_C \times \mathcal{G}_D = (V, E, (v_0, w_0), \ell)$ to $\mathcal{G}_C$ for $i = 1$ and to $\mathcal{G}_D$ for $i = 2$. Note that for each node v in $\mathcal{G}_C$ (v in $\mathcal{G}_D$) with $\ell_C(v) = \{\bot\}$ ($\ell_D(v) = \{\bot\}$) the subtree consisting of all nodes of the form (v, w) (if $i = 1$) or (w, v) (if $i = 2$) in $\mathcal{G}_C \times \mathcal{G}_D$ is mapped onto v in $\mathcal{G}_C$ ($\mathcal{G}_D$) by π_1 (π_2). By Theorem 41 we can conclude that $C \sqsubseteq C_{\mathcal{G}_C \times \mathcal{G}_D}$ and $D \sqsubseteq C_{\mathcal{G}_C \times \mathcal{G}_D}$.

Now let C' be an arbitrary common subsumer of C, D and $\mathcal{G}' = (V', E', v_0', \ell')$ the corresponding $\mathcal{ALE}$-description tree. Theorem 41 yields homomorphisms φ_1 from $\mathcal{G}_{C'}$ to $\mathcal{G}_C$ and φ_2 from $\mathcal{G}_{C'}$ to $\mathcal{G}_D$. We again define $\varphi(v') := (\varphi_1(v'), \varphi_2(v'))$ for all $v' \in V'$, and prove that (1) φ is well-defined, i.e., $\varphi(v') \in V$ for all $v' \in V'$, and (2) φ is a homomorphism according to Definition 39 from $\mathcal{G}_{C'}$ to $\mathcal{G}_C \times \mathcal{G}_D$.

The first point is again shown by induction on the length $\delta(v')$ of the path in $\mathcal{G}_{C'}$ leading from v_0' to v'.

$\delta(v') = 0$: Then we have $v' = v_0'$ and hence, $\varphi(v_0') = (\varphi_1(v_0'), \varphi_2(v_0')) = (v_0, w_0) \in V$.

$\delta(v') > 0$: Let $v' \in V'$ with $\delta(v') > 0$. Then there exists a unique predecessor $v'' \in V'$ of v', i.e., $v''rv' \in E'$ for some $r \in N_R$ or $v''\forall rv' \in E'$ for some $r \in N_R$. Assume $v''rv' \in E'$ for some $r \in N_R$. (The case $v''\forall rv' \in E'$ can be handled in the same way.) Obviously, we have $\delta(v'') = \delta(v') - 1$. By induction, we get $(\varphi_1(v''), \varphi_2(v'')) \in V$. Now, we distinguish several cases.

1. $\varphi_1(v'')r\varphi_1(v') \in E_C$ and $\varphi_2(v'')r\varphi_2(v') \in E_D$.
   Definition 42 yields $(\varphi_1(v'), \varphi_2(v'))$ as an r-successor of $(\varphi_1(v''), \varphi_2(v''))$ in $\mathcal{G}_C \times \mathcal{G}_D$ and hence, $(\varphi_1(v'), \varphi_2(v')) \in V$.
2. $\ell_C(\varphi_1(v'')) = \{\bot\}$ and $\varphi_2(v'')r\varphi_2(v') \in E_D$.
   Then we have $\varphi_1(v') = \varphi_1(v'')$ and Definition 42 yields $(\varphi_1(v''), \varphi_2(v''))r(\varphi_1(v''), \varphi_2(v')) \in E$ and hence, $(\varphi_1(v'), \varphi_2(v')) \in V$.
3. $\varphi_1(v'')r\varphi_1(v') \in E_C$ and $\ell_D(\varphi_2(v'')) = \{\bot\}$.
   Analogous to 2.
4. $\ell_C(\varphi_1(v'')) = \{\bot\}$ and $\ell_D(\varphi_2(v'')) = \{\bot\}$.
   Then we have $\varphi_1(v') = \varphi_1(v'')$ and $(\varphi_2(v') = \varphi_2(v'')$. Hence, $(\varphi_1(v'), \varphi_2(v')) \in V$.

It remains to show (2).

