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

LTCS–Report

Foundations of non-standard inferences for DLs with transitive roles

Sebastian Brandt and Anni-Yasmin Turhan and Ralf Küsters

LTCS-Report 03-02

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

Hans-Grundig-Str. 25
01062 Dresden
Germany

Foundations of non-standard inferences for DLs with transitive roles

Sebastian Brandt and Anni-Yasmin Turhan and Ralf Küsters *

Abstract

Description Logics (DLs) are a family of knowledge representation formalisms used for terminological reasoning. They have a wide range of applications such as medical knowledge-bases, or the semantic web. Research on DLs has been focused on the development of sound and complete inference algorithms to decide satisfiability and subsumption for increasingly expressive DLs. Non-standard inferences are a group of relatively new inference services which provide reasoning support for the building, maintaining, and deployment of DL knowledge-bases. So far, non-standard inferences are not available for very expressive DLs. In this paper we present first results on non-standard inferences for DLs with transitive roles. As a basis, we give a structural characterization of subsumption for DLs where existential and value restrictions can be imposed on transitive roles. We propose sound and complete algorithms to compute the least common subsumer (lcs).

* This work has been supported by the Deutsche Forschungsgemeinschaft, DFG Project BA 1122/4-1.

Contents

1 Introduction and Motivation

Description Logics (DLs) are a family of formalisms used to represent terminological knowledge of a given application domain in a structured and well-defined way. The basic notions of DLs are *concept-descriptions* and *roles*, representing unary predicates and binary relations, respectively. Atomic concepts and concept descriptions represent sets of individuals, whereas roles represent binary relations between individuals [5]. The main characteristic of a DL is the set of concept constructors by which complex concept descriptions can be built from atomic concepts and roles. In the present paper, we are concerned with the DL $\mathcal{FLE}^+$ which provides the constructors conjunction ($C \sqcap D$), existential restriction ($\exists r.C$), value restriction ($\forall r.C$), and the top concept ($\top$).

In $\mathcal{FLE}^+$, a role can be defined transitive. In this case it represents the transitive closure of a binary relation. Transitive roles appear naturally in many application domains, such as medicine and process engineering [1]. Consider, for instance, a machine that comprises several components which again consists of several devices. A natural way to represent such a machine by means of DLs would be to use some *has-part* role to reflect its compositional structure. It would be natural here to implicitly regard every part of a component also as a part of the whole. To this end, a DL with transitive roles is necessary.

Inference problems for DLs are divided into so-called standard and non-standard ones. Well known standard inference problems are satisfiability and subsumption of concept descriptions. These are well investigated for a great range of DLs. For many of them, sound and complete decision procedures could be devised and lower and upper bounds for the computational complexity have been found [11]. Many standard inference algorithms have been successfully extended to cope with transitive roles [13, 12] and are put into practice in state of the art DL Systems.

Prominent non-standard inferences are matching, the least common subsumer (lcs), the most specific concept (msc), and, more recently, approximation. Non-standard inferences resulted from the experience with real-world DL-knowledge bases (KBs), where standard inference algorithms sometimes did not suffice for building and maintaining purposes. For example, the problem of how to structure the application domain by means of concept definitions may not be clear at the beginning of the modelling task. Moreover, the expressive power of the DL under consideration sometimes makes it difficult to come up with a faithful formal definition of the concept originally intended. To alleviate these difficulties it is expedient to employ non-standard inferences [14, 8].

The lcs was first mentioned as an inference problem for DLs in [10]. Given two concept descriptions A and B in a description logic $\mathcal{L}$, the lcs of A and B is defined as the least (w.r.t. subsumption) concept description in $\mathcal{L}$ subsuming A and B. It has been argued in [8] that the lcs facilitates a "bottom-up"-approach to the above mentioned modelling task: a domain expert can select a number of intuitively related concept descriptions already existent in a KB and use the lcs operation to automatically construct a new concept description representing the closest generalization of them. This approach can be extended by means of the msc. Selecting one individual, i.e., an instance of a concept, from a KB the msc constructs the most specific concept expressible in the underlying DL representing the individual. Using this inference, the "bottom-up"-design of new concepts can start on the level of actual individuals which are sometimes more familiar to a domain expert than the more abstract concepts.

Matching in DLs was first proposed in [7]. A matching problem (modulo subsumption) consists of a concept description C and a concept *pattern* D, i.e., a concept description with variables. Matching D against C means finding a substitution of variables in D by concept descriptions such that C is subsumed by the instantiated concept pattern D. Among other applications, matching can be employed for queries in KBs: a domain expert unable to specify uniquely the concept he is looking for in a KB can use a concept pattern to retrieve all those concepts in the KB for which a matcher exists. The structural constraints expressible by patterns exceed the capabilities of simple "wildcards" familiar from ordinary search engines [8].

Approximation was first mentioned as a new inference problem in [4]. The upper (lower) approximation of a concept description C_1 from a DL $\mathcal{L}_1$ is defined as the least (greatest) concept description in another DL $\mathcal{L}_2$ which subsumes (is subsumed by) C_1. Approximation can be used to make non-standard inferences accessible to more expressive DLs by transferring a given inference problem to a less expressive DL where at least an approximate solution can be computed. Another application of approximation lies in user-friendly DL-systems offering a simplified frame-based view on KBs defined in a more expressive background DL [6]. Here approximation can be used to compute simple frame-based representations of otherwise overwhelmingly complicated concept descriptions.

Table 1: Syntax and semantics of $\mathcal{FLE}^+$-concept descriptions.

Construct name	Syntax	Semantics
top-concept	$\top$	$\Delta_\mathcal{I}$
conjunction	$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
existential restrictions	$\exists r.C$	$\{x \in \Delta_\mathcal{I} \mid \exists y : (x, y) \in r^\mathcal{I} \wedge y \in C^\mathcal{I}\}$
value restrictions	$\forall r.C$	$\{x \in \Delta_\mathcal{I} \mid \forall y : (x, y) \in r^\mathcal{I} \rightarrow y \in C^\mathcal{I}\}$
transitive roles	r^+	$\bigcup_{1 \leq n} (r^\mathcal{I})^n$

In contrast to standard inference problems, comparatively little research exists on non-standard inferences in DLs with transitive roles [2]. If existential restrictions can be expressed in a DL then the inferences matching and approximation are defined by means of the lcs operation. This central role of the lcs for non-standard inferences has lead us to make this inference problem the first to be extended to $\mathcal{FLE}^+$. After introducing some basic notions and notation, our first step towards the lcs will be a characterization of subsumption for $\mathcal{FLE}^+$-concept descriptions by means of so-called *description graphs*. We shall see that for two $\mathcal{FLE}^+$-concept descriptions A and B, subsumption ($A \sqsubseteq B$) holds if and only if there exists a simulation relation from the description graph of B into the one of A. The lcs inference of A and B is then defined as the graph product of the respective description graphs.

As a result, we shall see that the lcs of a finite set of $\mathcal{FLE}^+$-concept descriptions always exists and is uniquely determined up to equivalence. Moreover, an effective algorithm for the computation of the lcs will be provided.

2 Preliminaries

DLs are based on the following sets of names: N_C is the set of concept names, and N_R is the set of role names, and N_R^T is the set of transitive roles, where $N_R \cap N_R^T = \emptyset$. Concept descriptions are inductively defined starting from the set of concept names and use the concept constructors shown in Table 1.

The DL $\mathcal{FLE}$ offers the top-concept, conjunction, existential, and value restrictions, as displayed in Table 1. In $\mathcal{FLE}^+$, transitive roles can be used in existential and value restrictions.

As usual, the semantics of a concept description is defined in terms of an

	$\mathcal{FL}_0$	$\mathcal{EL}$	$\mathcal{FLE}$
top-concept	x	x	x
conjunction	x	x	x
existential restrictions		x	x
value restrictions	x		x

Table 2: Description Logics

interpretation $\mathcal{I} = (\Delta, \cdot^I)$. The domain Δ of $\mathcal{I}$ is a non-empty set and the interpretation function $\cdot^I$ maps each concept name $A \in N_C$ to a set $A^I \subseteq \Delta$ and each role name $r \in N_R \cup N_R^T$ to a binary relation $r^I \subseteq \Delta \times \Delta$. The extension of $\cdot^I$ to arbitrary concept descriptions is defined inductively, as shown in the second column of Table 1.

The DLs covered in this paper are extensions of the DLs shown in Table 2. Please note that none of these DLs provides (primitive) negation or the bottom concept and therefore can not express contradictions, thus all concept descriptions build in the above mentioned DLs are satisfiable.

One of the most important traditional inference services provided by DL systems is computing the subsumption hierarchy. The concept description C is *subsumed* by the description D ($C \sqsubseteq D$) iff $C^I \subseteq D^I$ for all interpretations $\mathcal{I}$; C and D are *equivalent* ($C \equiv D$) iff $C \sqsubseteq D$ and $D \sqsubseteq C$.

In this paper we focus on the *non-standard inference* of computing the *least common subsumer (lcs).*

Definition 1 (lcs) *Given $\mathcal{L}$-concept descriptions $C_1, \ldots, C_n$, for some description logic $\mathcal{L}$, the $\mathcal{L}$-concept description C is the* least common subsumer *(lcs) of $C_1, \ldots, C_n$ ($C = \mathsf{lcs}(C_1, \ldots, C_n)$ for short) iff (i) $C_i \sqsubseteq C$ for all $1 \leq i \leq n$, and (ii) C is the least concept description with this property, i.e., if C' satisfies $C_i \sqsubseteq C'$ for all $1 \leq i \leq n$, then $C \sqsubseteq C'$.*

The idea behind the lcs inference is to extract the commonalities of the input concepts. The lcs is uniquely determined up to equivalence. Therefore it is justified to speak about "the" lcs instead of "an" lcs.

3 Least common subsumer for $\mathcal{FL}_0^+$

In a first step the DL $\mathcal{FL}_0$is extended by transitive roles, resulting in $\mathcal{FL}_0^+$. For $\mathcal{FL}_0^+$the propagation of concepts appearing within value restrictions must be guaranteed for transitive roles.

We characterize subsumption of $\mathcal{FL}_0$-concept descriptions by a structural comparison and prove that this characterization is sound and complete. Based on this characterization we develop an algorithm to compute the lcs of two $\mathcal{FL}_0$-concept descriptions.

In order to use a structural comparison to test subsumption one has to make all the information captured in the concept descriptions explicit. In case of $\mathcal{FL}_0^+$-concept descriptions the propagation of value restrictions regarding transitive roles has to be ensured.

3.1 Normalizing $\mathcal{FL}_0^+$-concept descriptions

We follow the approach in [5] and use the following normal form of $\mathcal{FL}_0^+$-concept descriptions.

Definition 2 **($\mathcal{FL}_0^+$-normal form)** *a $\mathcal{FL}_0^+$-concept description is in $\mathcal{FL}_0^+$-*normal form *iff it is either $\top$ or a conjunction of the form $\forall r_1.\cdots.\forall r_n.A$ for $n \geq 0$ role names $r_1, \ldots, r_n \in N_R \cup N_R^T$ and a concept name $A \neq \top, A \in N_C$.*

We abbreviate $\forall r_1.\cdots.\forall r_n.A$ by $\forall r_1.\cdots.r_n.A$ where $r_1 \ldots r_n$ is considered a *role word* over $N_R \cup N_R^T$. In addition, we write $\forall L.C$ instead of $\forall w_1 \ldots w_m.C$, where the *role language* $L = \{w_1, \ldots, w_m\}$ is a finite set of words over $N_R \cup N_R^T$. The term $\forall \emptyset.A$ is considered to be equivalent to $\top$.

Definition 3 *Let $L \subseteq (N_R \cup N_R^T)^+$ be a* role language *and*

$$\widehat{r} := \begin{cases} r, \text{ if } r \in N_R \\ r^+, \text{ if } r \in N_R^T \end{cases}$$

then $\widehat{L} := \{\widehat{r_1}\widehat{r_2}\cdots\widehat{r_n} \mid r_1r_2\cdots r_n \in L\}$ is the transitive role language of *L.*

The interpretation function extends to transitive role languages as captured by the following Lemma.

Lemma 4 *Let A be a $\mathcal{FL}_0^+$-concept description, then*

1. *$d \in (\forall\widehat{L}.A)^\mathcal{I}$ iff $\forall w \in \widehat{L} : dw^\mathcal{I}e$ implies that $e \in A^\mathcal{I}$.*

2. $\forall L.A \equiv \forall \widehat{L}.A$

Proof. proof of 1.): follows directly from the semantics of value restrictions and transitive roles.
proof of 2.): follows directly from definition of $\widehat{L}$.

■

Let us consider the complexity for computing the $\mathcal{FL}_0^+$-normal form. For a concept C with $|C| = n$ the number of different role-words, the length of each role-word, and the number of concept names embedded in the value restrictions can each be bounded by n. Therefore there are at most n different role-words. Each one (of length n in the worst case) has to be copied for each conjunct to obtain value restrictions with only one embedded concept name A. Therefore the $\mathcal{FL}_0^+$-normal form can be computed in polynomial time.

3.2 Characterization of subsumption for $\mathcal{FL}_0^+$

Based on the $\mathcal{FL}_0^+$-normal form we can advise a structural check that determines subsumption between two $\mathcal{FL}_0^+$-concept descriptions. This characterization of subsumption is a prerequisite for the computation of the lcs in $\mathcal{FL}_0^+$. We begin with a theorem that characterizes the subsumption between value restrictions over possibly transitive roles.

Theorem 5 *Let A be a $\mathcal{FL}_0^+$-concept description , then $\forall L.A \sqsubseteq \forall L'.A$ iff $\widehat{L'} \subseteq \widehat{L}$.*

Proof. "→" It holds that $\forall L.A \sqsubseteq \forall L'.A$. We prove the claim by contradiction and assume $\widehat{L'} \not\subseteq \widehat{L}$, then there exists a word $w = r_1 r_2 \cdots r_n$ with $w \in \widehat{L'} \setminus \widehat{L}$. This implies that $(\forall w.A)^\mathcal{I} \not\subseteq (\forall \widehat{L}.A)^\mathcal{I}$ and $(\forall w.A)^\mathcal{I} \subseteq (\forall \widehat{L'}.A)^\mathcal{I}$. Therefore $(\forall \widehat{L}.A)^\mathcal{I} \not\subseteq (\forall \widehat{L'}.A)^\mathcal{I}$ and applying Lemma 4.2 it holds that $(\forall L.A)^\mathcal{I} \not\subseteq (\forall L'.A)^\mathcal{I}$. Consequently, we obtain a contradiction to our initial assumption.
"←" It holds that $\widehat{L'} \subseteq \widehat{L}$. Therefore $w \in \widehat{L'}$ implies $w \in \widehat{L}$. It follows from Lemma 4.1 that, $(\forall \widehat{L}.A)^\mathcal{I} \subseteq (\forall \widehat{L'}.A)^\mathcal{I}$ and thus $\forall \widehat{L}.A \sqsubseteq \forall \widehat{L'}.A$. ■

We need to introduce some notation to access the different parts of a concept description C in $\mathcal{FL}_0^+$-normal form:

- $prim(C)$ denotes the set of all concept names and the top concept occurring on the top-level of C.

- $\mathsf{val}_w(C) := C_1 \sqcap \cdots \sqcap C_n$, if there exist value restrictions of the form $\forall w.C_1, \ldots, \forall w.C_n$ on the top-level of C; otherwise, $\mathsf{val}_w(C) := \top$;

- $L(C)$ denotes the set of role-words appearing in the value restrictions on the top-level of C.

- $L_A(C) = \{w \mid \forall w.A$ occurs on the top-level of $C\}$.

The conditions for subsumption for $\mathcal{FL}_0$can be extended to arbitrary $\mathcal{FL}_0^+$-concept descriptions using Theorem 5.

Theorem 6 *Let C and D be two $\mathcal{FL}_0^+$-concept descriptions in $\mathcal{FL}_0^+$-normal form. Then, $C \sqsubseteq D$ iff $D = \top$, or it holds that*

1. *$prim(D) \subseteq prim(C)$, and*

2. *for all $A \in N_C$: $\widehat{L}_A(D) \subseteq \widehat{L}_A(C)$*

Proof. $\rightarrow$: Assume $C \sqsubseteq D$.

- Assume $prim(D) \not\subseteq prim(C)$. Then there exists an $A \in prim(D) \setminus prim(C)$. As all $\mathcal{FL}_0^+$-concept descriptions C is consistent. We may therefore consider a interpretation I with a canonical model of C. By definition, the individual $d_C \in \Delta^I$ for C does not occur in A^I, since $A \notin prim(C)$. Thus, $d \notin D^I$ and therefore $C \not\sqsubseteq D$, in contradiction to our assumption.

- Assume that there exists $A \in N_C$ with $\widehat{L}_A(D) \not\subseteq \widehat{L}_A(C)$. Thus there exists a role-word $w \in \widehat{L}_A(D)$ and $w \notin \widehat{L}_A(C)$. This implies that $(\forall w.A)^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ and $(\forall w.A)^{\mathcal{I}} \not\subseteq C^{\mathcal{I}}$ and thus obtain a contradiction to our initial assumption.

$\leftarrow$: Conditions 1 and 2 hold.
Assume $C \not\sqsubseteq D$. Due to the normalization C is a conjunction. Due to our assumption there must exist at least one conjunct C_i in C s.t. $C_i \sqsubseteq C$ and $C_i \not\sqsubseteq D$. There are two cases to distinguish:

1. $C_i \in prim(C)$: from $C_i \not\sqsubseteq D$ we can conclude that $C_i \notin prim(D)$. Thus we have a contradiction to our assumption that Condition 1 holds.

Least common subsumer for $\mathcal{FL}_0^+$
Input: Two $\mathcal{FL}_0^+$-concept descriptions C, D.
Output: $\mathcal{FL}_0^+$-lcs of C and D.

1. If $C \sqsubseteq D$ then $\mathsf{c\text{-}lcs}\text{-}\mathcal{FL}_0^+(C,D) := D$
   if $D \sqsubseteq C$ then $\mathsf{c\text{-}lcs}\text{-}\mathcal{FL}_0^+(C,D) := C$
2. Otherwise, transform C and D into $\mathcal{FL}_0^+$-normal form and return

$$\mathsf{c\text{-}lcs}\text{-}\mathcal{FL}_0^+(C,D) := \bigsqcap_{A \in prim(C) \cap prim(D)} A \sqcap \bigsqcap_{A \in N_C} \forall\, (\widehat{L}_A(C) \cap \widehat{L}_A(D))\,.\,A$$

Figure 1: The lcs algorithm for $\mathcal{FL}_0^+$

2. $C_i = \forall w.A$: from $C_i \not\sqsubseteq D$ we can conclude that $\forall w.A \not\sqsubseteq D$, thus $w \in \widehat{L}_A(C)$, but $w \notin \widehat{L}_A(D)$, which is a contradiction to our assumption that Condition 2 holds.

■

The complexity of a subsumption test for two normalized $\mathcal{FL}_0^+$-concept descriptions of size n is polynomial, since there are at most $n+1$ subset tests to perform and each of these tests has a complexity in P.

3.3 Computing least common subsumer for $\mathcal{FL}_0^+$

For DLs providing transitive roles the usual approach for computing the lcs by unwinding the value restrictions and making a recursive call for the embedded concepts does not suffice. For example, if $t \in N_R^T$, then the $\mathsf{lcs}(\forall t.\forall t.\forall t.A, \forall t.A) \not\equiv \forall t.\mathsf{lcs}(\forall t.\forall t.A, A)$. Instead the $\mathsf{lcs}(\forall t.\forall t.\forall t.A, \forall t.A) \equiv \forall t.\forall t.\forall t.A$ requiring at least 3 value restrictions for t. So, in general it is necessary for the computation of the lcs to find the commonalities of the role languages that refer to the same concept name.

In Figure 1 we advise an algorithm for effectively computing the lcs of two $\mathcal{FL}_0^+$-concept descriptions. The algorithm checks first for some cases where the lcs is trivial. Then it transforms both concept descriptions in $\mathcal{FL}_0^+$-normal form and computes the intersection of the concept names appearing on the top-level of C and D. These are then conjoined with the value restrictions obtained from intersecting the transitive role languages of role-words referring to the same concept name.

Precisely, the result obtained by the algorithm from Figure 1 is not in every case a $\mathcal{FL}_0^+$-concept description, since it is represented by transitive role languages. However, these results can easily be converted into a $\mathcal{FL}_0^+$-concept description by performing the steps from Definition 3 in the inverse order. More precisely, replace for every $A \in N_C$ each transitive role r_i^+ in the set $\widehat{L}_A(\textsf{c-lcs-}\mathcal{FL}_0^+(C, D))$ with r_i and write it as a separate value restriction.

The size of the sets $prim(C)$ and $prim(D)$ is finite and the size of their intersection is also. The sets $L(C)$ and $L(D)$ are represented by a finite number of elements and their intersection can also be represented by a finite number of elements. Since there are only finitely many intersections to be computed during the computation of the lcs it is easy to see that the $\textsf{c-lcs-}\mathcal{FL}_0^+$-algorithm always terminates.

The next we prove that the concept obtained by $\textsf{c-lcs-}\mathcal{FL}_0^+$is the lcs of the two input $\mathcal{FL}_0^+$-concept descriptions.

Theorem 7 *Let C and D be two $\mathcal{FL}_0^+$-concept descriptions, then* $\textsf{c-lcs-}\mathcal{FL}_0^+(C, D) \equiv \textsf{lcs}(C, D)$.

Proof. We assume that $C \not\sqsubseteq D$ and $D \not\sqsubseteq C$ since then the $\textsf{lcs}(C, D)$ is trivial. Let $\textsf{c-lcs-}\mathcal{FL}_0^+(C, D) = E_{\textsf{lcs}}$. It is sufficient to show that

(i) $C \sqsubseteq E_{\textsf{lcs}}$ and $D \sqsubseteq E_{\textsf{lcs}}$, and

(ii) for all F with $C, D \sqsubseteq F$ it follows that $E_{\textsf{lcs}} \sqsubseteq F$.

Ad i) Obviously it is sufficient to show $C \sqsubseteq E_{\textsf{lcs}}$. Assume $E_{\textsf{lcs}} \not\equiv \top$. Then by definition of the algorithm $\textsf{c-lcs-}\mathcal{FL}_0^+$the Conditions 1 and 2 from Theorem 6

1. $prim(E_{\textsf{lcs}}) \subseteq prim(C)$.

2. for all $A \in N_C$: $\widehat{L}_A(E_{\textsf{lcs}}) \subseteq \widehat{L}_A(C)$.

are satisfied for C and $E_{\textsf{lcs}}$ and therefore $E_{\textsf{lcs}}$ subsumes C.

Ad ii) Let F be a $\mathcal{FL}_0^+$-concept description with $C, D \sqsubseteq F$. If $C \sqsubseteq D$ or $D \sqsubseteq C$, we get $E_{\textsf{lcs}} \sqsubseteq F$. Assume $C \not\sqsubseteq D$ and $D \not\sqsubseteq C$. If $F \equiv \top$ nothing has to be shown. Assume $F \not\equiv \top$. We show that $E_{\textsf{lcs}}$ and F satisfy all Conditions from Theorem 6

1. Condition 1: since $prim(F) \subseteq prim(C)$ and $prim(F) \subseteq prim(C)$, it follows $prim(F) \subseteq prim(C) \cap prim(D) = prim(E_{\mathsf{lcs}})$.
2. Condition 2: analogously. Since for all $A \in N_C$: $\widehat{L}_A(F) \subseteq \widehat{L}_A(C)$ and $\widehat{L}_A(F) \subseteq \widehat{L}_A(C)$, it follows $\widehat{L}_A(F) \subseteq \widehat{L}_A(C) \cap \widehat{L}_A(D) = \widehat{L}_A(E_{\mathsf{lcs}})$.

Consequently, $E_{\mathsf{lcs}} \sqsubseteq F$, which completes the proof. ∎

The complexity of the c-lcs-$\mathcal{FL}_0^+$algorithm is polynomial, since the number of subsumption tests made and the number of intersections computed during the second step are linear in n and both, computing subsumption and intersection, can be performed in polynomial time.

We have advised an algorithm to effectively compute the lcs of $\mathcal{FLE}^+$-concept descriptions by representing the value restrictions by role words. Thus it was possible to extend the approach for $\mathcal{FL}_0$ to transitive roles seamlessly.

4 Least common subsumer for $\mathcal{EL}^+$

The DL $\mathcal{EL}$ provides only conjunction, the concept $\top$, and existential restrictions. The structural characterization of subsumption as well as the computation of the lcs in $\mathcal{EL}$ have been thoroughly investigated in [4]. We extend the approach based on description trees presented there to $\mathcal{EL}^+$ and subsequently to $\mathcal{ELH}^+$.

In $\mathcal{EL}^+$ transitive roles may be used in existential restrictions. In $\mathcal{FL}_0^+$the value restrictions implied by transitivity affect all role successors "further down" in a role chain. In $\mathcal{EL}^+$ the exist restrictions implied by transitivity affect the role successors "further up" the role chain by the direct role relations induced by transitivity. The following example illustrates this effect: if $\exists t.\exists t.C$ holds for an individual a, transitivity implies that there is also a direct relation between a and the t-successor of the t-successor of a. Thus, $\exists t.C$ is also implied for a. To characterize subsumption for $\mathcal{EL}^+$ concept descriptions these implied role relations must be taken into account.

In addition to $prim(C)$ we need also an accessor for the existential restrictions used in concept descriptions: $\mathsf{ex}_r(C) := \{C' \mid \text{there exists } \exists r.C' \text{ on}$ the flattened top-level conjunction of $C\}$. W.l.o.g. we assume all $\mathcal{EL}$-, $\mathcal{EL}^+$-, and $\mathcal{ELH}^+$-concept descriptions to be in the following normal form:

$$C = \bigsqcap_{P \in P_C} P \quad \sqcap \quad \bigsqcap_{r \in N_R \cup N_R^T} \ \bigsqcap_{C' \in \mathsf{ex}_r(C)} \exists r.C'$$

where P_C is a subset of N_C. This normal form preserves equivalence. Observe that no existential restriction is imposed on a role r in case $\mathsf{ex}_r(C)$ is empty.

4.1 Representing $\mathcal{EL}^+$-concept descriptions

We extend the approach to structural subsumption in $\mathcal{EL}$ presented in [4] by using a different embedding mechanism for the description trees. We first define description trees as an alternative representation for concept descriptions. More precisely, we call this representation $\mathcal{EL}$-concept trees (and not $\mathcal{EL}^+$-concept trees) because it does not reflect the transitivity of roles $t \in N_R^T$ in any explicit way.

Definition 8 ($\mathcal{EL}$-description tree) *An* $\mathcal{EL}$-description tree *is a labeled tree $\mathcal{D} = (V, E, v_0, \ell)$, with*

- *root node $v_0 \in V$,*
- *$E \subseteq V \times (N_R \cup N_R^T) \times V$, and*
- *a labeling function ℓ that labels all $v \in V$ with $\ell(v) \subseteq N_C$ ($\top$ is the empty label).*

An edge $vrw \in E$ will be denoted as a $\exists$-edge in the following. For $v \in V$ the tree $\mathcal{D}(v)$ denotes the subtree of G with root node v.

Every $\mathcal{EL}^+$ concept description can be translated into am $\mathcal{EL}$- description tree. For the translation we need the notion of the *depth* (written as: $depth(C)$) of a concept description C, which is the maximal number of embedded quantors in the concept description. The *depth* (written as: $depth(\mathcal{D})$) of a description tree $\mathcal{D}$ is the length of its longest path.

The translation of a concept description into a description tree can be defined inductively:

- $depth(C) = 0$: Then C is of the form $\bigsqcap_{P \in prim(C)} P$. In this case, define $V := v_o$, $E := E^+ := \emptyset$, and $\ell(v_0) := prim(C)$.
- $depth(C) > 0$: For every $r \in N_R \cup N_R^T$ and for every $C' \in \mathsf{ex}_r(C)$, let $\mathcal{D}(C') = (V', E', v_0', \ell')$ be the inductively defined description trees for the existential restriction C' in C. W.l.o.g., assume that the sets of vertices V' are pairwise disjoint. Define $\mathcal{D}_C$ by
  - $V := \{v_o\} \cup \bigcup_{r \in N_R \cup N_R^T} \bigcup_{C' \in \mathsf{ex}_r(C)} V'$;

– $E := \{v_0 r v_0' \mid r \in N_R \cup N_R^T \wedge C' \in \mathsf{ex}_r(C) \wedge v_0' \text{ root of } \mathcal{D}(C')\}$
$\cup \bigcup_{r \in N_R \cup N_R^T} \bigcup_{C' \in \mathsf{ex}_r(C)} E'$;

– $\ell(v) := \begin{cases} prim(C) & \text{if } v = v_0 \\ \ell'(v) & \text{if } v \in V' \text{ for } r \in N_R \cup N_R^T,\ C' \in \mathsf{ex}_r(C) \end{cases}$

The inverse translation from a description tree into a concept description can also be defined inductively:

- $depth(\mathcal{D}) = 0$: Define $C_{\mathcal{D}} := \bigsqcap_{P \in \ell(v_0)} P$. Note that in case $\ell(v_0) = \emptyset$ the empty conjunction yields the top-concept $\top$.
- $depth(\mathcal{D}) > 0$: Denote by R_0 be the set of all roles in $(N_R \cup N_R^T)$ for which the node v_0 has a direct successor in E. For every $r \in R_0$, denote by V_r the set of r-successors of v_0 w.r.t. the role r. For every r and for every node $v_r \in V_r$, denote by C_{v_r} the concept description obtained inductively by translating the subtree of $\mathcal{D}$ induced by v_r. Define

$$C_{\mathcal{D}} := \bigsqcap_{P \in \ell(v_0)} P \quad \sqcap \quad \bigsqcap_{r \in R_0} \bigsqcap_{v_r \in V_r} C_{v_r}.$$

The semantics of a description tree is defined by the semantics of its corresponding concept description. The translation from a concept description into a description tree (and back) preserves equivalence in the sense that $C \equiv C_{\mathcal{D}(C)}$ and $\mathcal{D} \equiv \mathcal{D}_{C_{\mathcal{D}}}$.

Example 9 *Let $C_1 = \exists r.((\exists t.\exists t.\exists t.A) \sqcap (\exists s.(B \sqcap \exists s.C))$ and $C_2 = \exists r.((\exists t.A) \sqcap (\exists s.B)))$ be two $\mathcal{EL}^+$-concept descriptions, where $t \in N_R^T$ and $r, s \in N_R$. The corresponding description trees are depicted in Figure 2. Every node v is shown along with its respective label $\ell(v)$.*

4.2 Characterization of subsumption for $\mathcal{EL}^+$

Equipped with description trees we characterize subsumption by a homomorphism from the $\mathcal{EL}$-description tree of the subsumer into the one of the subsumee. For each r-edge with $r \in N_R$ in the description tree of the subsumer at least one corresponding r-edge must exist in the description tree of the subsumee. If r is a transitive role, i.e., $r \in N_R^T$, then an r-edge in the description tree of the subsumer can also be associated with an r-path in the description tree of the subsumee—in the sense that the origin of the r-edge

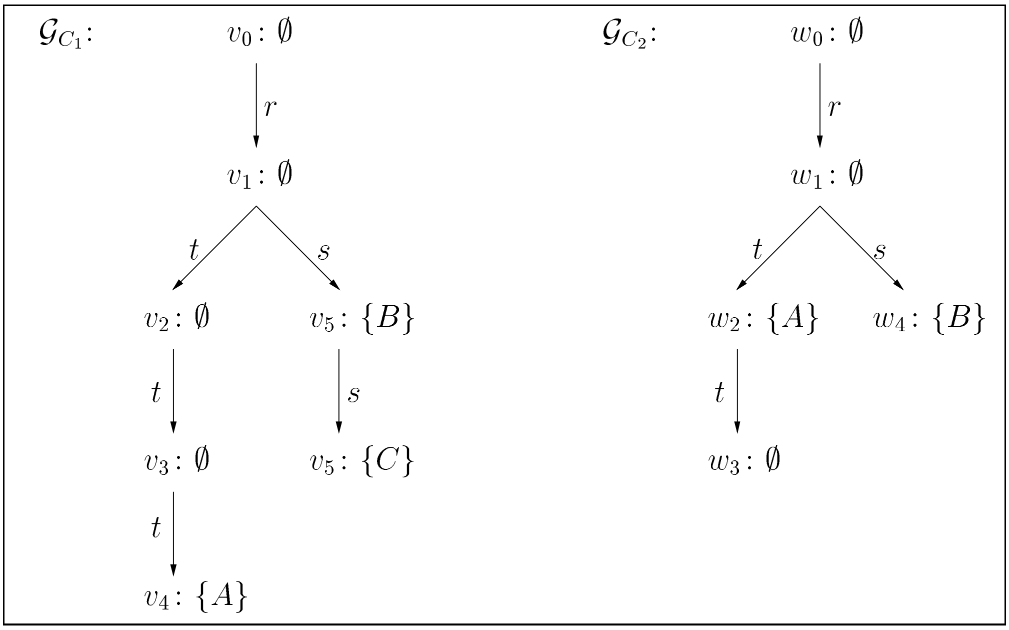

Figure 2: $\mathcal{EL}$-description trees

is mapped onto the first node of the r-path and the end point of the r-edge onto the last node of the path.

Definition 10 ($\mathcal{EL}^+$-Homomorphism) *Let $\mathcal{D} = (V_\mathcal{D}, E_\mathcal{D}, v_0, \ell_\mathcal{D})$ and $\mathcal{H} = (V_\mathcal{H}, E_\mathcal{H}, w_0, \ell_\mathcal{H})$ be $\mathcal{EL}$-description trees. A mapping $\varphi\colon V_\mathcal{H} \longrightarrow V_\mathcal{G}$ is an* $\mathcal{EL}^+$-Homomorphism *iff all of the following conditions hold:*

- $\varphi(w_0) = v_0$,
- *for all $w \in V_\mathcal{H}$: $\ell_\mathcal{H}(w) \subseteq \ell_\mathcal{G}(\varphi(w))$, and*
- *for all $vrw \in E_\mathcal{H}$:* $\begin{cases} \varphi(v) r \varphi(w) \in E_\mathcal{G} & \text{if } r \in N_R \\ \varphi(v) r^+ \varphi(w) \in E_\mathcal{G} & \text{if } r \in N_R^T \end{cases}$.

The following example illustrates the notion of $\mathcal{EL}^+$-homomorphisms.

Example 11 *Let C_1 and C_2 be defined as in Example 9. The homomorphism from $\mathcal{D}_{C_1}$ to $\mathcal{D}_{C_2}$ maps w_0 to v_0, w_1 to v_1, w_2 to v_2, and w_3 to v_6. The node w_3 can be mapped to v_6 since w_1 is connected to w_3 by a t-edge and v_1 is connected to v_6 by a path consisting only of t-edges. Observe, that transitivity of t implies a direct t-edge from v_1 to v_6. Therefore v_6 is also a* direct *role-successor of v_0 and w_3 can be mapped to v_6.*

The characterization of subsumption for $\mathcal{EL}^+$-concept descriptions is now given by the following theorem.

Theorem 12 *Let C and D be $\mathcal{EL}^+$-concept descriptions and let $\mathcal{D}_C$ and $\mathcal{D}_D$ be their corresponding $\mathcal{EL}$-description trees. Then $C \sqsubseteq D$ iff there exists an $\mathcal{EL}^+$-homomorphism φ from $\mathcal{D}_D$ to $\mathcal{D}_C$.*

PROOF. Let $\mathcal{D}_C = (V_C, E_C, v_0, \ell_C)$ and $\mathcal{D}_D = (V_D, E_D, w_0, \ell_D)$ be the corresponding $\mathcal{EL}$-description trees for C and D.

1.) "$\longrightarrow$": $C \sqsubseteq D$
Assume that $D \sqsubset \top$, otherwise the claim trivially holds. We prove the claim by showing that there always exists a mapping function φ between the $\mathcal{EL}$-description trees of D and C that fulfills all conditions from Definition 10. Condition 1 from Definition 10 trivially holds since the root nodes can be mapped to each other; $\varphi(w_0) := v_0$.

We show now that the Conditions 2 and 3 from Definition 10 hold for the mapping φ by induction on $depth(D)$.
Base case: $depth(D) = 0$
implies that $D = P_1 \sqcap \cdots \sqcap P_n$ for $n > 0$ and $P_i \in N_C$. Thus, $\ell_D(w_0) = \{P_1, \ldots, P_n\}$. Since $C \subseteq D$, we have $C^{\mathcal{I}} \subseteq (P_1 \sqcap \cdots \sqcap P_n)^{\mathcal{I}}$ this implies $\{P_1, \ldots, P_n\} \subseteq \ell_C(v_0)$ and since $v_0 = \varphi(w_0)$ we obtain $\ell_D(w_0) \subseteq \ell_C(\varphi(w_0))$.

Since $depth(D) = 0$ implies $E_C = \emptyset$ there is nothing to show for Condition 3 from Definition 10.

Induction step: $depth(D) > 0$
We first show that Condition 2 and 3 from Definition 10 hold for the first role-level and use the induction hypothesis for the subsequent role-levels. $depth(D) > 0$ implies $D = P_1 \sqcap \cdots \sqcap P_n \ \sqcap \ (\sqcap_{r \in N_R \cup N_R^T} \ \sqcap_{E \in \mathsf{ex}_r(D)} \exists r.E)$ for $n > 0$ and all E being arbitrary $\mathcal{EL}^+$-concept descriptions.

Again, for the root node w_0 holds that $\ell_D(w_0) = \{P_1, \ldots, P_n\}$. Since $C \subseteq D$, we have $C^{\mathcal{I}} \subseteq (P_1 \sqcap \cdots \sqcap P_n)^{\mathcal{I}}$ this implies $\{P_1, \ldots, P_n\} \subseteq \ell_C(v_0)$ and since $v_0 = \varphi(w_0)$ we obtain $\ell_D(w_0) \subseteq \ell_C(\varphi(w_0))$. Thus Condition 2 for a homomorphism holds for φ and the root node.

By definition of $\mathcal{D}_D$ for all existential restrictions $\{\exists r.E \mid r \in N_R \cup N_R^T, E \in \mathsf{ex}_r(D)\}$ in the concept D there must exists an edge $w_0 r w_r \in E_D$. Since D is satisfiable, there exists a interpretation $\mathcal{I}$ of D and a canonical model of D, where for every existential restriction $\exists r.E$ used on the top role-level of D there exists an individual a' s.t. $(a, a') \in r^{\mathcal{I}}$ and $a' \in E^{\mathcal{I}}$. We have to make a case distinction for $r \in N_R$ or $r \in N_R^T$.

- $r \in N_R$
  If $r \in N_R$, then $C \sqsubseteq D$ implies that there is an existential restriction $\exists r.C'$ on the top role-level of C, s.t. $C' \sqsubseteq E$. By definition of $\mathcal{D}_C$, there must be an r-edge from v_0 in $\mathcal{D}_C$ to another node, say v_r. Thus we can map w_r to v_r by φ and (since $v_0 = \varphi(w_0)$) we have $v_0 r v_r = \varphi(w_0) r \varphi(w_r)$ and the Condition 3 from Definition 10 holds for φ and all direct role-successors of non-transitive roles on top role-level of D.
- $r \in N_R^T$
  Since $C \sqsubseteq D$ and thus there exists in all interpretations of C and a canonical model of C with an individual b which has an r-successor b', with $(b, b') \in r^{\mathcal{I}}$ and $b' \in C'^{\mathcal{I}}$, s.t. $C' \sqsubseteq E$. If $r \in N_R^T$, then $(b, b') \in \bigcup_{1 \leq n} (r^{\mathcal{I}})^n$ and thus there has to exist a r-path from b to b' with length k $(1 \leq k)$ in the canonical model of C. Thus there have to exist k nested existential restrictions in C for the role r. From that follows by the definition of $\mathcal{D}_C$ that there exists a r-path of length k starting from v_0 to another node , say v_r. Thus we can map w_r to v_r by φ and (since $v_0 = \varphi(w_0)$) we have $v_0 r v_r = \varphi(w_0) r \varphi(w_r)$ and the Condition 3 from Definition 10 holds for φ and all direct role-successors of transitive roles on top role-level of D.

For every $\exists r.E$ in D there exists a node $w_r \in V_D$ of $\mathcal{D}_D$ s.t. $w_0 r w_r \in E_D$ and for every $\exists r.C'$ in C there exists a node $v_r \in V_C$ of $\mathcal{D}_C$ s.t. $v_0 r v_r \in E_C^+$. Since $C \sqsubseteq D$ implies that $C' \sqsubseteq E$ and $v_0 r v_r = \varphi(w_0) r \varphi(w_r)$ for every existential restriction in D we can conclude that there exists a homomorphism φ_r between $\mathcal{D}_D(w_r)$ and $\mathcal{D}_C(v_r)$ by induction hypothesis. So, using the mappings from the different φ_rs in φ, we obtain a homomorphism from $\mathcal{D}_D$ to $\mathcal{D}_C$.

2.) "$\longleftarrow$": a homomorphism φ from $\mathcal{D}_D$ to $\mathcal{D}_C$ exists.
We prove the claim by induction on $depth(D)$
Base case: $depth(D) = 0$
implies that $E_D = \emptyset$ and $D \equiv D_{\mathcal{D}_D} = \sqcap_{P_i \in \ell_D(w_0)} P_i$. Since a homomorphism φ exists, we have $\varphi(w_0) = v_0$ and thus $\ell_D(w_0) \subseteq \ell_C(\varphi(w_0))$. From this and the definition of $C_{\mathcal{D}_C}$ we can conclude that $C_{\mathcal{D}_C} \sqsubseteq (\sqcap_{P_i \in \ell_D(w_0)} P_i)$ and since $C \equiv C_{\mathcal{D}_C} \sqsubseteq (\sqcap_{P_i \in \ell_D(w_0)} P_i) = D_{\mathcal{D}_D} \equiv D$ we have $C \sqsubseteq D$.
Induction step: $depth(D) > 0$
By the definition of the translation from description tree to concept descriptions we know that:

$$D \equiv D_{\mathcal{D}_D} = (\sqcap_{P_i \in \ell_D(w_0)} P_i) \ \sqcap \ (\sqcap_{(w_0 r_j w_i) \in E_D} \exists r_j.D'), \text{where } D' := D_{\mathcal{D}_D(w_i)}.$$

We have to show that (1) for all $P_i \in \ell_D(w_0)$ holds $C \sqsubseteq P_i$ and that (2) for all $\exists r.D_i \in \{\exists r_j.D' \mid (r_j \in N_R \cup N_R^T) \wedge (w_0 r_j w_i) \in E_D \wedge D' := D_{\mathcal{D}_D(w_i)}\}$ is holds that: $C \sqsubseteq \exists r.D_i$

The proof for (1) is analogous to the base case: Since a homomorphism φ exists, we have $\varphi(w_0) = v_0$ and thus $\ell_D(w_0) \subseteq \ell_C(\varphi(w_0))$. From this and the definition of $C_{\mathcal{D}_C}$ we can conclude that $C_{\mathcal{D}_C} \sqsubseteq (\sqcap_{P_i \in \ell_D(w_0)} P_i)$ and thus $C \equiv C_{\mathcal{D}_C} \sqsubseteq (\sqcap_{P_i \in \ell_D(w_0)} P_i)$.