1. We have $\varphi(v_0') = (\varphi_1(v_0'), \varphi_2(v_0')) = (v_0, w_0)$, because φ_1 (φ_2) is a homomorphism from $\mathcal{G}_{C'}$ to $\mathcal{G}_C$ ($\mathcal{G}_D$).
2. We have to show $\ell'(v') \subseteq \ell(\varphi_1(v'), \varphi_2(v'))$ or $\ell(\varphi_1(v'), \varphi_2(v')) = \{\bot\}$ for all $v' \in V'$. We distinguish several cases.
   (a) $\ell'(v') \subseteq \ell_C(\varphi_1(v'))$ and $\ell'(v') \subseteq \ell_D(\varphi_2(v'))$.
   Then we have $\ell'(v') \subseteq \ell_C(\varphi_1(v')) \cap \ell_D(\varphi_2(v')) = \ell(\varphi(v'))$.
   (b) $\ell'(v') \subseteq \ell_C(\varphi_1(v'))$ and $\ell_D(\varphi_2(v')) = \{\bot\}$.
   By Definition 42 we have $\ell(\varphi_1(v'), \varphi_2(v')) = \ell_C(\varphi_1(v'))$, and hence $\ell'(v') \subseteq \ell(\varphi(v'))$.
   (c) $\ell_C(\varphi_1(v')) = \{\bot\}$ and $\ell'(v') \subseteq \ell_D(\varphi_2(v'))$.
   Analogous to 2.
   (d) $\ell_C(\varphi_1(v')) = \{\bot\}$ and $\ell_D(\varphi_2(v')) = \{\bot\}$.
   By Definition 42 we have $\ell(\varphi_1(v'), \varphi_2(v')) = \{\bot\}$.
3. Let $v'rw' \in E'$. We have to show $\varphi(v')r\varphi(w') \in E$, or $\varphi(v') = \varphi(w')$ and $\ell(\varphi(v')) = \{\bot\}$. We distinguish several cases.
   (a) $\varphi_1(v')r\varphi_1(w') \in E_C$ and $\varphi_2(v')r\varphi_2(w') \in E_D$.
   Due to (1) we have $(\varphi_1(v'), \varphi_2(v')) \in V$ and by Definition 42, we obtain $(\varphi_1(v'), \varphi_2(v'))r(\varphi_1(w'), \varphi_2(w')) \in E$.
   (b) $\varphi_1(v')r\varphi_1(w') \in E_C$ and $\varphi_2(v') = \varphi_2(w')$ and $\ell_D(\varphi(v')) = \{\bot\}$.
   Due to (1) we have $(\varphi_1(v'), \varphi_2(v')) \in V$ and again by Definition 42, we obtain $(\varphi_1(v'), \varphi_2(v'))r(\varphi_1(v'), \varphi_2(w')) \in E$.
   (c) $\varphi_1(v') = \varphi_1(w')$ and $\ell_C(\varphi_1(v')) = \{\bot\}$ and $\varphi_2(v')r\varphi_2(w') \in E_D$.
   Analogous to 2.
   (d) $\varphi_1(v') = \varphi_1(w')$ and $\ell_C(\varphi_1(v')) = \{\bot\}$ and $\varphi_2(v') = \varphi_2(w')$ and $\ell_D(\varphi(v')) = \{\bot\}$.
   Due to (1) we have $(\varphi_1(v'), \varphi_2(v')) \in V$ and by Definition 42, we get $\ell(\varphi_1(v'), \varphi_2(v')) = \{\bot\}$.
4. Let $v'\forall rw' \in E'$. As in the previous case, we can show $\varphi(v')\forall r\varphi(w') \in E$, or $\varphi(v') = \varphi(w')$ and $\ell(\varphi(v')) = \{\bot\}$.

By Theorem 41 we have $C_{\mathcal{G}_C \times \mathcal{G}_D} \sqsubseteq C'$. This completes the proof of Theorem 43. □

On the complexity of least common subsumer in $\mathcal{ALE}$

The results on the complexity of lcs in $\mathcal{FLE}$ also hold for $\mathcal{ALE}$. More precisely, if we refer to the concept descriptions C_n and D_n defined in Example 30 as $\mathcal{ALE}$-concept descriptions, then $C_{\mathcal{B}_n}$ also yields a minimal representation of the lcs of C_n and D_n in $\mathcal{ALE}$, i.e., there does not exist an $\mathcal{ALE}$-concept description C' such that $C' \equiv \mathcal{B}_n$ and $|C'| < |\mathcal{B}_n|$.

The size of the $\mathcal{ALE}$-description tree $\mathcal{G}_C$ corresponding to an $\mathcal{ALE}$-concept description C can, as for $\mathcal{FLE}$, be at most exponential in the size of C. Thus, the size of the lcs of n $\mathcal{ALE}$-concept descriptions can again be bounded by an exponential function.

Proposition 44 *The lcs of* two *$\mathcal{ALE}$-concept descriptions C, D may be of size exponential in the size of C and D.*

The size of the lcs of n $\mathcal{ALE}$-concept descriptions $C_1, \ldots, C_n$ can be bounded by an exponential function with $M := \mathsf{max}\{|C_1|, \ldots, |C_n|\}$ and n as its inputs.

6 Conclusion and future work

We have described a method for computing the least common subsumer in the description logic $\mathcal{ALE}$. In the worst case, the result of this computation may be exponential in the size of the input descriptions. However, the examples that show this exponential behavior [3] are rather artificial, and thus we believe that this complexity will not pose a problem in practice.