For the proof of (2) we use that by definition of $\mathcal{D}_D$ we have for every $\exists r.D_i$ on top-role level of D a node w_i s.t. $(w_0 r w_i) \in E_D$ and $D_{\mathcal{D}_D(w_i)} \equiv D_i$. Since a homomorphism φ exists from $\mathcal{D}_D$ to $\mathcal{D}_C$, it holds that $(\varphi(w_0) r \varphi(w_i)) \in E_C$, if $r \in N_R \cup N_R^T$ for every $(w_0 r w_i) \in E_D$. W.l.o.g. we assume that $\varphi(w_i) = v_i$ for some $v_i \in V_C$ and thus have a r-path (possibly of length 1) from v_0 to v_i in $\mathcal{D}_C$. Since there exists a homomorphism φ from $\mathcal{D}_D$ to $\mathcal{D}_C$ and $\varphi(w_i) = v_i$, it follows from the definition of a homomorphism that there exists a homomorphism φ' from $\mathcal{D}_D(w_i)$ to $\mathcal{D}_C(v_i)$ for every r-successor for all r. Applying the induction hypothesis we obtain that $C_{\mathcal{D}_C(v_i)} \sqsubseteq D_{\mathcal{D}_D(w_i)} \equiv D_i$ and thus $C \sqsubseteq \exists r.C_{\mathcal{D}_C(v_i)} \sqsubseteq \exists r.D_{\mathcal{D}_D(w_i)} \equiv \exists r.D_i$ for all $\exists r.D_i \in \{\exists r_j.D' \mid (r_j \in N_R \cup N_R^T) \wedge (w_0 r_j w_i) \in E_D \wedge D' := D_{\mathcal{D}_D(w_i)}\}$. From that and the proof of (1), where we concluded $C \sqsubseteq (\sqcap_{P_i \in \ell_D(w_0)} P_i)$ directly follows: $C \sqsubseteq D$, which completes the proof of the theorem. ∎

$\mathcal{EL}^+$-concept descriptions can be translated into $\mathcal{EL}$-description trees in polynomial time since only one traversal of the concept description is needed. In [3] the authors devise a polynomial-time algorithm to decide the existence of a homomorphism between two given $\mathcal{EL}$-description trees. In Figure 3, this algorithm is extended to $\mathcal{EL}^+$ by testing the existence of an r-path between two nodes in case r is a transitive role (see line($**$)). The general idea is to define a mapping $\delta\colon V_\mathcal{G} \longrightarrow \mathcal{P}(V_\mathcal{H})$ that labels every node $v \in V_\mathcal{G}$ with a set of nodes from $V_\mathcal{H}$ by once traversing the description tree $\mathcal{D}_\mathcal{H}$ from this leaves to its root w_0. If $w_0 \in \delta(v_0)$, then the mapping δ induces a homomorphism from $\mathcal{D}_\mathcal{H}$ to $\mathcal{D}_\mathcal{G}$.

We can now test subsumption between two $\mathcal{EL}^+$-concept descriptions C and D with the following decision procedure:

1. Translate C and D into the corresponding description trees $\mathcal{D}_C$ and $\mathcal{D}_D$.
2. Decide whether there exists a homomorphism from $\mathcal{D}_C$ to $\mathcal{D}_D$. In case such a homomorphism exists return “true”, otherwise return “false”.

Input: $\mathcal{EL}$-Description trees $\mathcal{D}_\mathcal{H} = (V_\mathcal{H}, E_\mathcal{H}, w_0, \ell_\mathcal{H})$,
$\mathcal{D}_\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G}, v_0, \ell_\mathcal{G})$.
Output: "True", if a homomorphism from $\mathcal{D}_\mathcal{H}$ to $\mathcal{D}_\mathcal{G}$ exists;
"False" otherwise.

Algorithm: $\mathsf{hom}(\mathcal{D}_\mathcal{H}, \mathcal{D}_\mathcal{G})$
Let $\{v_1, \ldots, v_n\}$ be $V_\mathcal{H}$ sorted in post-order.
Define a mapping $\delta : V_\mathcal{G} \longrightarrow \mathcal{P}(V_\mathcal{H})$ as follows:
Initialize δ with $\delta(v) := \emptyset$ for all $v \in V_\mathcal{G}$;
For $1 \leq i \leq n$
For $w \in V_\mathcal{G}$
If $\ell_\mathcal{H}(v_i) \subseteq \ell_\mathcal{G}(w) \wedge$
$\forall v_i r v \in E_\mathcal{H} : \exists w' \in V_\mathcal{G} :$
$\big(v \in \delta(w') \wedge$
$\big((r \in N_R \wedge w r w' \in E_\mathcal{G}) \vee (r \in N_R^T \wedge w r^+ w' \in E_\mathcal{G})\big)\big)$ (**)
Then $\delta(w) := \delta(w) \cup \{v_i\}$;

If $w_0 \in \delta(v_0)$, then return "True", else return "False".

Figure 3: Algorithm for deciding existence of an $\mathcal{EL}^+$-homomorphism between two $\mathcal{EL}$-description trees.

Proposition 13 *For $\mathcal{EL}^+$-concept descriptions the subsumption problem $C \sqsubseteq D$ can be decided in polynomial time.*

4.3 Computing least common subsumer for $\mathcal{EL}^+$

The subsumption test for $\mathcal{EL}$ could be extended to $\mathcal{EL}^+$ without significant changes to the definition of a description tree. It is therefore natural to try to extend the existing algorithm for the lcs-computation in $\mathcal{EL}$ to $\mathcal{EL}^+$ in a similar way. In [4], the lcs of $\mathcal{EL}$-concept descriptions is obtained from the tree-product of the respective description trees. For $\mathcal{EL}^+$, however, we first need to extend the notion of a description tree so as to make explicit the effect of transitive roles. To this end, $\mathcal{EL}^+$-description trees are now introduced.

Definition 14 ($\mathcal{EL}^+$-description tree) *Let (V, E, v_0, ℓ) be an $\mathcal{EL}$-description tree over N_C, N_R, and N_R^T. Let E^+ be a set of edges such that $urv \in E^+$ iff $r \in N_R^T$ and there exists an r-path from u to v in E whose length is at least*

2. Then the structure (V, E, E^+, v_0, ℓ) *is called an* $\mathcal{EL}^+$-description tree*. An edge in* E^+ *is called* forward edge.

Hence, in $\mathcal{EL}^+$-description trees additional forward edges reflect the transitivity of a roles. The translation function from $\mathcal{EL}^+$-concept descriptions to $\mathcal{EL}$-description trees can be extended to $\mathcal{EL}^+$-description trees with little effort:

- If $depth(C) = 0$ then the set of forward edges E^+ is empty.
- In case $depth(C) > 0$ the set E^+ is computed inductively in the following way:

$$E^+ := \{v_0 r v' \mid r \in R_C^+ \wedge \exists r\text{-path from } v_0 \text{ to } v' \text{ in } E\} \cup \bigcup_{r \in N_R \cup N_R^T} \bigcup_{C' \in \mathsf{ex}_r(C)} E'^+$$

  where E'^+ denotes the set of forward edges in the subtree induced by the existential restriction $C' \in \mathsf{ex}_r(C)$.

Example 15 *Consider the concept descriptions* C_1 *and* C_2 *from the previous example. The corresponding* $\mathcal{EL}^+$*-description trees of* C_1 *and* C_2 *are shown in Figure 4. Forward edges are depicted as dotted edges. Since* $\mathcal{G}_{C_2}$ *has no t-path longer than 1, no forward edges are added.*

It is easy to see that the size of an $\mathcal{EL}^+$-description tree is polynomial in the size of the original $\mathcal{EL}^+$-concept description. The usage of forward edges bypassing transitive role paths can also be seen as a means of structure sharing in an otherwise exponentially larger ordinary description tree.

The inverse translation from an description tree into a concept description can also be adapted easily from the translation procedure for ordinary $\mathcal{EL}^+$-description trees. For a given $\mathcal{EL}^+$-description tree, nothing changes in case $depth(\mathcal{D}) = 0$. If $depth(\mathcal{D}) > 0$ then the union $E \cup E^+$ is used instead of E for the inductive construction of $C_\mathcal{D}$. Again, we find that the translations for $\mathcal{EL}^+$-description trees also preserve equivalence in the sense that $C \equiv C_{\mathcal{D}(C)}$ and $\mathcal{D} \equiv \mathcal{D}_{C_\mathcal{D}}$. It should however be noted that the concept description $C_{\mathcal{D}(C)}$ is not necessarily equal to C anymore, as the following example shows.

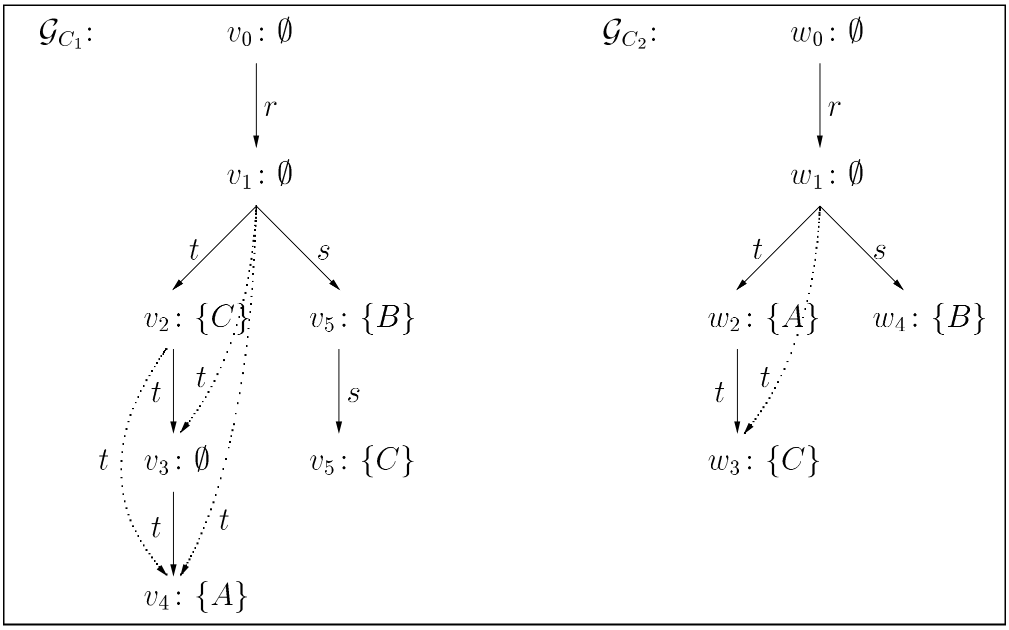

Figure 4: $\mathcal{EL}^+$-Description trees

Example 16 *Consider the $\mathcal{EL}^+$-description tree $\mathcal{G}_{C_1}$ from Figure 4. The original $\mathcal{EL}^+$-concept description was $C_1 = \exists r.((\exists t.\exists t.\exists t.A) \sqcap (\exists s.(B \sqcap \exists s.C))$. As the backward translation additionally takes into account forward edges, we obtain*

$$C_{\mathcal{D}(C_1)} = \exists r.\big(\exists t.((\exists t.\exists t.A) \sqcap \exists t.A) \sqcap \exists t.A \sqcap \exists t.\exists t.A \sqcap (\exists s.(B \sqcap \exists s.C))\big)$$

which is equivalent but obviously not equal to C_1.

The lcs of two normalized $\mathcal{EL}^+$-concept descriptions can be obtained by computing the product of their corresponding description trees, with a product operation defined inductively as follows:

Definition 17 (Product of $\mathcal{EL}^+$-description trees) *Let $\mathcal{G} := (V_\mathcal{G}, E_\mathcal{G}, E^+_\mathcal{G}, v_0, \ell_\mathcal{G})$ and $\mathcal{H} := (V_\mathcal{H}, E_\mathcal{H}, E^+_\mathcal{H}, w_0, \ell_\mathcal{H})$ be two $\mathcal{EL}^+$-description trees. The* product tree *$\mathcal{G} \times \mathcal{H}$ is inductively defined as follows.*

- *The root node is (v_0, w_0).*
- *The set of vertices $V_{\mathcal{G}\times\mathcal{H}}$ is a subset of $V_\mathcal{G} \times V_\mathcal{H}$ containing the root node and the sets of vertices inductively generated for the successors of v_0*

and w_0*:*

$$V_{\mathcal{G}\times\mathcal{H}} := \{(v_0, w_0)\} \cup \bigcup_{r\in N_R \cup N_R^T} \bigcup_{v_0rv\in E_\mathcal{G}} \bigcup_{w_0rw\in E_\mathcal{H}\cup E^+_\mathcal{H}} V_{\mathcal{G}(v)\times\mathcal{H}(w)} \cup \bigcup_{r\in N_R \cup N_R^T} \bigcup_{v_0rv\in E^+_\mathcal{G}} \bigcup_{w_0rw\in E_\mathcal{H}} V_{\mathcal{G}(v)\times\mathcal{H}(w)},$$

where $V_{\mathcal{G}(v)\times\mathcal{H}(w)}$ *denotes the vertex set of the tree product* $\mathcal{G}(v)\times\mathcal{H}(w)$ *of the subtrees induced by the nodes* v *and* w*.*

- *In the product tree, the root* (v_0, w_0) *node is connected to a node* (v, w) *by an* r*-edge whenever the individual* r*-edges* v_0rv *and* w_0rw *exist in* $\mathcal{G}$ *and* $\mathcal{H}$ *respectively. The only exception is that not both edges may be forward edges. The rest of* $E_{\mathcal{G}\times\mathcal{H}}$ *is obtained inductively:*

$$E_{\mathcal{G}\times\mathcal{H}} := \{(v_0, w_0)r(v, w) \mid v_0rv \in E_\mathcal{G} \wedge w_0rw \in E_\mathcal{H} \cup E^+_\mathcal{H}\} \cup \{(v_0, w_0)r(v, w) \mid v_0rv \in E^+_\mathcal{G} \wedge w_0rw \in E_\mathcal{H}\} \cup \bigcup_{r\in N_R \cup N_R^T} \bigcup_{v_0rv\in E_\mathcal{G}} \bigcup_{w_0rw\in E_\mathcal{H}\cup E^+_\mathcal{H}} E_{\mathcal{G}(v)\times\mathcal{H}(w)} \cup \bigcup_{r\in N_R \cup N_R^T} \bigcup_{v_0rv\in E^+_\mathcal{G}} \bigcup_{w_0rw\in E_\mathcal{H}} E_{\mathcal{G}(v)\times\mathcal{H}(w)}$$

where $E_{\mathcal{G}(v)\times\mathcal{H}(w)}$ *denotes the set of edges of the tree product* $\mathcal{G}(v)\times\mathcal{H}(w)$ *of the subtrees induced by the nodes* v *and* w*.*

The product of two $\mathcal{EL}^+$-description trees is an ordinary $\mathcal{EL}$-description tree, i.e., does not contain forward edges. The following example takes up the description trees shown previously to show the effect of the product operation.

Example 18 *Consider the* $\mathcal{EL}^+$*-description trees* $\mathcal{G}_{C_1}$ *and* $\mathcal{G}_{C_2}$ *from Example 15. By definition, the root node of the product tree* $\mathcal{G}_{C_1}\times\mathcal{G}_{C_2}$ *is* (v_0, w_0)*. Now we have to consider all pairs of successors of* v_0 *and* w_0 *that agree on the edge label—excluding those pairs where both successors are reached via forward edges. In case of the root nodes* v_0 *and* w_0*, only the pair* $(v1, w1)$ *of* r*-successors is found. Hence, the root node of the product tree has* $(v1, w1)$ *as the only successor. The rest of the product tree is computed inductively as the product of the subtrees* $\mathcal{G}_{C_1}(v_1)$ *and* $\mathcal{G}_{C_2}(w_1)$*. The node* v_1 *has 3 direct*

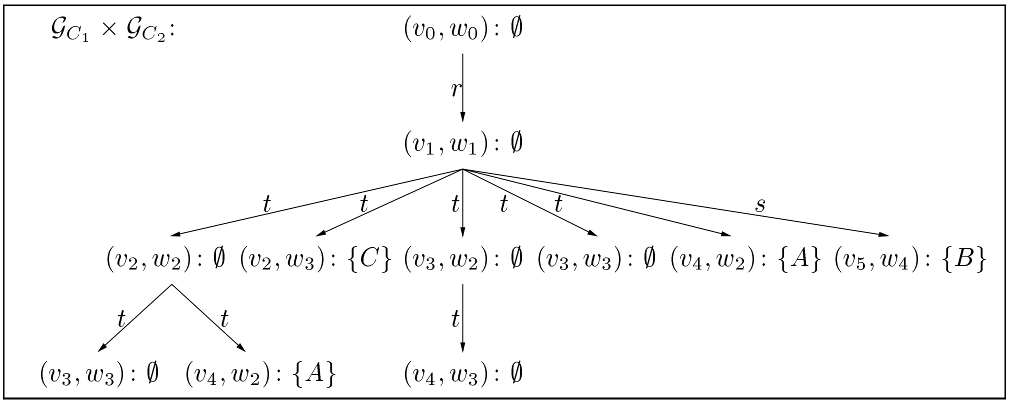

Figure 5: $\mathcal{EL}^+$-product tree

t-successors, namely v_2 and (w.r.t. forward edges) v_3 and v_4. The node w_1 has 2 direct t-successors, namely w_2 and, via a forward edge, w_3. By definition, the node (v_1, w_1) in the product tree has therefore 5 direct t-successors, namely (v_2, w_2), (v_2, w_3), (v_3, w_2), (v_3, w_3), and (v_4, w_2). Note that the pairs (v_3, w_3) and $v_4, w_3)$ are omitted because the definition forbids that both nodes in a pair are reached via forward edges. As both v_1 and w_1 have exactly one r-successor, the node (v_1, w_1) furthermore has (v_5, w_4) as a direct r-successor. The label set of every node $\ell(v_i, w_j)$ is the intersection of the label sets $\ell(v_i)$ and $\ell(v_j)$. The final result of the product tree computation is presented in Figure 5.

We still have to show that the product tree of two description trees, computed in the way described above, in fact produces the description tree of the lcs.

Theorem 19 *Let C and D be two $\mathcal{EL}^+$-concept descriptions and let $\mathcal{D}_C$ and $\mathcal{D}_D$ be their corresponding $\mathcal{EL}^+$-description trees. Then $C_{\mathcal{D}_C \times \mathcal{D}_D}$ is the lcs of C and D.*

PROOF. Let $\mathcal{D}_C \times \mathcal{D}_D = (V_{\mathcal{D}_C \times \mathcal{D}_D}, E_{\mathcal{D}_C \times \mathcal{D}_D}, (v_0, w_0), \ell_{\mathcal{D}_C \times \mathcal{D}_D})$. We have to show that $C_{\mathcal{D}_C \times \mathcal{D}_D}$ meets the two conditions:

1. $C \sqsubseteq C_{\mathcal{D}_C \times \mathcal{D}_D}$ and $D \sqsubseteq C_{\mathcal{D}_C \times \mathcal{D}_D}$, and
2. if E satisfies $C \sqsubseteq E$ and $D \sqsubseteq E$, then $C_{\mathcal{D}_C \times \mathcal{D}_D} \sqsubseteq E$.

We show 1.) by constructing a homomorphism φ from $C_{\mathcal{D}_C\times\mathcal{D}_D}$ to $\mathcal{D}_C$. The projection π_i with $i \in \{1,2\}$, yields a homomorphism from $C_{\mathcal{D}_C\times\mathcal{D}_D}$ to $\mathcal{D}_C$ for $i=1$ and to $\mathcal{D}_D$ for $i=2$. By Theorem 12 this implies $C \sqsubseteq C_{\mathcal{D}_C\times\mathcal{D}_D}$ and $D \sqsubseteq C_{\mathcal{D}_C\times\mathcal{D}_D}$.

To show 2.) assume that E is an arbitrary subsumer of C and D, and let $\mathcal{D}_E = (V', E', v_0', \ell')$ be the corresponding description tree. Theorem 12 yields then a homomorphism φ_1 from $\mathcal{D}_E$ to $\mathcal{D}_C$ and φ_2 from $\mathcal{D}_E$ to $\mathcal{D}_D$. Define a mapping $\varphi := <\varphi_1, \varphi_2>$ from $\mathcal{D}_E$ to $\mathcal{D}_C \times \mathcal{D}_D$ 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 (a) φ is well-defined, i.e., $\varphi(v') \in V_{\mathcal{D}_C\times\mathcal{D}_D}$ for all $v' \in V'$, and that (b) φ is a homomorphism from $\mathcal{D}_E$ to $\mathcal{D}_C \times \mathcal{D}_D$ according to Definition 10.

Claim (a) is shown by induction on the length of the path $\delta(v')$ in $\mathcal{D}_E$ from v_0' to v'.

- $\delta(v') = 0$.
  Then we have $v' = v_0'$ and hence, $\varphi(v_0') = (\varphi_1(v_0'), \varphi_2(v_0')) = (v_0, w_0) \in V_{\mathcal{D}_C\times\mathcal{D}_D}$.
- $\delta(v') > 0$.
  Since $\mathcal{D}_E$ is a tree, there exists a unique predecessor $v'' \in V'$ of v', i.e., $v''rv' \in E'$ for some $r \in N_R \cup N_R^T$. Assume $v''rv' \in E'$ for some $r \in N_R \cup N_R^T$. Obviously, we have $\delta(v'') = \delta(v') - 1$. By induction, we know $(\varphi_1(v''), \varphi_2(v'')) \in V_{\mathcal{D}_C\times\mathcal{D}_D}$. Since φ_1 and φ_2 are homomorphisms and since C and D are in $\mathcal{EL}^+$-normal form, we have direct r-successors $\varphi_1(v'')r\varphi_1(v') \in E_C$ and $\varphi_2(v'')r\varphi_2(v') \in E_D$ (even if $r \in N_R^T$). Definition 17 yields $(\varphi_1(v'), \varphi_2(v'))$ as an r-successor of $(\varphi_1(v''), \varphi_2(v''))$ in $\mathcal{D}_C \times \mathcal{D}_D$ and hence, $(\varphi_1(v'), \varphi_2(v')) \in V_{\mathcal{D}_C\times\mathcal{D}_D}$.

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 $\mathcal{D}_E$ to $\mathcal{D}_C$ ($\mathcal{D}_D$).
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 $\ell'(v') \subseteq \ell_C(\varphi_1(v')) \cap \ell_D(\varphi_2(v')) = \ell_{\mathcal{D}_C\times\mathcal{D}_D}(\varphi_1(v'), \varphi_2(v'))$.
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 17, it is $(\varphi_1(v'), \varphi_2(v'))r(\varphi_1(w'), \varphi_2(w')) \in E_{\mathcal{D}_C\times\mathcal{D}_D}$.

Now Theorem 12 implies $C_{\mathcal{D}_C\times\mathcal{D}_D} \sqsubseteq E$ which completes the proof. ∎

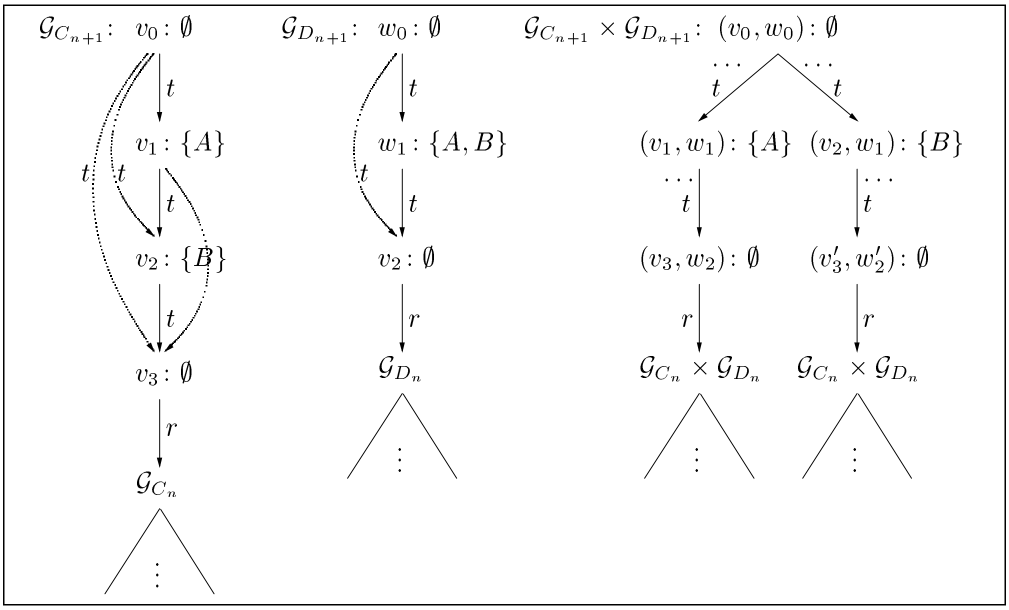

Figure 6: $\mathcal{EL}^+$-least common subsumer

As a consequence of the above result the following procedure is sufficient to compute the lcs of two given $\mathcal{EL}^+$-concept descriptions C and D:

1. Translate C and D into their corresponding $\mathcal{EL}^+$-description trees $\mathcal{D}_C$ and $\mathcal{D}_D$.
2. Compute the product of the description trees $\mathcal{D}_C \times \mathcal{D}_D$.
3. Translate $\mathcal{D}_C \times \mathcal{D}_D$ back into the concept description $C_{\mathcal{D}_C \times \mathcal{D}_D}$.

The size of the lcs can be exponential in the size of the original $\mathcal{EL}^+$-concept descriptions. The following example briefly presents such a case.

Example 20 *Let $N_R := \{r\}$ and $N_R^T := \{t\}$. For some $n \in \mathbb{N}$, let C_n and D_n be two $\mathcal{EL}^+$-concept descriptions inductively defined as seen below:*

$$C_0 := \top \qquad D_0 := \top$$
$$C_{n+1} := \exists t.\exists t.\exists t.\exists r.C_n \qquad D_{n+1} := \exists t.\exists t.\exists r.D_n$$

The relevant $\mathcal{EL}^+$-description trees are shown in Figure 6. For $n > 0$, the description tree of C_{n+1} does not end at the node denoted $\mathcal{G}_{C_n}$, but proceed just as it begins at v_0. The same holds for $\mathcal{G}_{D_n}$. The third graph in Figure 6 shows *that part of the product tree* $\mathcal{G}_{C_{n+1} \times D_{n+1}}$ *in which the exponential blow-up can be seen easily.*

Since v_1 *and* w_1 *are both t-successors of their respective root-nodes the root of the product tree has* (v_1, w_1) *as one t-successor. Its label set is* $\{A\}$*, the intersection of* $\ell(v_1)$ *and* $\ell(w_1)$*. From* v_1*, the node* G_{C_n} *is reached via one t-forward edge (reaching* v_3*) and one r-edge. Similarly, from* w_1 *we arrive at node* G_{C_n} *via one t-edge (reaching* w_2*) and one r-edge. In the product tree the node* (v_1, w_1) *therefore has a t-successor (namely* (v_3, w_2)*) with an r-successor for which the subtree* $\mathcal{G}_{C_n} \times \mathcal{G}_{D_n}$ *must be computed.*

A similar result is obtained for v_2 *and* w_1*: the root node of the product tree has node* $(v2, w_1)$ *as t-successor (with label set* $\{B\}$*) and from there we similarly arrive at a node for which* $\mathcal{G}_{C_n} \times \mathcal{G}_{D_n}$ *must be computed (see Figure 6). Because of the different labels in* (v_1, w_1) *and* (v_2, w_1) *none of these paths is redundant.*

The computation of $\mathcal{G}_{C_n} \times \mathcal{G}_{D_n}$ *produces the same branch as seen at the root node* (v_0, w_0)*, so that finally a description tree with exponentially many leaves (in n) emerges. Hence, an exponentially large concept description (in n) is returned as lcs of* C_{n+1} *and* D_{n+1}*.*

The previous example has shown that cases exist where the lcs of two $\mathcal{EL}^+$-concept descriptions is exponentially large. On the other hand it is not difficult to see that the computation of the lcs takes at most exponential time in the size of the input concept descriptions. In $\mathcal{EL}$, the lcs of two concept descriptions is polynomial in the size of the input concepts and can be computed in polynomial time. The extension of transitive roles therefore increases the computational complexity both in space and time.

5 Least common subsumer for $\mathcal{FLE}^+$

The lcs has already been investigated for sub-logics of $\mathcal{FLE}^+$. The work of Baader, Küsters, and Molitor [4, 3] investigates the computation of the lcs in $\mathcal{FLE}$ and its sublanguages. In [1], the lcs is defined for $\mathcal{EL}$ with role-value maps and terminological cycles. Since transitivity is expressible by role-value-maps, this work might be regarded as the first to provide results on an extension of the lcs to transitive roles.

As long as a sublanguage of $\mathcal{FLE}$ does not allow for both existential and value restrictions it is comparatively easy to adapt the existing lcs algorithms to transitive roles as we have seen in the last sections of this report. For $\mathcal{EL}^+$, it is possible to translate a concept C into an equivalent one in $\mathcal{EL}$. Thus, all the additional restrictions imposed by transitive roles in C are made explicit. This simple approach, however, does not work for $\mathcal{FLE}^+$-concept descriptions, as the following example illustrates.

Example 21 *Consider the $\mathcal{FLE}^+$-concept description $C_{ex} := (\forall r.\exists r.A) \sqcap \exists r.A$, where r is transitive. To explicitly satisfy the (transitive) value restriction, we need to propagate $\forall r.\exists r.A$ to the existential restriction. This yields $(\forall r.\exists r.A) \sqcap \exists r.(A \sqcap \exists r.A \sqcap \forall r.\exists r.A)$ which equals $(\forall r.\exists r.A) \sqcap \exists r.(A \sqcap C_{ex})$. Obviously, an attempt of exhaustive propagation would not terminate.*

Hence, our first aim is to find a finite representation of $\mathcal{FLE}^+$-concept descriptions in which the transitivity of roles is made explicit. Such a representation is introduced by the following section.

5.1 Description Graphs

In this section we will not only introduce description graphs as a syntactic construct but also provide a model-theoretic semantics for them—similar to the semantics of concept descriptions. This makes it easier to examine the equivalence between a concept description and a description graphs directly, i.e., without re-translation of the description graph back into a concept.

Definition 22 (description graph) *Let $\mathcal{G} := (V, E, v_0, \ell_V, \ell_E)$ be a rooted, directed, and connected graph with labeling functions for vertices and edges. The labeling function ℓ_V assigns a set of concept descriptions to every vertex in V and ℓ_E assigns a label of the form Qr to every edge in E, where $Q \in \{\forall, \exists\}$ and $r \in N_R \cup N_R^T$. An edge labeled $\forall r$ is called* forall-edge*, an edge labeled $\exists r$* exists-edge*. If every vertex v in $\mathcal{G}$ has at most one outgoing forall-edge per role r then it is called a* description graph.

For the sake of simplicity, we use the notation $(v\, Qr\, w) \in E$ to express that (i) $(v, w) \in E$ and (ii) $\ell_E(v, w) = \{Qr\}$. Note that description graphs can be cyclic. Like concept descriptions, description graphs are interpreted w.r.t. a model-theoretic semantics to be introduced next.

Definition 23 (semantics of description graphs) *Let $\mathcal{G} := (V, E, v_0, \ell_V, \ell_E)$ be a description graph and let $\mathcal{I} := (\Delta, \cdot^I)$ be an interpretation. A mapping $\pi : V \to 2^{\Delta^{\mathcal{I}}} \setminus \emptyset$ is called a* model mapping *iff for all $v, w \in V$ it holds that:*

- $\pi(v) \subseteq C^{\mathcal{I}}$ *for all* $C \in \ell(v)$*;*
- *if* $(v\,\exists r\,w) \in E$ *for* $r \in N_R$ *and* $x \in \pi(v)$ *then there exists some* $y \in \Delta^{\mathcal{I}}$ *with* $(x, y) \in r^{\mathcal{I}}$ *and* $y \in \pi(w)$*;*
- *if* $(v\,\exists r\,w) \in E$ *for* $r \in N_R^T$ *and* $x \in \pi(v)$ *then there exists some* $y \in \Delta^{\mathcal{I}}$ *with* $(x, y) \in (r^{\mathcal{I}})^+$ *and* $y \in \pi(w)$*;*
- *if* $(v\,\forall r\,w) \in E$ *for* $r \in N_R \cup N_R^T$ *and* $x \in \pi(v)$ *then* $(x, y) \in r^{\mathcal{I}}$ *implies* $y \in \pi(w)$*.*

For a given $x \in \Delta^{\mathcal{I}}$*, define* $\mathcal{I}, x \models \mathcal{G}$ *iff there is a model mapping* π *with* $x \in \pi(v_0)$*. The semantics of* $\mathcal{G}$ *w.r.t.* $\mathcal{I}$ *is defined as* $\mathcal{G}^{\mathcal{I}} := \{x \in \Delta^{\mathcal{I}} \mid \mathcal{I}, x \models \mathcal{G}_C\}$*.*

There is a similarity between the semantics of description graphs and that of concept descriptions as defined in Section 2. A (transitive) $\exists r$-edge $(v\,\exists r\,w)$ like an existential restriction implies a corresponding r-edge (r-path) for all $x \in \pi(v)$ in the model. Similarly, every $\forall r$-edge $(v\,\forall r\,w)$ imposes restrictions on every witness in the model reachable via an r-edge from some $x \in \pi(v)$.

Regarded as a description graph the syntax tree of every $\mathcal{FLE}$-concept description C is equivalent to C. This, however, is not generally true of $\mathcal{FLE}^+$-concept descriptions. Moreover, there are description graphs for which no equivalent $\mathcal{FLE}^+$-concept description exists. One example is a graph $\mathcal{G}$ consisting of two vertices v_0 and v_1 connected by two existential edges $(v_0\,\exists r\,v_1)$ and $(v_0\,\exists s\,v_1)$. There is no equivalent concept because an $\mathcal{FLE}^+$-concept description cannot express the fact that the *same* successor is required in both role restrictions. Ultimately, however, we are interested in description graphs guaranteed to represent concept descriptions. To this end, we introduce six conditions to restrict description graphs further, leading to the notion of simple description graphs.

As a prerequisite, we need to specify the notion of a simulation relation for description graphs.

Definition 24 (simulation relation) *For* $i \in \{1, 2\}$*, let* $\mathcal{G}_i := (V_i, E_i, v_{0i}, \ell_{V_i}, \ell_{E_i})$ *be description graphs. Then,* $\mathcal{G}_2 \precsim \mathcal{G}_1$ *iff there exists a relation* $R \subseteq V_2 \times V_1$ *with:*

1. $(v_{02}, v_{01}) \in R$
2. $\ell_V(v) \cap N_C \subseteq \ell_V(v') \cap N_C$ *for all* $(v, v') \in R$*.*

3. *If* $(v\, Qr\, w) \in E_2$ *and* $(v, v') \in R$ *then there exists a vertex* $w' \in V_1$ *such that* $(v'\, Qr\, w') \in E_1$ *and* $(w, w') \in R$.

For vertices $v_1 \in V_1$ *and* $v_2 \in V_2$*, denote by* $\mathcal{G}_2(v_2) \precsim \mathcal{G}_1(v_1)$ *the fact that a simulation relation* R *exists between the subgraph of* $\mathcal{G}_2$ *reachable from* v_2 *and the subgraph of* $\mathcal{G}_1$ *reachable from* v_1*. In particular, this implies* $(v_2, v_1) \in R$.

With these preliminaries, simple description graphs can be introduced.

Definition 25 (simple description graph) *Let* $\mathcal{G} := (V, E, v_0, \ell_V, \ell_E)$ *be a description graph.* $\mathcal{G}$ *is a* simple description graph *iff the following properties hold.*

1. *There exists a spanning tree s.t.,* $\mathcal{G}$ *has no forall-forward edges and no cross edges. Every exists-forward edge only connects vertices connected by a path of exists-tree edges w.r.t. one transitive role.*

2. *If* $(v_0\, Q_0 r_0\, v_1 \ldots v_{n-1}\, Q_{n-1} r_{n-1}\, v_0)$ *is a cycle in* E *with pairwise distinct vertices then there exists one transitive role* r *with* $r_i = r$ *for all* i.

3. *If* $(v_0\, Q_0 r\, v_1 \ldots v_{n-1}\, Q_{n-1} r\, v_0)$ *is a cycle in* E *with pairwise distinct vertices and* $r \in N_R^T$ *then* v_0 *has a* $\forall r$*-successor.*

4. *If* $\{(u\, \forall r\, v), (u\, \exists r\, w)\} \subseteq E$ *then* $\mathcal{G}(v) \precsim \mathcal{G}(w)$*. If* $r \in N_R^T$ *then there exists a vertex* w' *such that* $(w\, \forall r\, w') \in E$ *and* $\mathcal{G}(v) \precsim \mathcal{G}(w')$.

5. *If* $(u\, \forall r\, v) \in E$ *with* $r \in N_R^T$ *then there exists a vertex* v' *such that* $(v\, \forall r\, v') \in E$ *and* $\mathcal{G}(v) \precsim \mathcal{G}(v')$.

6. *If* $B \in \ell(v)$ *then* $\mathcal{G}_B \precsim \mathcal{G}(v)$*, where* B *is a* $\mathcal{FLE}^+$*-concept description and* $v \in V$.

The idea behind the above definition to is imitate the propagation of existential and value restrictions in the graph structure. For instance, Condition 4 ensures that no subgraph representing an existential restriction may be more general that a corresponding subgraph representing a value restriction. Hence, a value restriction must be propagated over all existential restrictions. Condition 5 similarly ensures that value restrictions over transitive roles are propagated to deeper role levels, as $\forall r.A$ implies $\forall r.(A \sqcap (\forall r.A))$ and so on. Conditions 2 and 3 ensure that cycles cannot occur arbitrarily. The last condition guarantees that the reachability graph of a vertex is "according" the label set of that vertex. The first condition excludes a number of irregularities which would make the proofs over description graphs more intricate. The following lemma can be shown for all description graphs.

Lemma 26 *Let $\mathcal{G}$ and $\mathcal{H}$ be description graphs with $\mathcal{H} \precsim \mathcal{G}$. Then $\mathcal{G} \sqsubseteq \mathcal{H}$.*

Proof. Let I be a model of $\mathcal{G}$, i.e., there is an $x \in \Delta^I$ with $I, x \models \mathcal{G}$. It is to be shown that $I, x \models \mathcal{H}$. To this end, we construct a model mapping $\pi'\colon V_\mathcal{H} \to \Delta^I$ such that $x \in \pi'(\text{root}(\mathcal{H}))$.

The simulation $\mathcal{H} \precsim \mathcal{G}$ implies the existence of a simulation relation $S \subseteq V_\mathcal{H} \times V_\mathcal{G}$ which respects the properties stated in Definition 24. If $I, x \models \mathcal{G}$ then there exists a model mapping π with $x \in \pi(\text{root}(\mathcal{G}))$. Define

$$\pi'\colon V_\mathcal{H} \to \Delta^I$$
$$v \mapsto \bigcup_{(v,w)\in S} \pi(w).$$

We have to show that π' is a model mapping and that $x \in \pi'(\text{root}(\mathcal{H}))$. The second claim is not difficult to prove. The definition of the simulation relation S guarantees that $(\text{root}(\mathcal{H}), \text{root}(\mathcal{G})) \in S$ and the model mapping π maps $\text{root}(\mathcal{G})$ onto a set containing x.

Consider an arbitrary $v \in V_\mathcal{H}$ and an $x \in \pi'(v)$. Then there is a vertex $w \in V_\mathcal{G}$ such that $(v, w) \in S$ and $x \in \pi(w)$.

- For the pair (v, w) the simulation relation guarantees that $\ell(v) \subseteq \ell(w)$. As the model mapping π ensures that $x \in A^I$ for all $A \in \ell(w)$ we consequently obtain $x \in A^I$ also for all $A \in \ell(v)$.

- If $(v\, \exists r\, v') \in E_\mathcal{H}$ for a transitive role r then the simulation relation S guarantees the existence of a vertex $w' \in V_\mathcal{G}$ such that $(v', w') \in S$ and $(w\, \exists r\, w') \in E_\mathcal{G}$. Due to this edge π guarantees some $y \in \pi(w')$ such that $(x, y) \in (r^I)^*$. As by construction y occurs in $\pi'(v')$ we find that π' has the required property. The case of an existential edge w.r.t. a non-transitive role is analogous.

- If $(v\, \forall r\, v') \in E_\mathcal{H}$ for a transitive role r then the simulation relation S again guarantees an analogous edge $(w\, \forall r\, w') \in E_\mathcal{G}$ with $(v', w') \in S$. Assume that $(x, y) \in (r^I)^*$ for some $y \in \Delta^I$. Due to the model mapping π we know that $y \in \pi(w')$. As $(v', w') \in S$ we find that $y \in \pi'(v')$, concluding the argument.

■

Note that the reverse does not hold in general. For a non-transitive role r, consider the two graphs $\mathcal{G} := (\{v_0, v_1, v_2\}, \{(v_0\,\forall r\, v_1), (v_0\,\exists r\, v_2)\}, \ell_\mathcal{G})$ and $\mathcal{H} := (\{w_0, w_1, w_2\}, \{(w_0\,\forall r\, w_1), (w_0\,\exists r\, w_2)\}, \ell_\mathcal{H})$ where $\ell_\mathcal{G}(v_0) = \ell_\mathcal{H}(w_0) = \emptyset$ and $\ell_\mathcal{G}(v_1) = \ell_\mathcal{H}(w_1) = \ell_\mathcal{H}(w_2) = \{A\}$. The only difference between $\mathcal{G}$ and $\mathcal{H}$ lies in the label of the existential successor of the root vertex. Here we have $\ell_\mathcal{G}(v_2) = \emptyset$ and $\ell_\mathcal{H}(w_2) = \{A\}$. It is easy to show that $\mathcal{G} \equiv \mathcal{H}$ but $\mathcal{H} \not\preceq \mathcal{G}$.

Having defined syntax and semantics of description graphs in general the next step is to translate $\mathcal{FLE}^+$-concept descriptions into equivalent description graphs.

5.2 From $\mathcal{FLE}^+$-concept descriptions to $\mathcal{FLE}^+$-description graphs

To show that every $\mathcal{FLE}^+$-concept description has a corresponding $\mathcal{FLE}^+$-description graph we devise a translation of concept descriptions to $\mathcal{FLE}^+$-description graphs. As a technical prerequisite, we require a normal form for $\mathcal{FLE}$-concept descriptions, as introduced in [3]. The purpose of this normal form is merely to flatten conjunctions, to make the top-concept explicit, and to propagate value restrictions over existential restrictions. The problem of implicit information induced by transitive roles remains untouched here.

Definition 27 ($\mathcal{FLE}$ 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*

1)	$\forall r.\top \longrightarrow \top$		3)	$\forall r.E \sqcap \forall r.F \longrightarrow \forall r.(E \sqcap F)$
2)	$E \sqcap \top \longrightarrow E$		4)	$\forall r.E \sqcap \exists r.F \longrightarrow \forall r.E \sqcap \exists r.(E \sqcap F)$
			5)	$E \sqcap (F \sqcap G) \longrightarrow E \sqcap F \sqcap G.$

*A concept description is in $\mathcal{FLE}$-*normal form *if the $\mathcal{FLE}$-normalization rules have bee applied to it exhaustively.*

The normalization 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. It has been shown in [3] that exhaustive application of the $\mathcal{FLE}$-normalization rules may produce concept descriptions of size exponential in the size of the original concept description. During the translation of an $\mathcal{FLE}^+$-concept description into an $\mathcal{FLE}^+$-description graph we need to apply the $\mathcal{FLE}$-normalization rules only to the top level of the $\mathcal{FLE}^+$-concept.

The following definition provides the framework of the translation of an $\mathcal{FLE}^+$-concept description into a description graph. For a given concept description C we start with an empty description graph $\mathcal{G}$ consisting only of a root vertex v_0 with C in its label. Then we exhaustively apply graph generation rules (defined in detail in Figure 7) producing new vertices and edges. In this process, tree edges ($E^{\mathcal{D}}$), forward edges (E^+), and back edges ($E^{\circlearrowleft}$) are distinguished. As soon as no production rules are applicable, all non-atomic concept descriptions are removed from the label sets of $\mathcal{G}$ and the graph is returned.