Our method depends on the characterization of subsumption by homomorphisms on description trees, because this allows us to construct the lcs as the product of the description trees. For sub-languages of CLASSIC, a similar method has been used to construct the lcs [10, 11, 13], even though the characterization of subsumption (via a structural subsumption algorithm [6]) is not explicitly given in terms of homomorphisms. The main difference is that these languages do not allow for existential restrictions. The results for simple conceptual graphs and conjunctive queries mentioned below Theorem 7 characterize subsumption (resp. containment) with the help of homomorphisms, but they do not consider the lcs, and they cannot handle value restrictions.

The language $\mathcal{ALE}$ is expressive enough to be quite useful in our process engineering application. In fact, the descriptions of standard building blocks of process models that we currently represent in our DL system can all be expressed within this language. However, in order to support the "bottom-up" approach for constructing knowledge bases outlined in the introduction, we must also be able to compute the most specific concept for individuals. Unfortunately, the msc need not always exist in $\mathcal{ALE}$. For the DL $\mathcal{ALN}$, it was shown in [2] that this problem can be overcome by allowing for cyclic concept description, but $\mathcal{ALN}$ does not allow for existential restrictions. Thus, we must either extend the approach of [2] to $\mathcal{ALE}$, or resort to an approximation of the msc, as proposed in [11]. In the process engineering application, we can also use the lcs operation directly to structure the existing knowledge base. In fact, it has turned out that the subsumption hierarchy obtained from the knowledge base of standard building blocks is rather flat. To obtain a deeper hierarchy (which better supports search), we will try to construct intermediate levels of concepts by applying the lcs operation. Of course, this only makes sense if the lcs yields concepts that have an intuitive meaning in the application domain.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.

[2] F. Baader and R. Küsters. Computing the least common subsumer and the most specific concept in the presence of cyclic $\mathcal{ALN}$-concept descriptions. In O. Herzog and A. Günter, editors, *Proceedings of the 22nd Annual German Conference on Artificial Intelligence (KI'98)*, volume 1504 of *Lecture Notes in Computer Science*, pages 129–140, Bremen, Germany, 1998. Springer–Verlag.

[3] F. Baader, R. Küsters, and R. Molitor. Computing least common subsumer in description logics with existential restrictions. LTCS-Report 98-09, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 1998. See http://www-lti.informatik.rwth-aachen.de/Forschung/Papers.html.

[4] F. Baader, R. Molitor, and S. Tobies. On the relation between descripion logics and conceptual graphs. LTCS-Report 98-11, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 1998. See http://www-lti.informatik.rwth-aachen.de/Forschung/Papers.html.

[5] F. Baader and U. Sattler. Knowledge representation in process engineering. In *Proceedings of the 1996 International Workshop on Description Logic (DL'96)*, 1996.

[6] A. Borgida and P. Patel-Schneider. A semantics and complete algorithm for subsumption in the CLASSIC description logic. *Journal of Artificial Intelligence Research*, 1:277–308, 1994.

[7] R.J. Brachman, D. McGuinness, P. Patel-Schneider, L. Resnick, and A. Borgida. Living with CLASSIC: When and how to use a KL-ONE-like language. In John F. Sowa, editor, *Principles of Semantic Networks.* Morgan Kaufmann, 1991.

[8] M. Chein, M. L. Mugnier, and G. Simonet. Nested graphs: A graph-based knowledge representation model with fol semantics. In A.G. Cohn, L.Schubert, and S.C. Shapiro, editors, *Proceedings of the 6th International Conference on Knowledge Representation (KR'98)*, pages 524–534, Trento, Italy, June 1998. Morgan Kaufman Publishers Inc.

[9] M. Chein and M.L. Mugnier. Conceptual graphs: Fundamental notions. *Revue d'Intelligence Artificielle*, 6(4):365–406, 1992.

[10] W.W. Cohen, A. Borgida, and H. Hirsh. Computing least common subsumers in description logics. In W. Swartout, editor, *Proceedings of the 10th National Conference on Artificial Intelligence*, pages 754–760, San Jose, CA, July 1992. MIT Press.

[11] W.W. Cohen and H. Hirsh. Learning the CLASSIC description logic: Theoretical and experimental results. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourth International Conference (KR'94)*. Morgan Kaufmann, 1994.

[12] F.M. Donini, M. Lenzerini, D. Nardi, B. Hollunder, W. Nutt, and A.M. Spaccamela. The complexity of existential quantification in concept languages. *Journal of Artificial Intelligence*, 52:309–327, 1992.

[13] M. Frazier and L. Pitt. CLASSIC learning. *Machine Learning*, 25:151–193, 1996.

[14] B.C. Pierce. *Basic Category Theory for Computer Scientists.* Foundations of Computing Series. The MIT Press, Cambridge, Massachusetts, 1991.

[15] S.W. Reyner. An analysis of a good algorithm for the subtree problem. *SIAM Journal of Computing*, 6(4):730–732, December 1977.

[16] U. Sattler. *Terminological knowledge representation systems in a process engineering application.* PhD thesis, RWTH Aachen, 1998.

[17] J.F. Sowa. *Conceptual Structures – Information, Processing in Mind and Machine.* Addison Wesley, 1984.