For the actual definition, a shorthand notation needs to be introduced first. For a set $\{C_1, \ldots, C_n\}$ of $\mathcal{FLE}^+$-concept descriptions, let $\{C_1, \ldots, C_n\}^*$ denote the corresponding set in which (i) the $\mathcal{FLE}^+$normalization rules defined above have been applied exhaustively on the top-level of every C_i and (ii) every C_i is split into its conjuncts. Observe that there is at most one value restriction per role r in $\{C_1, \ldots, C_n\}^*$.

Definition 28 ($\mathcal{FLE}^+$-description graph) *Let C be a $\mathcal{FLE}^+$-concept description. The $\mathcal{FLE}^+$-*description graph *$\mathcal{G}_C$ is obtained by the following procedure:*

1. *Initialize the sets $V := \{v_0\}$, $\ell_V = \ell_V(v_0) = \{C\}^*$, and* $E := E^+ := E^{\mathcal{D}} := E^{\circlearrowleft} := \emptyset$.
2. *Apply the $\mathcal{FLE}^+$-description graph generation rules from Figure 7 exhaustively to obtain $\mathcal{G}'_C := (V, E, v_0, \ell_V, \ell_E)$, where $E = E^{\mathcal{D}} \cup E^{\circlearrowleft} \cup E^+$.*
3. *Reduce the label sets of vertices: $\forall v \in V$: $\ell'_V(v) := \ell_V(v) \cap N_C$.*
4. *return $\mathcal{G}_C := (V, E, v_0, \ell'_V, \ell_E)$.*

All non-atomic concept descriptions in the label sets of the vertices of $\mathcal{G}$ are discarded afterwards because their information (as we shall see) is then represented by the structure of the graph. It remains to define the generation rules used in Step 2 of the above definition.

Figure 7 shows the relevant generation rules referred to in Definition 28. For every v, $\rho(v)$ denotes the (unique) path from v_0 to v w.r.t. tree edges.

R$_\exists$: If $(\exists r.C') \in \ell_V(v)$, $(\forall r.C'') \notin \ell_V(v)$ for some C', C'', and
there is no $v'' \in V : (v, \exists r, v'') \in E^{\mathcal{D}} \cup E^{\circlearrowleft} \;\wedge\; \{C'\}^* = \ell_V(v'')$,
then if there is $v_i \in V : v_i$ appears in $\rho(v) \wedge \ell_V(v_i) = \{C'\}^*$,
then $E^{\circlearrowleft} := E^{\circlearrowleft} \cup \{(v, \exists r, v_i)\}$,
else $V := V \cup \{v'\}$, $E^{\mathcal{D}} := E^{\mathcal{D}} \cup \{(v, \exists r, v')\}$, $\ell_V(v') := \{C'\}^*$.

R$_{\exists\forall}$: If $r \in N_R$, and $\{(\exists r.C'), (\forall r.C'')\} \subseteq \ell_V(v)$ for some C', C'', and
there is no $v'' \in V : (v, \exists r, v'') \in E^{\mathcal{D}} \cup E^{\circlearrowleft} \;\wedge\; \{C'\}^* = \ell_V(v'')$,
then if there is $v_i \in V : v_i$ appears in $\rho(v) \wedge \ell_V(v_i) = \{C'\}^*$
then $E^{\circlearrowleft} := E^{\circlearrowleft} \cup \{(v, \exists r, v_i)\}$,
else $V := V \cup \{v'\}$, $E^{\mathcal{D}} := E^{\mathcal{D}} \cup \{(v, \exists r, v')\}$, $\ell_V(v') := \{C'\}^*$.

R$_{\exists\forall+}$: If $r \in N_R^T$, and $\{(\exists r.C'), (\forall r.C'')\} \subseteq \ell_V(v)$ for some C', C'', and
there is no $v'' \in V : (v, \exists r, v'') \in E^{\mathcal{D}} \cup E^{\circlearrowleft} \wedge \{C', \forall r.C''\} = \ell_V(v'')$,
then if there is $v_i \in V : v_i$ appears in $\rho(v) \wedge \ell_V(v_i) = \{C', \forall r.C''\}^*$
then $E^{\circlearrowleft} := E^{\circlearrowleft} \cup \{(v, \exists r, v_i)\}$,
else $V := V \cup \{v'\}$, $E^{\mathcal{D}} := E^{\mathcal{D}} \cup \{(v, \exists r, v')\}$, $\ell_V(v') := \{C', \forall r.C''\}^*$.

R$_\forall$: If $r \in N_R$, and $(\forall r.C') \in \ell_V(v)$ for some C', and
there is no $v'' \in V : (v, \forall r, v'') \in E^{\mathcal{D}} \cup E^{\circlearrowleft}$
then if there is $v_i \in V : v_i$ appears in $\rho(v) \wedge \ell_V(v_i) = \{C'\}^*$
then $E^{\circlearrowleft} := E^{\circlearrowleft} \cup \{(v, \forall r, v_i)\}$,
else $V := V \cup \{v'\}$, $E^{\mathcal{D}} := E^{\mathcal{D}} \cup \{(v, \forall r, v')\}$, $\ell_V(v') := \{C'\}^*$.

R$_{\forall+}$: If $r \in N_R^T$, and $(\forall r.C') \in \ell_V(v)$ for some C', and
there is no $v'' \in V : (v, \forall r, v'') \in E^{\mathcal{D}} \cup E^{\circlearrowleft}$
then if there is $v_i \in V : v_i$ appears in $\rho(v) \wedge \ell_V(v_i) = \{C', \forall r.C'\}^*$
then $E^{\circlearrowleft} := E^{\circlearrowleft} \cup \{(v, \forall r, v_i)\}$,
else $V := V \cup \{v'\}$, $E^{\mathcal{D}} := E^{\mathcal{D}} \cup \{(v, \forall r, v')\}$, $\ell_V(v') := \{C', \forall r.C'\}^*$.

R$_{E^+}$: If $r \in N_R^T$, and $\{(v, \exists r, v'), (v', \exists r, v'')\} \in E^{\mathcal{D}}$ and $(v, \exists r, v'') \notin E^+$
then $E^+ := E^+ \cup \{(v, \exists r, v'')\}$

Figure 7: $\mathcal{FLE}^+$-Description Graph Generation Rules.

Intuitively, the idea of the rules is to use the concept descriptions occurring in the label set of a vertex v to extend the description graph "accordingly" in the following sense: if an existential restriction $\exists r.C$ occurs in $\ell_V(v)$ then a vertex w must be introduced (or probably only found) such that (i) w is connected to v by an exists-edge and (ii) a concept equivalent to C occurs in $\ell_V(w)$. Moreover, a value restriction $\forall r.D$ probably also occurring in $\ell_V(v)$

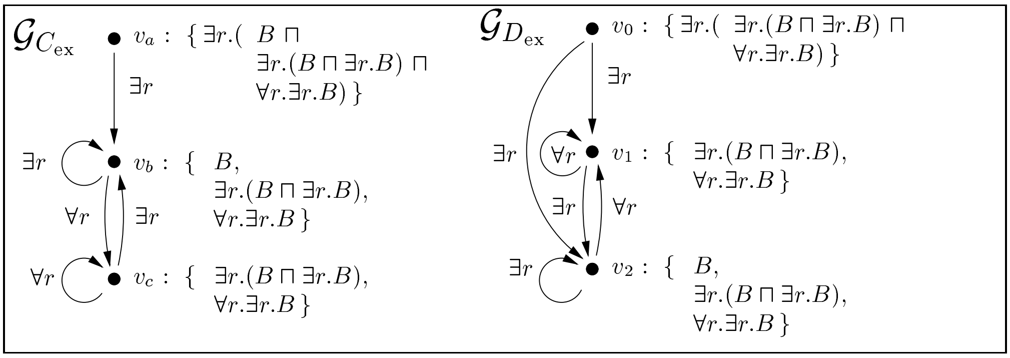

Figure 8: $\mathcal{FLE}^+$-description graphs

must be propagated to $\ell(w)$ likewise.

Starting at a given vertex v, the rules $\mathbf{R}_{\exists}$, $\mathbf{R}_{\exists\forall}$, and $\mathbf{R}_{\exists\forall+}$ all produce new exists-edges, possibly to a newly generated vertex. $\mathbf{R}_{\exists}$ applies if only an existential restriction is present in $\ell_V(v)$, $\mathbf{R}_{\exists\forall}$ applies if an additional value restriction (w.r.t. the same non-transitive role) is present, and $\mathbf{R}_{\exists\forall+}$ covers the case of an additional value restriction for the transitive case. Similarly, $\mathbf{R}_{\forall}$ and $\mathbf{R}_{\forall+}$ address the case where only a value restriction (non-transitive or transitive) is present. The rule $\mathbf{R}_{\exists+}$ never introduces new vertices but only adds forward edges over exists-paths w.r.t. one transitive role.

To avoid generating infinitely many new vertices, every generation rule has a *blocking condition*[1] testing whether or not a new vertex can be avoided by a back edge to an already existing one. For every vertex v, a back edge to an ancestor u of v is added instead of a new vertex w if the ancestor vertex has the same label set the new vertex would get, i.e., $\ell_V(u) = \ell_V(w)$. The vertex u is regarded as ancestor of v iff u lies on a (the) tree-path from the root vertex to v. Note that the condition $\ell_V(u) = \ell_V(w)$ determines u uniquely and that $v = w$ is not excepted.

The following example shows the corresponding $\mathcal{FLE}^+$-description graph of two simple $\mathcal{FLE}^+$-concept descriptions.

Example 29 *Let* $C_{\rm ex} := \exists r.(B \sqcap \exists r.B \sqcap \forall r.\exists r.B)$ *and* $D_{\rm ex} := \exists r.(\exists r.B \sqcap \forall r.\exists r.B)$ *for a transitive role* r *and an atomic concept* B*. The corresponding* $\mathcal{FLE}^+$*-description graphs are depicted in Figure 8. The figure also shows*

[1] Blocking strategies originally have been introduced in the DL context in [9] for a tableaux-based satisfiability tester for expressive DLs. In the relevant work, blocking controlled the generation of new sub-tableaux in the computation of a completed tableau.

the normalized label sets of every vertex. Note that the non-atomic concept descriptions in the label sets are used only during the generation of the description graphs.

It remains to be shown that the resulting $\mathcal{FLE}^+$-description graphs are in fact equivalent to the original concept descriptions.

Lemma 30 *Let C be an $\mathcal{FLE}^+$-concept description. Then $C \equiv \mathcal{G}_C$*

PROOF. ($\sqsupseteq$). Consider a model I of $\mathcal{G}_C$. Show that $x \in \mathcal{G}_C^I$ implies $x \in C^I$. If $I, x \models \mathcal{G}_C$ then there exists a model mapping $\pi\colon V_C \to \Delta^I$ with $x \in \pi(\text{root}(\mathcal{G}_C))$. To show that $x \in C^I$, it is sufficient to show that the witnesses of every vertex $v \in V_C$ are also witnesses of every concept in label(v). For a given $v \in V$, let $D \in \text{label}(v)$. Proof by induction on the structure of D.

- $D = A \in N_C$
  Then the model mapping π guarantees that $\pi(v) \in A^I$.

- $D = \exists r.(D'_1 \sqcap \cdots \sqcap D'_n)$ with $r \in N_R^T$
  Then by construction of $\mathcal{G}_C$ we know that there exists an $\exists r$-successor w of v such that $D'_i \in \text{label}(w)$ for every i. By induction hypothesis we know that every $y \in \pi(w)$ is a witness of every D'_i. According to the definition of π, for every $x \in \pi(v)$ and $y \in \pi(w)$ it holds that $(x, y) \in (r^I)^*$. The fact that w is a witness of all D'_i thus implies that every $x \in \pi(v)$ is a witness of D.

- $D = \exists r.(D'_1 \sqcap \cdots \sqcap D'_n)$ with $r \in N_R$
  Analogous, only that r^I is relevant instead of $(r^I)^*$.

- $D = \forall r.(D'_1 \sqcap \cdots \sqcap D'_n)$ with $r \in N_R^T$
  Then by construction of $\mathcal{G}_C$ we know that an $\forall r$-successor w of v exists such that every D'_i is in label(w). Again, by induction hypothesis every $y \in \pi(w)$ is a witness of every D'_i. If $x \in \pi(v)$ then the edge $(v\,\forall r\,w)$ by definition if π implies that every $y \in \Delta^I$ with $(x, y) \in (r^I)^*$ occurs in $\pi(w)$. Hence, every (transitive) r-successor of x in I is a witness of every D'_i. Consequently, x is a witness of D.

- $D = \forall r.(D'_1 \sqcap \cdots \sqcap D'_n)$ with $r \in N_R^T$
  Analogous.

($\sqsubseteq$). Consider a model I of C. Show that $x \in \mathcal{G}_C^I$ implies $x \in C^I$. If $I, x \models \mathcal{G}_C$ then Lemma 39 states that a witness relation ρ exists between $\mathcal{G}_C$ and I. It is easy to see that the mapping

$$\begin{aligned} \pi \colon V_C &\to \Delta^I \\ v &\mapsto \rho(v) \end{aligned}$$

is a valid model mapping between $\mathcal{G}_C$ and I with $x \in \pi(\mathrm{root}(\mathcal{G}_C))$. ■

Lemma 31 *Let C be an $\mathcal{FLE}^+$-concept description. Then $\mathcal{G}_C$ is a simple description graph.*

Proof. We have to show that $\mathcal{G}_C$ respects Conditions 1 to 5 from Definition 25.

1. The procedure from Definition 28 introduces $\forall$-edges $(v\,\forall r\,w)$ only if a value restriction $\forall r.D$ is present in label(v). If label(v) equals a label set on the path from root($\mathcal{G}_C$) to v then $(v\,\forall r\,w)$ becomes a back edge. Otherwise, w is introduced as a new node. Hence, $\mathcal{G}_C$ contains no $\forall$-forward edges.

   The argument for cross edges is analogous. Edges newly introduced by he procedure from Definition 28 either point to a newly introduced vertex or to a predecessor of the starting vertex.

   As the last step in the procedure, $\exists$-forward edges are introduced over every existential path (of length greater that 1) w.r.t. one fixed transitive role. Before that step no existential forward edges are introduced as can be seen analogously to the case of $\forall$-forward edges above.

   Assume that $v \neq w$ are connected both via an $\exists r$-edge and a $\forall s$-edge. As argued above, both edges are neither forward edges, because $(v\,\forall s\,w)$ is no forward edge, nor tree edges, because then their destination vertices would be different. As a result of Condition 2 we also know that $s = r$ and that r is transitive.

2. Consider a cycle $(v_0\,Qr_0\,v_1 \ldots v_{n-1}\,Qr_{n-1}\,v_0)$ with pairwise distinct vertices v_i. For all $i < n-1$, the edge $(v_i\,Qr_i\,v_{i+1})$ are tree edges, $(v_{n-1}\,Qr_{n-1}\,v_0)$ is a back edge. The existence of the back edge implies that during the execution of the procedure from Definition 28, the label set of the Qr_{n-1}-successor of v_{n-1} was found to be equal to

label(v_0). Assume that there exists an index i such that $r_i \neq r_{i+1}$. Then the maximum role depth of concepts in label(v_{i+2}) is smaller than the maximum in every label(v_j) with $j \leq i$ because no propagation occurs over inhomogeneous role paths. Since the maximum role depth of concepts in the label set of vertices cannot increase over tree edges it follows that the maximum role depth of concepts in the $Q_{n-1}r_{n-1}$-successor of v_{n-1} cannot equal that of label(v_0). Hence, the two label sets cannot be equal. Consequently, all role names $r_i = r$ for all i and for some role name r. It is obvious that r must be transitive because otherwise the maximum role depth of concepts in the respective label sets would decrease by 1 in every transition of an edge.

3. Consider a cycle $(v_0\, Qr\, v_1 \dots v_{n-1}\, Qr\, v_0)$ with pairwise distinct vertices v_i. The above Condition allows us to restrict our attention to cycles over only one transitive role r. In the procedure from Definition 28, a value-restriction in any label(v_i) would be propagated to all other sets label(v_j) due to the transitivity of r. Hence, assume for every i that no $\forall r$-successor of v_i exists. In this case, no propagation occurs, implying that the maximum role depth of concepts in label(v_i) decreases with greater i. Again, this contradicts the back edge $(v_{n-1}\, Qr\, v_0)$.

4. $\forall$-$\exists$-Prop
   Consider vertices u, v, w with $\{(u\, \forall r\, v), (u\, \exists r\, w)\} \subseteq E_C$ where r is a transitive role. In the procedure from Definition 28, the label set of u contains a value restriction $\forall r.D$ and an existential restriction $\exists r.E$ such that label(v) $= \{D, \forall r.D\}$ and $\sqcap$label(w) $\sqsupseteq \sqcap\{E, D, \forall r.D\}$. Hence, there is a subsumption relation of the concepts $\sqcap$label(w) $\sqsubseteq$ $\sqcap$label(v). By Lemma 44 this implies $\mathcal{G}_{\bigsqcap \text{label}(v)} \rightleftarrows \mathcal{G}_{\bigsqcap \text{label}(w)}$. It is easy to see that there are simulation relations between $\mathcal{G}_{\bigsqcap \text{label}(v)}$ and the subgraph of $\mathcal{G}_C$ reachable from v because both are determined by label(v). The same holds for $\mathcal{G}_{\bigsqcap \text{label}(w)}$ and the subgraph of $\mathcal{G}_C$ reachable from w. Consequently, we can devise a simulation relation between the two reachability subgraphs as a combination of three simulation relations. As $\mathcal{G}_{\bigsqcap \text{label}(v)} \rightleftarrows \mathcal{G}_{\bigsqcap \text{label}(w)}$ implies that there exist simulation relations which contain the pair of the roots of $\mathcal{G}_{\bigsqcap \text{label}(v)}$ and $\mathcal{G}_{\bigsqcap \text{label}(w)}$ it is clear that the combined simulation relation contains the pair (v, w).

   We have seen that label(w) contains the concept $\forall r.D$. By the procedure from Definition 28, this implies the existence of a $\forall r$-successor w' of

w with $\sqcap \text{label}(w') \sqsupseteq \sqcap\{D, \forall r.D\}$. Hence, we again have $\sqcap \text{label}(w') \sqsubseteq \sqcap \text{label}(v)$. The rest of the argument is analogous.

In case of a non-transitive role r we can use the same approach as above only that the value restriction $\forall r.D$ is not propagated to existential or universal successors of v and w. Moreover, there is nothing to show for a universal successor w' of w.

5. $\forall$-Prop
   Consider vertices u, v with $(u\, \forall r\, v) \in E_C$ where r is a transitive role. By definition of the $\mathcal{FLE}^+$-description graph generation procedure from Definition 28, the label set of u contains a value restriction $\forall r.D$ such that $\sqcap \text{label}(v) \equiv D \sqcap \forall r.D$. By definition of the $\mathcal{FLE}^+$-description graph generation procedure, there exists a $\forall r$-successor v' of v such that $\sqcap \text{label}(v') \equiv D \sqcap \forall r.D$. Analogous to the previous case Lemma 44 yields $\mathcal{G}_{\sqcap \text{label}(v)} \rightleftarrows \mathcal{G}_{\sqcap \text{label}(v')}$. Based on a simulation relation between $\mathcal{G}_{\sqcap \text{label}(v)}$ and $\mathcal{G}_{\sqcap \text{label}(v')}$ we can again construct the relevant simulation relations on the reachability subgraphs of v and v'.

∎

As a result, we know how to encode the information represented by $\mathcal{FLE}^+$-concept descriptions in $\mathcal{FLE}^+$-description graphs. Our next step is to find a way to translate description graphs back to concept descriptions.

5.3 Translation of simple description graphs into $\mathcal{FLE}^+$-concept descriptions

It has already been mentioned in Section 5.1 that description graphs exist without an equivalent $\mathcal{FLE}^+$-concept description. We shall see that it suffices to restrict our backward translation procedure to the class of simple description graphs introduced in the previous section.

For the backward translation from description graphs to concept descriptions we may not rely on complex concept descriptions in the label sets of the graphs in question. On the contrary, the idea is to re-build complex concept descriptions in the label sets while preserving equivalence to the original description graph. This process is continued until the desired concept description occurs in the root label. Note that this strategy is just the reverse of the generation procedure of $\mathcal{FLE}^+$-description graphs, where the label of the root vertex generated the entire description graph.

To formalize the notion of re-building complex labels we devise an operation which modifies a given description graph by altering its label function. Intuitively, the function acc "accumulates" complex concept descriptions in the label sets of the vertices.

Definition 32 *Let $\mathcal{G} := (V, E, v_0, \ell_V, \ell_E)$ be a description graph and $|E| := n$. Then,* $\mathrm{acc}(\mathcal{G}) := (V, E, v_0, \ell'_V, \ell_E)$ *where ℓ'_V is defined as follows. For every $v \in V$,*

$$\begin{aligned}\ell'_V(v) := (\ell_V(v) \cap N_C\,)\\ \cup \bigcup_{r \in N_R \cup N_R^T} \bigcup_{(v\,\exists r\,w) \in E} \exists r. \sqcap \ell_V(w)\\ \cup \bigcup_{r \in N_R \cup N_R^T} \bigcup_{(v\,\forall r\,w) \in E} \left(\forall r. \sqcap(\ell_V(w) \setminus \{\forall r.\top\}) \sqcap \bigsqcap_{(w\,\exists r\,w') \in E} \exists r. \sqcap \ell_V(w')\right).\end{aligned}$$

Define $\mathrm{conc}(\mathcal{G}) := \sqcap \ell_V(v'_0)$*, where v'_0 denotes the root vertex of* $\mathrm{acc}^n(\mathcal{G})$*.*

For every vertex v, the modified label function ℓ'_V contains the same atomic labels as before but additionally has an existential restriction based on the label of every $\exists r$-successor of v. Forall-edges are treated similarly only that a restriction $\forall r.\top$ is ignored. Observe that $\mathrm{acc}(\mathcal{G})$ is still a simple description graph.

To illustrate the effect of the function acc, consider the a simple description graph $\mathcal{G}$ with only one vertex v_0 with a label $\ell_V(v_0) = \{A\}$ and edges $E := \{(v_0, \exists r, v_0), (v_0\,\forall r\,v_0)\}$. In $\mathrm{acc}(\mathcal{G})$ the root vertex has the label $\{A, \exists r.A, \forall r.A\}$. Applying acc again we obtain the root label of $\mathrm{acc}^2(\mathcal{G})$ which equals $\{A, \exists r.(A \sqcap \exists r.A \sqcap \forall r.A), \forall r.(A \sqcap \exists r.A \sqcap \forall r.A)\}$.

The idea now is to show that applying the function acc at most $|E|$ times produces a root label such that the conjunction of all contained concepts is equivalent to $\mathcal{G}$.

Lemma 33 *For every simple description graph $\mathcal{G}$ it holds that $\mathcal{G} \equiv \mathrm{acc}(\mathcal{G})$.*

Proof. Let $\mathcal{G} := (V, E, v_0, \ell_V, \ell_E)$. Show ($\sqsupseteq$). Assume that I is a model of $\mathcal{G}$ and $x_0 \in \mathcal{G}^I$. Then, by definition, there exists a model mapping $\pi\colon V \to \Delta^I$ with $x_0 \in \pi(v_0)$. For every $v \in V$ the modification of $\ell_V(v)$ by acc can be represented in two steps. Firstly, all non-atomic concepts are removed from $\ell_V(v)$ and secondly, new concepts are included for every exists-edge $(v\,\exists r\,w)$ and for every forall edge $(v\,\forall r\,w)$ with a non-empty label $\ell_V(w)$. The first step obviously does not affect the fact that π is a model mapping onto I because, the new label imposes less restrictions on possible models.

For the second step, assume that an existential restriction $\exists r.(E_1 \sqcap \cdots \sqcap E_n)$ for a transitive role r has been added to the label of v. Then, by definition of acc, there exists a vertex $w \in V$ with $(v\,\exists r\,w) \in E$ and $E_i \in \ell_V(w)$ for all i. We know that $y \in E_i^I$ for all $y \in \pi(w)$ and we know that $\pi(w)$ is not empty. Moreover, as π is a model mapping, $(x, y) \in (r^I)^*$. Consequently, x is a witness of $\exists r.(E_1 \sqcap \cdots \sqcap E_n)$. The non-transitive case is analogous.

Assume that a value restriction $\forall r.(E_1 \sqcap \cdots \sqcap E_n)$ for a transitive role r has been added to the label of v. Then, similarly, there is an edge $(v\,\exists r\,w) \in E$ with $E_i \in \ell_V(w)$ for all i. As because I is a model of $\mathcal{G}$, we know for every y with $(x, y) \in (r^I)^*$ that $y \in E_i^I$ for every i . Hence, x is a witness of $\forall r.(E_1 \sqcap \cdots \sqcap E_n)$. The non-transitive case is analogous.

As a result we obtain that π is also a model mapping on $\mathrm{acc}(\mathcal{G})$. Hence, $x_0 \in \mathrm{acc}(\mathcal{G})^I$.

Show ($\sqsubseteq$). Assume that I is a model of $\mathrm{acc}(\mathcal{G})$ and $x_0 \in \mathcal{G}^I$. Then, by definition, there exists a model mapping $\pi\colon V \to \Delta^I$ with $x_0 \in \pi(v_0)$. Note that $\mathrm{acc}(\mathcal{G})$ has the same set of vertices and edges as $\mathcal{G}$. Consider a vertex $v \in V$. Denote by $C_1, \ldots, C_n$ the set of non-atomic concepts present $\ell_V(v)$ before the application of acc. The modification from $\mathrm{acc}(\mathcal{G})$ back to $\mathcal{G}$ can be seen as (1) discarding all non-atomic labels in $\ell_V(v)$ and (2) restoring the original concepts C_i. The first step, as in the previous case, preserves the fact that π is a model mapping onto I. In the second step, concepts C_i are added to the label of v for which we know (Condition 6) that $\mathcal{G}_{C_i} \rightleftarrows \mathcal{G}(v)$. Hence, every $x \in \pi(v)$ is also a witness of every C_i, implying that π is still a model mapping onto I. ∎

As a result we now know that any number of applictions of acc to a simple description graph $\mathcal{G}$ preserves equivalence. Our next step is to show that it suffices to apply acc as often as there are edges in $\mathcal{G}$ to extract a concept description equivalent to $\mathcal{G}$ from its root label. In the following lemma we need the notion of limited reachability graphs which will be introduced in preparation.

Definition 34 *Let* $\mathcal{G} := (V, E, v_0, \ell_V, \ell_E)$ *be a description graph. For a natural number* $i \in \mathbb{N}$ *and a vertex* $v \in V$*, denote by* $\mathrm{reach}_i(v)$ *the subgraph of*

$\mathcal{G}$ induced by all paths of length at most i starting from v.

Obviously, $\text{reach}_0(v) = (\{v\}, \emptyset, v, \ell_V, \ell_E)$ and $\text{reach}_1(\mathcal{G}, v) = (\{v\}, E \cap (\{v\} \times V), v, \ell_V, \ell_E)$. With these preliminaries, we can show that the concept computed by $\text{conc}(\mathcal{G})$ is subsumed by $\mathcal{G}$.

Lemma 35 *For all $v \in V$ and for all $i \in \mathbb{N}$ it holds that $x \in (\ell'_i(v))^I$ implies $x \in \text{reach}_i(v)^I$.*

Proof. Proof by induction on i. In case $i = 0$ we are only concerned with graphs consisting of only one vertex without edges. For these the assertion trivially holds. Assume $i > 0$ and $x \in (\ell'_i(v))^I$. By definition,

$$\begin{aligned}\ell'_i(v) = & \sqcap(\ell_V(v) \cap N_C) \\ & \sqcap \sqcap_r \sqcap_{(v\,\exists r\,w)} \exists r.\ell'_{i-1}(w) \\ & \sqcap \sqcap_r \sqcap_{(v\,\forall r\,w)} \forall r.\big(\ell'_{i-1}(w) \sqcap \sqcap_{(w\,\exists r\,w')} \exists r.\ell'_{i-1}(w')\big)\end{aligned}$$

Let $\text{reach}_i(v) =: (V_{i,v}, E_{i,v}, v, \ell_V, \ell_E)$. We have to show that there exists a model mapping $\pi\colon V_{i,v} \to \Delta^I \setminus \emptyset$ with $x \in \pi(v)$.

As I is a model of $\ell'_i(v)$ it follows that I contains submodels for every existential and value restriction in $\ell'_i(v)$. In I these submodels are reachable from x via edges (or paths) of the respective roles. By definition of $\ell'_i(v)$, every existential restriction for a role r is of the form $\exists r.\ell'_{i-1}(w)$, where $(v\,\exists r\,w) \in E_{i,v}$. Similarly, every value restriction is more specific than $\forall r.\ell'_{i-1}(w)$ with $(v\,\forall r\,w) \in E_{i,v}$.

Consequently, by induction hypothesis there exists a model mapping $\pi_{(v\,Qr\,w)}\colon V_{i-1,w} \to \Delta^I \setminus \emptyset$ from $\text{reach}_{i-1}(w)$ onto I such that every $y \in (\ell'_{i-1}(w))^I$ is in $\pi(w)$. It is easy to see that $\text{reach}_i(v)$ can be represented as a merging of $\text{reach}_1(v)$ and all $\text{reach}_{i-1}(w)$ with $(v\,Qr\,w) \in E_{i,v}$. Note that these subgraphs are not necessarily disjoint. Our aim now is to construct $\pi\colon V_{i,v} \to \Delta^I \setminus \emptyset$ from the individual model mappings $\pi_{(v\,Qr\,w)}$:

$$\pi(u) := \begin{cases} \bigcap\{\pi_{(v\,Qr\,w)}(u) \mid u \in V_{i-1,w}\}^I & \text{for } u \neq v \\ \ell'_i(v)^I \cap \bigcap\{\pi_{(v\,Qr\,w)}(v) \mid v \in V_{i-1,w}\}^I & \text{otherwise} \end{cases}$$

A necessary condition for π to be a model mapping is that $\emptyset$ does not occur as an image of a vertex u, i.e., the intersection over all $\pi_{(v\,Qr\,w)}(u)$ is never empty.

For a vertex $u \in V_{i-1,w_1} \cap V_{i-1,w_2}$, assume that the intersection $\pi_{(v\,Q_1 r_1\,w_1)}(u) \cap \pi_{(v\,Q_2 r_2\,w_2)}(u)$ is empty. This implies that no witness y exists in I which meets the restrictions imposed by the edges starting from u in both $\text{reach}_{i-1}(w_1)$ and $\text{reach}_{i-1}(w_2)$. However, already in the first step $\ell'_1(u)$ contains a value or existential restriction for *every* edge starting from u (excepting trivial value restrictions) implying that eventually $\ell'_i(v)$ contains a concept description which enforces a witness in I meeting all the restrictions originating from the vertex u in both subgraphs $\text{reach}_{i-1}(w_1)$ and $\text{reach}_{i-1}(w_2)$.

By construction, $x \in \pi(v)$. Hence, we still have to show that π is in fact a model mapping from $V_{i,v}$ onto Δ^I. The fact that $\pi(u) \subseteq C^I$ for all $C \in \ell_V(u)$ either holds because of an existing model mapping $\pi_{(v\,Qr\,w)}$ with $u \in V_{i-1,w}$ in case $u \neq v$ or because of the fact that every $y \in \pi(v)$ is a witness of $\ell'_i(v)$.

By construction of π we need to show the remaining edge-conditions only for edges of the form $(v\,Qr\,w)$ not part of one of the subgraphs for which sub-model mappings have already been obtained by induction hypothesis. Nevertheless, we need to discriminate the case of cyclic edges of the form $(v\,Qr\,v)$.

For $w \neq v$, consider an exists-edge $(v\,\exists r\,w) \in E_{i,v}$ w.r.t. a transitive role r and $z \in \pi(v)$. By definition, the concept $\ell'_i(v)$ contains an existential restriction $\exists r.\ell'_{i-1}(w)$. Since $z \in \pi(v)$ we know that a witness $z' \in \Delta^I$ exists such that $(z, z') \in (r^I)^*$ and $z' \in \ell'_{i-1}(w)^I$. By induction hypothesis, $\ell'_{i-1}(w)$ is more specific than $\text{reach}_{i-1}(w)$ implying that z' is also a witness of $\text{reach}_{i-1}(w)$. Consequently, z' appears in $\pi_{(v\,\exists r\,w)}(w)$ which by construction implies $z' \in \pi(w)$. The case of a non-transitive role r is analogous.

In case of a cyclic exists-edge $(v\,\exists r\,v) \in E_{i,v}$, the induction in principle works just as in the non-transitive case, yielding $z' \in \pi_{(v\,\forall r\,v)}(v)$. However, we cannot analogously deduce that z' therefore also appears in $\pi(v)$, because now we have to make sure that the loop $(v\,\exists r\,v)$ is also reflected by every witness in the model I. Condition 2 of simple description graphs guarantees that r is a transitive role. Moreover, Condition 3 implies a forall-edge $(v\,\forall r\,w) \in E_{i,v}$ starting at v. We know by Condition 4 that a simulation relation exists from $\mathcal{G}(w)$ into $\mathcal{G}(v)$. Altogether, the conditions of simple description graphs imply an exists-edge from w back to v, so that the value restriction imposed by the edge $(v\,\forall r\,w) \in E_{i,v}$ 'contains' the existential restriction imposed by $(v\,\exists r\,v) \in E_{i,v}$. Moreover, on our case Condition 1 implies that there is exactly one forall-edge starting at v. Due to the edge $(v\,\forall r\,w)$ the concept $\ell'_i(v)$ contains a value restriction in which, as a result of the additional conjunction

$$\bigsqcap_{(w\,\exists r\,w')} \exists r.\ell'_{i-1}(w'),$$

an existential restriction for r occurs which can only be satisfied by a model with the following property: from every witness of v it is possible to traverse an arbitrary number (> 1) of r-edges arriving at a witness of v. Hence, the concept $\ell'_i(v)$ reflects the loop $(v\,\exists r\,v) \in E_{i,v}$ in $\text{reach}_i(v)$.

For $w \neq v$, consider a forall-edge $(v\,\forall r\,w) \in E_{i,v}$ w.r.t. a transitive role r and assume that $z \in \pi(v)$ and $(z, z') \in r^I$. Again, the concept $\ell'_i(v)$ contains a value restriction more specific than $\forall r.\ell'_i(v)$. The fact that $z \in \pi(v)$ and $(z, z') \in r^I$ implies that z' is a witness of $\ell'_i(v)$. Hence, by induction hypothesis, z' is also a witness of $\text{reach}_{i-1}(w)$ which means that z' occurs in $\pi_{(v\,\forall r\,w)}(w)$. By construction of π this implies $z' \in \pi(w)$. The non-transitive case is analogous.

The case of a cyclic forall-edge $(v\,\forall r\,v) \in E_{i,v}$ is a little simpler than that of an exists-edge because (i) Condition 2 again guarantees us that r is transitive and (ii) the value restriction $\forall r.\ell'_{i-1}(v)$ automatically, i.e., by the semantics of concept descriptions, restricts all admissible models to those where every r-path from v leads to a witness of $\ell'_i(v)$. Note that this property corresponds to Condition 5 for simple description graphs. ∎

Lemma 36 *For every simple description graph $\mathcal{G} := (V, E, v_0, \ell_V, \ell_E)$ it holds that* $\text{acc}^{|E|}(\mathcal{G}) \equiv \text{conc}(\mathcal{G})$.

Proof. Show ($\sqsubseteq$). By definition of conc it is sufficient to show for an arbitrary $\mathcal{G}$ that $G \sqsubseteq C$ for every $C \in \ell_V(v_0)$. By definition of description graphs, every model I of $\mathcal{G}$ has the property that $x \in C^I$ for every $x \in \pi(x_0)$, where π is the relevant model mapping for I. Hence, every witness of $\mathcal{G}$ by definition is also a witness of C.

Show ($\sqsupseteq$). This is an immediate consequence of Lemma 35 because $\text{reach}_{|E|}(v_0) = \mathcal{G}$ and $\text{conc}(\mathcal{G}) = \ell'_{|E|}(v_0)$. ∎

Hence, we obtain the following theorem.

Theorem 37 *For every simple description graph $\mathcal{G} = (V, E, v_0, \ell_V, \ell_E)$ it holds that* $\text{conc}(\mathcal{G}) \equiv \mathcal{G}$.

The idea of the proof is to show the equivalence $\mathrm{conc}(\mathcal{G}) \equiv \mathcal{G}$ in three steps. Firstly, we show for every $\mathcal{G}$ that a single application of acc preserves equivalence, i.e., $\mathcal{G} \equiv \mathrm{acc}(\mathcal{G})$. This immediately implies $\mathcal{G} \equiv \mathrm{acc}^{|E|}(\mathcal{G})$. Secondly, due to the semantics of description graphs it is also easy to see that every concept description in the root label of $\mathrm{acc}^{|E|}(\mathcal{G})$ subsumes $\mathrm{acc}^{|E|}(\mathcal{G})$. Hence, $\mathrm{acc}^{|E|}(\mathcal{G}) \sqsubseteq \mathrm{conc}(\mathcal{G})$. Thirdly, we can show that every model of $\mathrm{conc}(\mathcal{G})$ is also a model of $\mathrm{acc}^{|E|}(\mathcal{G})$.

Now the necessary means are provided to translate $\mathcal{FLE}^+$-concept descriptions (back and forth) into a representation where the transitivity of roles is made explicit. To define the lcs operation w.r.t. description graphs we first need a complete characterization of subsumption in this representation.

5.4 Characterization of subsumption in $\mathcal{FLE}^+$

In this section the description graphs introduced previously are employed to characterize subsumption. As a preliminary, an auxiliary definition is required to simplify the notation for relations.

Definition 38 *R binary relation over S, T and $s \in S$. Then $R(s) := \{t \in T \mid (s,t) \in R\}$*

The following lemma will show that the subsumption $C \sqsubseteq D$ implies the existence of a simulation relation from $\mathcal{G}_D$ into $\mathcal{G}_C$.

Lemma 39 *Let C be an $\mathcal{FLE}^+$-concept description and $\mathcal{G}_C$ its corresponding concept graph. Let I be a model of C. Then there exists a relation $\rho \subseteq V_C \times \Delta^I$ such that for all vertices $v, w \in V_C$:*

1. *$\rho(\mathrm{root}(\mathcal{G}_C)) = C^I \neq \emptyset$;*
2. *$v_\rho \in (\sqcap \mathrm{label}(v))^I$ for every $v_\rho \in \rho(v)$;*
3. *if $(v\, \exists r\, w) \in E_C$ and $v_\rho \in \rho(v)$ then there exists one $w_\rho \in \rho(w)$ with $(v_\rho, w_\rho) \in r^I$ if $r \in N_R$ and $(v_\rho, w_\rho) \in (r^I)^*$ if $r \in N_R^T$;*
4. *if $(v\, \forall r\, w) \in E_C$ for $r \in N_R$ and $v_\rho \in \rho(v)$ and there exists one $x \in r^I(v_\rho)$ then $x \in \rho(w)$.*
   If $(v\, \forall r\, w) \in E_C$ for $r \in N_R^T$ and $v_\rho \in \rho(v)$ and there exists one $x \in (r^I)^*(v_\rho)$ then $x \in \rho(w)$.

Proof. Since $\mathcal{G}(C)$ is the concept graph of C the conjunction $\sqcap \text{label}(\text{root}(\mathcal{G}(C)))$ of the concepts in the label of the root node is equivalent to C. As I is a model of C we also know that there exists a witness $x \in \Delta^I$ such that $x \in C^I$. Consequently, by including the pair $(\text{root}(\mathcal{G}(C)), x)$ in ρ for every such witness x we have satisfied Condition 1 and Condition 2 for $v = \text{root}(\mathcal{G}(C))$.

$$
\begin{array}{rcl}
v & \stackrel{\rho}{\longrightarrow} & v_\rho \\
\exists r \big\downarrow & & r \big\downarrow \\
v' & \stackrel{\rho}{\longrightarrow} & v'_\rho \\
\mathcal{G}_C & & I
\end{array}
$$

Consider an existential r-edge from $\text{root}(\mathcal{G}(C))$ to a vertex w which has not been traversed yet. If w is a successor w.r.t. a transitive role $r \in N_R^T$ then there exists an existential restriction $C' \in \mathsf{ex}_r(C)$ such that the conjunction $\sqcap \text{label}(w)$ is equivalent to $C' \sqcap \mathsf{val}_r(C) \sqcap \forall r.\mathsf{val}_r(C)$. The fact that x is a witness of C implies the existence of another witness $y \in C'^I$ with $(x, y) \in (r^I)^*$. Moreover, y must also be witness of $\mathsf{val}_r(C)$ and $\forall r.\mathsf{val}_r(C)$ because otherwise x would be no witness of C. Hence, y is a witness of $C' \sqcap \mathsf{val}_r(C) \sqcap \forall r.\mathsf{val}_r(C)$. We may now extend the relation ρ by the pair (w, y) for every such witness y and thereby meet Condition 2 for w and Condition 3 for $\text{root}(\mathcal{G}_C)$ and w. The case of a non-transitive role r is analogous—only that the conjunct $\forall r.\mathsf{val}_r(C)$ is missing and that the pair $(x, y) \in r^I$ instead of the transitive closure of r^I.

Consider a universal r-edge from $\text{root}(\mathcal{G}(C))$ to w w.r.t. a transitive role r which has not yet been traversed. If no witness x of $\text{root}(\mathcal{G}(C))$ has a successor w.r.t. r in then we do not have to assign witnesses to w as permitted by Condition 4. If on the other hand the set of r-successors (w.r.t. the transitive closure of r^I) of x is $\{y_1, \ldots, y_n\}$ then we have already seen in the existential case that every y_i is a witness of $\mathsf{val}_r(C) \sqcap \forall r.\mathsf{val}_r(C)$. Otherwise x would be no witness of $\sqcap \text{label}(\mathcal{G}_C)$. As $\mathsf{val}_r(C) \sqcap \forall r.\mathsf{val}_r(C)$ is equivalent to $\sqcap \text{label}(w)$ we may extend ρ by the pair (w, y_i) for every i. This satisfies Condition 2 for w and Condition 4 for $\text{root}(\mathcal{G}_C)$ and w.

Following the above procedure for existential and universal edges recursively until all edges in $\mathcal{G}_C$ have been traversed we arrive at a relation ρ which satisfies the proposition. ∎

Note: call such a relation ρ *witness-relation*.

$$\begin{array}{ccccc} v & \stackrel{\rightarrow}{\sim} & w & \stackrel{\rho}{\longrightarrow} & w_\rho \\ \forall r \downarrow & & \forall r \downarrow & & r \downarrow \\ v' & \stackrel{\rightarrow}{\sim} & w' & \stackrel{\rho}{\longrightarrow} & w'_\rho \\ \mathcal{G}_D & & \mathcal{G}_C & & I \end{array}$$

Lemma 40 *Let C, D be $\mathcal{FLE}^+$-concept descriptions such that $\mathcal{G}(D) \stackrel{\rightarrow}{\sim} \mathcal{G}(C)$. Let I be a model of C and let ρ be a relation over $V_C \times \Delta^I$ respecting the conditions states in Lemma 39. Then, for all vertices $v \in V_D$ and for all $w \in V_C$ and for all concepts $E \in \text{label}(v)$ it holds that $v \stackrel{\rightarrow}{\sim} w$ implies $w_\rho \in E^I$ for every $w_\rho \in \rho(w)$.*

Proof. Proof by induction on the structure of E.

- $E \in N_C$
  If $v \stackrel{\rightarrow}{\sim} w$ then we know that $\text{label}(v)$ is a subset of $\text{label}(w)$ w.r.t. primitive labels, implying that E also occurs in $\text{label}(w)$. Hence E subsumes $\sqcap \text{label}(w)$. By definition, every $w_\rho \in \rho(w) \subseteq \Delta^I$ is a witness of $\sqcap \text{label}(w)$ and therefore also a witness of E.

- $E = \exists r.(E'_1 \sqcap \cdots \sqcap E'_n)$
  If r is transitive then, by definition of $\mathcal{G}(D)$, there exists an existential r-successor v' of v such that for every $1 \leq i \leq n$ the concept E'_i occurs in $\sqcap \text{label}(v')$. Due to the simulation relation we know that there exists a vertex $w' \in V_C$ with $v' \stackrel{\rightarrow}{\sim} w'$. By induction hypothesis, it holds for w' that $w'_\rho \in E'^I_i$ for every $w'_\rho \in \rho(w')$ and for every i. Moreover, the existential r-edge $(v \, \exists r \, v') \in E_D$ implies that there exists a vertex $w \in V_C$ such that $v \stackrel{\rightarrow}{\sim} w$ and $(wrw') \in E_C$. By definition of the relation ρ it holds that $(w_\rho, w'_\rho) \in r^I$ for every $w_\rho \in \rho(w)$ and $w'_\rho \in \rho(w')$. Consequently, every w_ρ is a witness of E.

- $E = \forall r.(E'_1 \sqcap \cdots \sqcap E'_n)$
  If r is transitive then there exists a universal r-successor v' of v such that every concept E'_i occurs in $\text{label}(v')$. Again, there exist vertices $w, w' \in E_C$ such that w' is a universal r-successor of w and the simulations $v \stackrel{\rightarrow}{\sim} w$ and $v' \stackrel{\rightarrow}{\sim} w'$ hold. Consider the case where $\rho(w) \neq \emptyset$. If $\rho(w')$ is empty then, by Condition 4, no $w_\rho \in \rho(w)$ has an r-successor in I. Consequently, every w_ρ is a trivial witness of E. If $\rho(w')$ is not empty

then we know by Condition 4 that every r-successor x of every w_ρ is in $\rho(w')$. Moreover, we know by induction hypothesis that every such x is a witness of E_i for every i. Hence, every w_ρ is a witness of E.

The argument for non-transitive roles r is analogous in both cases. ■

Our next step is to introduce a class of models for concept descriptions. The idea is to obtain a simple model for a description graph by renaming the labels of its edges.

Definition 41 *Let C be an $\mathcal{FLE}^+$-concept description and $\mathcal{G}_C$ its corresponding concept graph. The* induced model $I(C)$ *of C is defined as follows:*

- $\Delta^{I(C)} := V_C$;
- $A^{I(C)} := \{v \in V_C \mid A \in \text{label}(v)\}$ *for all* $A \in N_C$;
- *For all* $r \in N_R \cup N_R^T$, $(v, w) \in r^I$ *iff* $(v\, Qr\, w) \in E_C$ *for* $Q \in \{\exists, \forall\}$;

To avoid confusion between the sets $\Delta^{I(C)}$ and V_C, every vertex $v \in V_C$ is denoted by v^Δ when referring to the corresponding vertex in $\Delta^{I(C)}$.

By *weak congruence* we denote the fact that a description graph and its induced model are congruent except for the quantor signs at the labels. We still have to show that induced models are in fact models of their respective concept description. The following lemma proves this.

Lemma 42 *Let C be an $\mathcal{FLE}^+$-concept description and $I(C)$ its corresponding induced model. Then,*

1. *$I(C)$ is a model of C;*
2. *The identity* $\text{Id} := \{(v, v) \mid v \in V\}$ *is a witness-relation on* $V_C \times \Delta^{I(C)}$.

Proof. Proof by induction on the number s of steps needed to generate G_C.

- $s = 1$
  Then $\mathcal{G}_C$ consists of only one vertex v with no edges. According to the above definition, v^Δ is a witness of all atomic concepts occurring in C. As $\mathcal{G}_C$ has no edges, we know that C consists of atomic concepts only. Obviously, Id is an appropriate witness-relation.

- $s > 1$

  Consider the case where the algorithm for $\mathcal{G}_C$ adds an existential r-edge (r transitive) in the first step, i.e., $(v\,\exists r\,w)$ is the first edge added to E_C. Hence, there is an existential restriction C' in C which caused the algorithm to add the relevant edge. In this case, two separate tasks remain for the generation of $\mathcal{G}_C$: firstly, the subgraph for $C' \sqcap \mathsf{val}_r(C) \sqcap \forall r.\mathsf{val}_r(C)$ has to be generated starting at w; secondly, the graph for the rest of C, i.e. $C \setminus \exists r.C'$ has to be generated starting at v. It is easy to see that the number of steps needed to accomplish these two tasks is less than s.

  By induction hypothesis, $I(C' \sqcap \mathsf{val}_r(C) \sqcap \forall r.\mathsf{val}_r(C))$ is a model of $C' \sqcap \mathsf{val}_r(C) \sqcap \forall r.\mathsf{val}_r(C)$ and $I(C \setminus C')$ a model of $C \setminus C'$. Moreover, id is a witness-relation between $G_{C' \sqcap \mathsf{val}_r(C) \sqcap \forall r.\mathsf{val}_r(C)}$ and $I(C' \sqcap \mathsf{val}_r(C) \sqcap \forall r.\mathsf{val}_r(C))$ and also between $G_{C \setminus C'}$ and $I(C \setminus C')$.

  In the description graph $\mathcal{G}_C$, an $\exists r$-edge leads from root($\mathcal{G}_C$) to the subgraph for $C' \sqcap \mathsf{val}_r(C) \sqcap \forall r.\mathsf{val}_r(C))$. Consequently, by definition of $I(C)$ (congruence of $\mathcal{G}_C$ and $I(C)$), the submodel $I(C' \sqcap \mathsf{val}_r(C) \sqcap \forall r.\mathsf{val}_r(C)))$ is also connected to the root of $I(C)$ by an r-edge. Similarly, $I(C)$ contains a submodel of $C \setminus C'$ starting at the root node. Hence, $I(C)$ is a model of C. Moreover, as the conjunction of all concepts in label(root($\mathcal{G}_C$)) is equivalent to C, the relation id is a witness-relation between $\mathcal{G}_C$ and $I(C)$. The case of a non-transitive role r is analogous.

  If the algorithm for $\mathcal{G}_C$ adds a universal r-edge $(v\,\forall r\,w)$ in the first step (r transitive) then this is caused by the (only) value restriction on the toplevel of C, i.e., $\mathsf{val}_r(C)$. In this case the generation of the entire description graph $\mathcal{G}_C$ firstly requires the generation of the description graph of $valr(C)$ at vertex w and secondly that of

  $$E := (C \setminus \mathsf{val}_r(C))[C'/C' \sqcap \mathsf{val}_r(C) \sqcap \forall r.\mathsf{val}_r(C) \mid C' \in \mathsf{ex}_r(C)].$$

  We know by induction hypothesis that $I(\mathsf{val}_r(C))$ is a model of $\mathsf{val}_r(C)$ and $I(E)$ one of E. Moreover, in both cases Id serves as witness relation.

  In the description graph $\mathcal{G}_C$, a universal r-edge leads from the root vertex to the subgraph $G_{\mathsf{val}_r(C)}$. This edge is reflected in the model $I(C)$ by an r-edge from the root vertex to the submodel $I(\mathsf{val}_r(C))$. Moreover, the submodel $I(E)$ shares the root vertex with $I(C)$. Hence,

$I(C)$ is a model of $E \sqcap \forall r \mathsf{val}_r(C)$. In case of a transitive role r, the value restriction $\mathsf{val}_r(C)$ as well as the complete subconcept $\forall r.\mathsf{val}_r(C)$ holds for every existential restriction $C' \in \mathsf{ex}_r(C)$. Therefore the conjunction $E \sqcap \forall r \mathsf{val}_r(C)$ is equivalent to C, which makes $I(C)$ a model of C. By induction hypothesis we know that Id is a witness-relation between the relevant subdescriptions and submodels. As the conjunction of all concepts in label(root($\mathcal{G}_C$)) is equivalent to C we also obtain that the identical relation Id is is a witness-relation between $\mathcal{G}_C$

∎

The result on induced models will be of use in the following lemma. We will now show that a subsumption $C \sqsubseteq D$ of concept descriptions implies a certain structural similarity of the respective concept descriptions.

Lemma 43 *Let C, D be $\mathcal{FLE}^+$-concept descriptions with $C \sqsubseteq D$. Let δ be a witness-relation between $\mathcal{G}_D$ and $I(C)$. Let $P_D := (v_0\, Q_0 r_0\, v_1 \ldots v_{n-1}\, Q_{n-1} r_{n-1}\, v_n)$ be a path from $\text{root}(\mathcal{G}_D)$ to v_n in $\mathcal{G}_D$. Then there exists a path $P_C = (w_0\, Q_0 r_0\, w_1 \ldots w_{n-1}\, Q_{n-1} r_{n-1}\, w_n)$ from $\text{root}(\mathcal{G}_C)$ to w_n in $\mathcal{G}_C$ such that for all $0 \leq i \leq n$:*

1. *If for a prefix P'_D of P_D a corresponding path P'_C exists then P_C can be chosen as continuation of P'_C.*
2. $w_i^\Delta \in \delta(v_i)$
3. $\text{atlabel}(v_i) \subseteq \text{atlabel}(w_i)$
4. *For all edges $(v_n\, Qr\, v) \in E_D$ and for $Q \in \{\exists, \forall\}$ there exists an edge $(w_n\, Qr\, w) \in E_C$.*

Proof. Proof by induction on the length n of P_D.

- $n = 0$
  Then $v_0 = v_n = \text{root}(G_D)$. In this case an analogous path P_C in C exists trivially. Due to Lemma 42 we know that Id is a witness-relation between $\mathcal{G}_C$ and $I(C)$. This implies firstly, $\text{root}(\mathcal{G}_C)^\Delta \in \delta(\text{root}(\mathcal{G}_D))$, and secondly, $\text{atlabel}(\text{root}(\mathcal{G}_C)) = \text{atlabel}(\text{root}(\mathcal{G}_C))^\Delta \supseteq \text{atlabel}(\text{root}(\mathcal{G}_D))$. Hence, Conditions 1 and 2 hold.

Consider an existential r-edge $(\text{root}(\mathcal{G}_D)\, \exists r\, v) \in E_D$. Since $I(C)$ is a model of D we know that a corresponding r-edge exists at $\text{root}(I(C))$ because the root of $I(C)$ is a witness of C and thus (by subsumption) also one of D. The weak congruence between $I(C)$ and $\mathcal{G}_C$ consequently implies the existence of either an existential or a universal r-edge starting from $\text{root}(\mathcal{G}_C)$. Assume that a universal r-edge but no existential r-edge is present at $\text{root}(\mathcal{G}_C)$. In this case we can remove the corresponding r-edge in $I(C)$, yielding another model I' of C. However, I' is no model of D any more because the existential r-restriction is not reflected in I', in contradiction to the subsumption $C \sqsubseteq D$.

Consider a universal r-edge $(\text{root}(\mathcal{G}_D)\, \forall r\, v) \in E_D$ representing a non-trivial value restriction. Again, the fact that $I(C)$ is a model of D together with the weak congruence between $I(C)$ and $\mathcal{G}_C$ lets us infer that either a universal or an existential r-edge starts at $\text{root}(\mathcal{G}_C)$. Assume that only an existential r-edge exists but no universal r-edge. We can modify the model $I(C)$ to obtain I' by adding another existential r-edge from $\text{root}(\mathcal{G}_C)$ to a new vertex v' labeled by a new atomic concept A'. The root vertex $\text{root}(I')$ of the modified model is still a witness of C (since C has no value restriction w.r.t. the role r that could be violated) but obviously no witness of D because the newly introduced existential restriction does not reflect the r-value restriction on the toplevel of D. This contradicts $C \sqsubseteq D$.

- $n > 0$
  Let $P_D := (v_0\, Q_0 r_0\, v_1 \dots v_{n-1}\, Q_{n-1} r_{n-1}\, v_n)$. By induction hypothesis there exists a path $P'_C = (w_0\, Q_0 r_0\, w_1 \dots w_{n-2}\, Q_{n-2} r_{n-2}\, w_{n-1})$ in $\mathcal{G}_C$ which respects Conditions 1 to 4 w.r.t. the subpath $(v_0 \dots v_{n-1})$.

  Consider the case $Q_{n-1} = \exists$. Then Condition 4 for the path P'_C ensures that there is a vertex $w_n \in V_C$ with $(w_{n-1}\, \exists r\, w_n) \in E_C$. It remains to be shown that the Conditions 1 to 4 hold for $P_C := (w_0\, Q_0 r_0\, w_1 \dots w_{n-1}\, \exists r\, w_n)$.

  Condition 1 holds due to the inductive construction of P_C which could be built as an extension of any shorter path in $\mathcal{G}_C$ matching the respective prefix of P_D.

  By induction hypothesis we already know that Condition 2 holds for all $1 \leq i \leq n-1$. We now show that an appropriate $w_n \in V_C$ with $w_n^\Delta \in \delta(v_n)$ can always be found. The witness-relation δ relates v_n to a witness x_n in $I(C)$ which has a predecessor x_{n-1} w.r.t. the role r such that x_{n-1} is a witness of v_{n-1}. The weak congruence of $I(C)$ and $\mathcal{G}_C$ and the fact that Id is a witness relation between $\mathcal{G}_C$ and $I(C)$ implies (1) that $x_{n-1} = v_{n-1}^\Delta$ and (2) that $x_n = v_n^\Delta$ for some $\exists r$- or $\forall r$-successor v_n of v_{n-1} in $\mathcal{G}_C$. Analogous to the case for $n = 0$ we can refute the assumption that no existential r-successor v_n can be found, proving Condition 2.

For Condition 3 only the case $i = n$ remains to be shown. Since w_n^Δ is a witness of all concepts in label(v_n) and since atlabel(w_n^Δ) = atlabel(w_n) the fact that Id is a witness-relation between $\mathcal{G}_C$ and $I(C)$ suffices to show that atlabel$(v_n) \subseteq$ atlabel(w_n).

To show Condition 4, consider an arbitrary edge $(v_n\, \exists r\, v) \in E_D$. The fact that w_n^Δ is a witness of every concept description in label(v_n) implies that w_n^Δ has an r-successor w^Δ. Hence, weak congruence and the witness-relation Id between $\mathcal{G}_C$ and $I(C)$ yield an existential or universal r-successor w of w_n in $\mathcal{G}_C$. The assumption that w_n has no existential successor can be shown to contradict the subsumption $C \sqsubseteq D$ in analogy to the case $n = 0$. For a universal edge $(v_n\, \exists r\, v) \in E_D$ we can similarly show that the absence of a corresponding edge $(w_n\, \forall r\, w) \in E_C$ again allows us to modify the model $I(C)$ in such a way that we end up with a model for C which is no model of D, again in contradiction to the subsumption $C \sqsubseteq D$.

Consider the case $Q_{n-1} = \forall$. By induction hypothesis, Condition 4 ensures that w_{n-1} has a universal r-successor w_n in $\mathcal{G}_C$, thus proving the existence of a path P_C in $\mathcal{G}_C$ with the correct labels. The weak congruence of $\mathcal{G}_C$ and $I(C)$ implies an r-successor w_n^Δ of w_{n-1}^Δ related to w_n by the witness-relation Id. To prove Condition 2 for $i = n$, assume that $w_n^\Delta \notin \delta(v_n)$. In this case, another r-successor w^Δ of w_{n-1}^Δ must exist with $w^\Delta \in \delta(v_n)$. Consequently, due to the weak congruence of $\mathcal{G}_C$ and $I(C)$, there is an edge from w_{n-1} to w in $\mathcal{G}_C$ and w is related to w^Δ by the witness-relation Id. Since every vertex in $\mathcal{G}_C$ has at most one $\forall r$-successor we also know that w is connected to w_{n-1} by an $\exists r$-edge. The description graph $\mathcal{G}_C$, however, is defined in such a way that every $\exists r$-successor of w_{n-1} also respects the value restriction, i.e., all concepts in label(w_n). Hence, w_n would also occur in $\delta(v_n)$.

Conditions 3 and 4 can be shown analogously to the case $Q_{n-1} = \exists$

because here the label of the last edge is not relevant here.

∎

The previous lemma is now employed for our original goal—to show that subsumption can be characterized by means of simulation relations on the respective concept descriptions.

Theorem 44 *Let C, D be $\mathcal{FLE}^+$-concept descriptions. Then, $C \sqsubseteq D$ iff $\mathcal{G}_D \rightsquigarrow \mathcal{G}_C$.*

PROOF. ($\Rightarrow$)
If $C \equiv \bot$ then $C \sqsubseteq D$ trivially holds. Otherwise, we have to show for every model I of C that $x \in D^I$ holds for every witness $x \in C^I$. By Lemma 39 we may assume a relation $\rho \subseteq V_C \times \Delta^I$ so that Conditions 1 to 4 are satisfied. This implies that every witness x of C occurs in $\rho(\text{root}(\mathcal{G}_C))$.

Since $\text{root}(\mathcal{G}_D) \rightsquigarrow \text{root}(\mathcal{G}_C)$ we know by Lemma 40 that every $x \in \rho(\text{root}(\mathcal{G}_C))$ is a witness of all concepts in $\text{label}(\text{root}(\mathcal{G}_D))$. Consequently, x is a witness of $\sqcap \text{label}(\text{root}(\mathcal{G}_D))$ which is equivalent to D.

($\Leftarrow$)
If $C \sqsubseteq D$ then we can construct a simulation relation R between $\mathcal{G}_D$ and $\mathcal{G}_C$ in the following way: Initially, let $R := \{(\text{root}(\mathcal{G}_D), \text{root}(\mathcal{G}_C)\}$. Starting from $\text{root}(\mathcal{G}_D)$, we conduct a breadth-first search on $\mathcal{G}_D$. Upon reaching an unvisited vertex v we use Lemma 43 to find a path P_C in $\mathcal{G}_C$ corresponding to the path $(\text{root}(\mathcal{G}_D) \dots v)$ in $\mathcal{G}_D$. The pair consisting of v and the endpoint of P_C is then added to R. For every successor v' of v in $\mathcal{G}_D$ Condition 1 of Lemma 43 allows us to find an extension of the path P_C as corresponding path to $(\text{root}(\mathcal{G}_D) \dots v')$. Applying this strategy exhaustively on $\mathcal{G}_D$, we end up with a simulation relation R. ∎

The reverse direction is only required implicitly throughout this paper. However, the proof of the 'only if'-direction is easily obtained as a consequence of Lemma 30, Lemma 31, and two results shown in the following sections, namely Lemma 26 and Theorem 37.

To illustrate the above result, we return to the example introduced in the previous section.

Example 45 *Recall the example concepts from Example 29. The only difference between C_{ex} and D_{ex} is the atomic concept B in the outermost existential restriction of C_{ex}. Hence, $C_{\text{ex}} \sqsubseteq D_{\text{ex}}$. It is easy to see that $R :=$*

$\{(v_0, v_a), (v_1, v_c), (v_2, v_b)\}$ *is in fact a simulation relation from* $\mathcal{G}_{D_{ex}}$ *into* $\mathcal{G}_{C_{ex}}$. *For all pairs it holds that the label set of the first vertex is a subset of that of the second one and every edge which can be traveled starting from the first vertex can also be traveled from the second one, reaching again a pair in* R. *Note that this property does not hold without the transitive edge* $(v_0\, \exists r\, v_2)$ *in* $\mathcal{G}_{D_{ex}}$.

5.5 Computation of the lcs in $\mathcal{FLE}^+$

With all the information captured in a $\mathcal{FLE}$-concept description made explicit by simple description graphs the next step is to extract the commonalities of the simple description graphs of the input concepts. Similar to other approaches to computing the lcs [1, 4] the graph product is employed to this end. In a description graph $\mathcal{G}$ the *depth* of a vertex v is defined as the distance to the root vertex w.r.t. tree edges of the breadth-first-spanning tree.

Definition 46 (Product of $\mathcal{FLE}^+$-description graphs) *The* product $\mathcal{G}_C \times \mathcal{G}_D$ *of two* $\mathcal{FLE}^+$*-description graphs* $\mathcal{G}_A = (V_A, E_A, v_{0A}, \ell_{V_A}, \ell_{E_A})$ *for* $A \in \{C, D\}$ *is defined by induction on the depth of the* $\mathcal{FLE}^+$*-description graphs. The vertex* (v_{0C}, v_{0D}) *labeled with* $\ell_{V_C}(v_{0C}) \cap \ell_{V_D}(v_{0D})$ *is the root vertex of* $\mathcal{G}_C \times \mathcal{G}_D$. *For each pair* $(v_C, v_D), v_C \in V_C, v_D \in V_D$ *s.t.* v_C *is a* Qr*-successor of* v_{0C} *in* $\mathcal{G}_C$ *and for* v_D *is a* Qr*-successor of* v_{0D} *in* $\mathcal{G}_D$*, we obtain a* Qr*-successor* (v_C, v_D) *of* (v_{0C}, v_{0D}) *in* $\mathcal{G}_C \times \mathcal{G}_D$. *The vertex* (v_C, v_D) *is the root vertex of the inductively defined product of* $\mathcal{G}_C \times \mathcal{G}_D$. *The graph* $\mathcal{H} = \mathcal{G}_C \times \mathcal{G}_D$ *is called the* product graph.

The product graph $\mathcal{G}_C \times \mathcal{G}_D$ is rooted, connected, and directed. Since all vertices in $\mathcal{G}_C$ and $\mathcal{G}_D$ have at most one outgoing forall-edge, every vertex in the product graph has at most one outgoing forall-edge. Thus, product graphs are description graphs.

Example 47 *Let us return to the concept descriptions* C_{ex} *and* D_{ex} *from Example 29. The product of their* $\mathcal{FLE}^+$*-description graphs is displayed in Figure 9. The edges between* v_{b2} *and* v_{c1} *are cross edges.*

Note that by construction of the product graph there trivially exist simulations $Z\colon \mathcal{G}_C \times \mathcal{G}_D \rightleftarrows \mathcal{G}_C$ and between $Z'\colon \mathcal{G}_C \times \mathcal{G}_D \rightleftarrows \mathcal{G}_D$, s.t. for $\{(v_C v_D)\} \in V_{\mathcal{G}_C \times \mathcal{G}_D}$ and $\{v_C\} \in V_C$ holds $Z_C((v_C v_D)) = \{v_C\}$. We call this simulation the *origin simulation to C* denoted Z_{OC}.

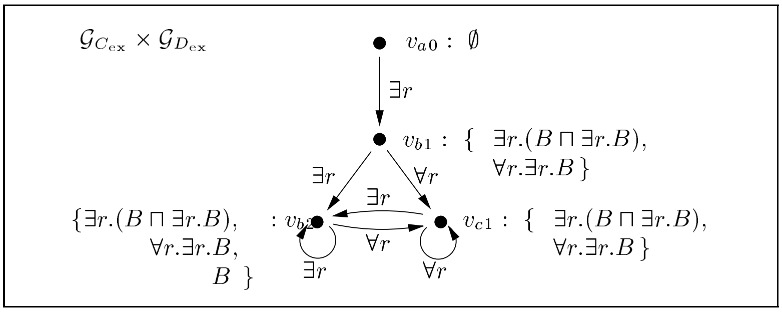

Figure 9: Product Graph for $\mathcal{G}_{C_{\text{ex}}}$ and $\mathcal{G}_{D_{\text{ex}}}$

Once the product graph is obtained, we need to transform this representation into a $\mathcal{FLE}^+$-concept description. In order to apply the conc function introduced in Definition 32, we have to check whether the obtained graph is a simple description graph. Unfortunately, this is not the case since the product graph may contain cross edges (w.r.t. a breadth-first spanning tree).

Cross edges violate the Condition 1 for simple description graphs from Definition 25. Thus, we have to perform the translation of a product graph into a concept description in two steps. First, we have to eliminate cross edges; then, we can use the function conc to read out the concept description. The elimination of cross-edges is performed by an unraveling algorithm that introduces a vertex named with the path by which this vertex is connected to the root vertex and yields a tree with additional back-edges. Thus the obtained graph may still have cycle, but is cross edge free. In order to present the algorithm we need some preliminaries for paths. Let $p = v_1 v_2 \ldots v_n$ be a path, then we denote by $\text{Tail}(p) = v_n$ the last element in p. Let furthermore q be a path, then $p|q$ is the path obtained by the concatenation of p and q. We also need the set $\text{Final}-\text{Path}(\mathcal{G}) := \{(v_1 v_2 \ldots v_n) \in V_{\mathcal{G}}^n \mid (v_i\, Qr\, v_{i+1}) \in E_{\mathcal{G}}, x_j \neq x_i \text{ for } j \neq i\}$. The unraveling is performed according the unravel-algorithm depicted in Figure 10.

The function first eliminates all existential forward edges in the graph and then eliminates all cross edges recursively by calling the function eliminate with the root vertex as start vertex. This function in turn traverses the graph starting from the vertex v^* and eliminates every cross edge $(v^*\, Qr\, w)$ by removing it from the set of edges, traversing and eliminating the cross edges from the reachability graph of w, making a copy of this sub-graph and introducing a new Qr-successor for v^* as the root vertex of this copy. A product graph can now be transformed into a cross edge-free graph by

$$\begin{array}{l}
\text{unravel}(\mathcal{G} = (V, E, v_0, \ell_V, \ell_E)) \\
\mathcal{G}_1 := \text{remove transitivity edges}(\mathcal{G}) \\
\mathcal{G}_2 := \text{eliminate}(\mathcal{G}_1, v_0, \emptyset) \\
\mathcal{G}_3 := \text{For } r \in N_R^T \text{ do transitive-closure}(\mathcal{G}_2, \exists r) \\
\text{return } \mathcal{G}_3
\end{array}$$

$$\begin{aligned}
&\text{eliminate}(\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G}, v_{0_\mathcal{G}}, \ell_{V_\mathcal{G}}, \ell_{E_\mathcal{G}})) \\
V' &:= \text{Final}-\text{Path}(\mathcal{G}) \\
E' &:= \{(p\ Qr\ p|Qrv) \in V' \times V' \mid (\text{Tail}(p)\ Qr\ v) \in E_{\mathcal{G}^*}\} \cup \\
&\quad\ \{((p|Q_1 r_1 v|q)\ Qr\ (p|Q_1 r_1 v)) \in V' \times V' \mid (\text{Tail}(q)\ Qr\ v) \in E_\mathcal{G}\} \\
\ell'_{V_\mathcal{G}}(p) &:= \ell_{V_\mathcal{G}}(\text{Tail}(p)) \\
\ell'_{E_\mathcal{G}}(pQrq) &:= \ell_{E_\mathcal{G}}(\text{Tail}(p) Qr \text{Tail}(q)) \\
&\text{return } \mathcal{G}
\end{aligned}$$

Figure 10: Unravel Function for Description Graphs

applying the unravel function. The graph obtained by the unravel function is equivalent to the original one.

Lemma 48 *Let C, D be $\mathcal{FLE}^+$-concept descriptions and $\mathcal{G}_C, \mathcal{G}_D$ their corresponding $\mathcal{FLE}^+$-description graphs. Then,*

1. Z: $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D) \rightleftarrows \mathcal{G}_C \times \mathcal{G}_D$ *and*
   Z': $\mathcal{G}_C \times \mathcal{G}_D \rightleftarrows \text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$, *and*
2. Z'': $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D) \rightleftarrows \mathcal{G}_C$.

Proof. *Proof of 1:* We prove the claim in two steps, by advising two relations between $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$ and $\mathcal{G}_C \times \mathcal{G}_D$ and then show that these relations are simulations. We use $\mathcal{G}_u$ as short-hand notation for $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$ and $\mathcal{G}_x$ as short-hand notation for $\mathcal{G}_C \times \mathcal{G}_D$.

- The relation from $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$ to $\mathcal{G}_C \times \mathcal{G}_D$ is defined as: $Z(p) = \text{Tail}(p)$. We have to show now that this relation fulfills the definition of a simulation. Since $v_{0_{\mathcal{G}_x}} \in \text{Final}-\text{Path}(\mathcal{G}_x)$ the roots are mapped onto each other $Z(v_{0_{\mathcal{G}_x}}) = \text{Tail}(v_{0_{\mathcal{G}_x}}) = v_{0_{\mathcal{G}_x}}$. The label set of each vertex $p \in \text{Final}-\text{Path}$ in the unraveled graph is defined by $\ell'_{V_{\mathcal{G}_u}}(p) := \ell_{V_{\mathcal{G}_x}}(\text{Tail}(p))$, thus it fulfills $\ell'_{V_{\mathcal{G}_u}}(p) \subseteq \ell_{V_{\mathcal{G}_x}}(Z(p))$. It remains to be shown that if $(p\ Qr\ p|Qrq) \in E_{\mathcal{G}_u}$ and $||q|| = 1$, then $\exists v' \in V_{\mathcal{G}_x}$: $(v\,Qr\,v') \in E_{\mathcal{G}_x}$ and $Z(q) = v$. If $(p\ Qr\ p|Qrq) \in E_{\mathcal{G}_u}$ then by definition of $\mathcal{G}_u = \text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$: $(\text{Tail}(p)\ Qr\ q) \in E_{\mathcal{G}_x}$, since for $||q|| = 1$ holds that $Z(q) = q$.

- The relation from $\mathcal{G}_C \times \mathcal{G}_D$ to $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$ is defined as: $Z'(v) = \{p \mid \text{Tail}(p) = v\}$. We have to show again that all properties from the definition of simulations hold. As above, the roots are mapped onto each other, since $v_{0_{\mathcal{G}_x}} \in \text{Tail}((v_{0_{\mathcal{G}_x}})) = Z'(v_{0_{\mathcal{G}_x}})$. The definition of the function unravel implies that the label of the vertices trivially fulfill the condition: $\ell_{V_{\mathcal{G}_x}}(Z'(v)) := \ell_{V_{\mathcal{G}_x}}(\text{Tail}(\{p \mid \text{Tail}(p) = v\})) = \ell_{V_{\mathcal{G}_x}}(v)$. Thus it remains to be shown that if $(vQrw) \in E_{\mathcal{G}_x}$ then $\forall p \in Z'(v) : \exists q \in Z'(w) : (p\ Qr\ p|Qrq) \in E_{\mathcal{G}_u}$. From $p \in Z'(v)$ follows by the definition of Z' that $\text{Tail}(p) = v$. According to whether w appears more that once in p we have to make a case distinction.

  (1) If $(p = p_1|Qrw|p_2)$ for $\exists p_1, p_2 : ||p_i|| \geq 1$ with $i \in \{1, 2\}$. Since $\text{Tail}(p) = \text{Tail}(p_2) = v$, we have $(\text{Tail}(p)\,Qr\,w) \in E_{\mathcal{G}_x}$, thus by definition of $E_{\mathcal{G}_u}$ there must exist $(p_1|Qrw|p_2\ Qr\ p_1|Qrw) \in E_{\mathcal{G}_u}$. Since $(p_1|Qrw) \in \text{Tail}(w)$ we have found the required successor.

  (2) If $p = p_1\ Q'r'v$, then by definition of $\text{Final}-\text{Path}(\mathcal{G}_x)$: $(p|Qrw) \in \text{Final}-\text{Path}(\mathcal{G}_x)$ and $\text{Tail}(p) = v$ and thus $(\text{Tail}(p)\ Qr\ w) \in E_{\mathcal{G}_x}$ and by definition of $E_{\mathcal{G}_u}$ there must also exist $(p\ Qr\ p|Qrw) \in E_{\mathcal{G}_u}$.

Thus both relations Z and Z' are simulations.
Proof of 2: In the Lemma 48 claim (2) is an immediate consequence of (1). Since Z_1: $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D) \rightleftarrows \mathcal{G}_C \times \mathcal{G}_D$ and there always exists a simulation Z_2: $\mathcal{G}_C \times \mathcal{G}_D \rightleftarrows \mathcal{G}_C$, there always exists a simulation $Z_2 \circ Z_1$: $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D) \rightleftarrows \mathcal{G}_C$. ∎

Lemma 49 *Let C, D be $\mathcal{FLE}^+$-concept descriptions and $\mathcal{G}_C, \mathcal{G}_D$ their corresponding $\mathcal{FLE}^+$-description graphs, then* $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$ *is a simple description graph.*

PROOF. Since the root vertex of $\mathcal{G}_C \times \mathcal{G}_D$ is also the root vertex of $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$ and the unravel function yields connected and directed graphs, $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$ is a description graph. To prove the claim we show that all properties from Definition 25 hold. Again we use $\mathcal{G}_u$ as short-hand notation for $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$ and $\mathcal{G}_x$ as short-hand notation for $\mathcal{G}_C \times \mathcal{G}_D$.

- *Proof of Property 1* Since neither $\mathcal{G}_C$ nor $\mathcal{G}_D$ have forall-forward edges, $\mathcal{G}_x$ cannot have forall-forward edges by construction w.r.t. a breadth first search tree. Neither the function unravel nor the function eliminate introduce new forall-forward edges.

  Since the function eliminate traverses and unravels the whole graph the graph $\mathcal{G}_u$ is cross edge-free. Since the exists-forward edges in $\mathcal{G}_x$ are removed in the first step of the unravel function, no forward edges are created by the eliminate function, but only edges to "fresh" vertices or back loops, and since the exists-forward edges introduced by the last step in the unravel function are the only forward edges in $\mathcal{G}_u$ and these edges connect vertices connected by a path of exists-tree edges w.r.t. one transitive role. Hence Property 6 holds for $\mathcal{G}_u$.

- *Proof of Property 2:* We show by contradiction that Property 2 holds. Assume there exists a cycle $\{(v_1\, Q_1 r_1\, v_2)(v_2\, Q_2 r_2\, v_3)\dots(v_n\, Q_n r_n\, v_1)\} \in E_{\mathcal{G}_u}$, where $n \geq 1$ and $v_i \neq v_j$ for $i \neq j$ and either $r_i \neq r_j$ or $r \notin N_R^T$.

  The edges are introduced in the function unravel by the last step and the call of the function eliminate. The last step in function unravel only introduces forward-edges and thus no cycles. Since Lemma 48 holds, we know that the cycle $\{(v_1\, Q_1 r_1\, v_2)(v_2\, Q_2 r_2\, v_3)\dots(v_n\, Q_n r_n\, v_1)\} \in \mathcal{G}_u$ can be simulated in $\mathcal{G}_x$. Thus $\mathcal{G}_x$ contains a cycle for the same sequence of roles $Q_1 r_1 Q_2 r_2 \dots Q_n r_n$. Since there exist simulation relations $\mathcal{G}_x \rightleftarrows \mathcal{G}_C$ and $\mathcal{G}_x \rightleftarrows \mathcal{G}_D$ corresponding cycles must exist in $\mathcal{G}_C$ and in $\mathcal{G}_D$. Since $\mathcal{G}_C$ and $\mathcal{G}_D$ are simple description graph they fulfill Property 2 and thus our initial assumption is false.

- *Proof of Property 3:* Consider a cycle $(p_1\, Qr\, p_2 \dots p_n\, Qr\, p_1)$ in $\mathcal{G}_u$ with pairwise distinct vertices. The above Condition allows us to restrict our attention to cycles over only one transitive role r. From Lemma 48 follows for $\mathcal{G}_u$ that there exists a simulation to $\mathcal{G}_x$ and vice versa. Thus it suffices to show the claim for $\mathcal{G}_x$. Since there is a simulation $Z\colon \mathcal{G}_u \rightleftarrows \mathcal{G}_x$, there is also a cycle $((w_{0C} w_{0D})\, Qr\, (w_{1C} w_{1D}) \dots (w_{nC} w_{nD})\, Qr\, (w_{0C} w_{0D}))$ in $\mathcal{G}_x$ with pairwise distinct vertices. From the definition of product graphs follows that there must exist the cycles $((w_{0E}\, Qr\, w_{1E}) \dots (w_{nE}\, Qr\, w_{0E}))$ in $\mathcal{G}_E$ with pairwise distinct vertices w_{iE} for all $E \in \{C, D\}$. Since $\mathcal{G}_C$ and $\mathcal{G}_D$ are simple description graphs, Lemma 31 guarantees that Property 3 holds for $\mathcal{G}_C$ and $\mathcal{G}_D$. Consequently there must exist a forall-successor of w_{0C} in $\mathcal{G}_C$ and there must exist a forall-successor of

w_{0D} in $\mathcal{G}_D$. Since $(w_{0C}w_{0D})$ is a vertex in $\mathcal{G}_x$ the definition of product graphs requires that there exists a forall-successor of $(w_{0C}w_{0D})$. The simulation from $\mathcal{G}_x$ to $\mathcal{G}_u$ implies that there also exists a forall-successor of $(v_{0C}v_{0D})$ in $\mathcal{G}_u$.

- *Proof of Property 4:* Since Lemma 48 holds it suffices to show the claim for $\mathcal{G}_x$. Assume $\{((u_Cu_D)\,\forall r\,(v_Cv_D)), ((u_Cu_D)\,\exists r\,(w_Cw_D))\} \subseteq E_{\mathcal{G}_x}$. The definition of the product graph $\mathcal{G}_x$ implies that there exist $\{(u_C\,\forall r\,v_C), (u_C\,\exists r\,w_C)\} \subseteq E_{\mathcal{G}_C}$ and $\{(u_D\,\forall r\,v_D), (u_D\,\exists r\,w_D)\} \subseteq E_{\mathcal{G}_D}$. According Lemma 31 $\mathcal{G}_C$ and $\mathcal{G}_D$ are simple description graphs. Thus $\mathcal{G}_C$ and $\mathcal{G}_D$ fulfill Property 4 and there exist simulations Z_E s.t. $\mathcal{G}_E(v_E) \rightleftarrows \mathcal{G}_E(w_E)$ for all $E \in \{C, D\}$. Thus $Z_C \circ Z_{OC} \colon (\mathcal{G}_x)((v_Cv_D)) \rightleftarrows \mathcal{G}_C(w_C)$ and $Z_D \circ Z_{OD} \colon (\mathcal{G}_x)((v_Cv_D)) \rightleftarrows \mathcal{G}_D(w_D)$. Thus for every vertex (vv') in $(\mathcal{G}_x)((v_Cv_D))$ holds: If (vv') has a Qr'-successor there must exist a Qr'-successor of $Z_C(v)$ in $\mathcal{G}_C$ and a Qr'-successor of $Z_D(v')$ in $\mathcal{G}_D$. By definition of product graphs there must be a vertex $(Z_C(v)Z_D(v'))$ in $\mathcal{G}_x$. Consider the labels in $\mathcal{G}_x$: If $\big(Z_C \circ Z_{OC}(vv')\; Z_D \circ Z_{OD}(vv')\big) = (ww')$ then, by the definition of simulation holds $\ell_{\mathcal{G}_x}(vv') \subseteq \ell_{\mathcal{G}_x}(ww')$. Since $(Z_C(v_C)Z_D(v_D)) = (w_Cw_D)$ we obtain a simulation from (v_Cv_D) to (w_Cw_D).

  If $r \in N_R^T$ we have to show that there also exists a vertex $(w'_Cw'_D) \in V_{\mathcal{G}_x}$ such that $((w_Cw_D)\,\forall r\,(w'_Cw'_D)) \in E_{\mathcal{G}_x}$ and $(\mathcal{G}_x)((v_Cv_D)) \rightleftarrows (\mathcal{G}_x)((w'_Cw'_D))$. As above we know from Lemma 31 that $\mathcal{G}_C$ and $\mathcal{G}_D$ fulfill Property 4 and thus there exists a vertex $(w'_Cw'_D)$ and the simulations Z'_E s.t. $\mathcal{G}_E(v_E) \rightleftarrows \mathcal{G}_E(w'_E)$ for all $E \in \{C, D\}$. Thus there is again a composition of simulations $Z'_C \circ Z_{OC} \colon (\mathcal{G}_x)((v_Cv_D)) \rightleftarrows \mathcal{G}_C(w'_C)$ and $Z'_D \circ Z_{OD} \colon (\mathcal{G}_x)((v_Cv_D)) \rightleftarrows \mathcal{G}_D(w'_D)$. Hence for every vertex (vv') in $(\mathcal{G}_x)((v_Cv_D))$ with a Qr'-successor there must exist a Qr'-successor of $Z'_C(v)$ in $\mathcal{G}_C$ and a Qr'-successor of $Z'_D(v')$ in $\mathcal{G}_D$. Thus we can, as above, conclude that for every vertex $(Z'_C(v)Z'_D(v'))$ in $\mathcal{G}_x$, where for $\big(Z'_C \circ Z_{OC}(vv')\; Z'_D \circ Z_{OD}(vv')\big) = (ww')$ by the definition of simulation holds that $\ell_{\mathcal{G}_x}(vv') \subseteq \ell_{\mathcal{G}_x}(ww')$. Since $(Z'_C(v_C)Z'_D(v_D)) = (w'_Cw'_D)$ we obtain a simulation from (v_Cv_D) to $(w'_Cw'_D)$.

- *Proof of Property 5:* Proof is analogous to the Proof of Property 4. Again, since Lemma 48 holds it suffices to show the claim for $\mathcal{G}_x$. Assume $((u_Cu_D)\,\forall r\,(v_Cv_D)) \in E_{\mathcal{G}_x}$ for $r \in N_R^T$. The definition of $\mathcal{G}_x$ implies that there exist $(u_C\,\forall r\,v_C) \in E_{\mathcal{G}_C}$ and $(u_D\,\forall r\,v_D) \in E_{\mathcal{G}_D}$. According Lemma 31 $\mathcal{G}_C$ and $\mathcal{G}_D$ are simple description graphs. Thus $\mathcal{G}_C$ and $\mathcal{G}_D$ also fulfill Property 5 and thus there exist the vertices $(v_E \forall r v'_E)$ and simulations Z_E s.t. $\mathcal{G}_E(v_E) \rightleftarrows \mathcal{G}_E(v'_E)$ for all $E \in \{C, D\}$. Thus $Z_C \circ Z_{OC} \colon (\mathcal{G}_x)((u_C u_D)) \rightleftarrows \mathcal{G}_C(v'_C)$ and $Z_D \circ Z_{OD} \colon (\mathcal{G}_x)((u_C u_D)) \rightleftarrows \mathcal{G}_D(v'_D)$. Thus for every vertex (uu') in $(\mathcal{G}_x)((u_C u_D))$ holds: If (uu') has a Qr'-successor there must exist a Qr'-successor of $Z_C(u)$ in $\mathcal{G}_C$ and a Qr'-successor of $Z_D(u')$ in $\mathcal{G}_D$. By definition of product graphs then there must be a vertex $(Z_C(u)Z_D(u'))$ in $\mathcal{G}_x$. Consequently there is the vertex $(Z_C(u_C)Z_D(u_D)) = (v'_C v'_D)$ in $\mathcal{G}_x$. Consider the labels in $\mathcal{G}_x$: If $\big(Z_C \circ Z_{OC}(uu')\ Z_D \circ Z_{OD}(uu')\big) = (vv')$, then by the definition of simulation holds $\ell_{\mathcal{G}_x}(uu') \subseteq \ell_{\mathcal{G}_x}(vv')$. Since $(Z_C(u_C)Z_D(u_D)) = (v'_C v'_D)$ we obtain a simulation from $(u_C u_D)$ to $(v'_C v'_D)$.

- *Proof of Property 6 6:* Consider the vertex $(v_C v_D) \in V_{\mathcal{G}_u}$ with $B \in \ell_{\mathcal{G}_u}(v_C v_D)$. Once again, from Lemma 48 follows for unravel$(\mathcal{G}_x)$ that there exists a simulation to $\mathcal{G}_x$ and vice versa. Thus there exists $(v'_C v'_D) \in V_{\mathcal{G}_x}$ where $B \in \ell_{\mathcal{G}_x}(v'_C v'_D)$. It follows from the definition of product graphs that, if $B \in \ell_{\mathcal{G}_x}(v'_C v'_D)$ then $B \in N_C$. Consequently the simulation from $\mathcal{G}_B$ to $(\mathcal{G}_u(v_C v_D)$ trivially exists.

Since all properties from Definition 25 hold, $\mathcal{G}_u$ is a simple description graph. ∎

Since the graph obtained by the function unravel is a simple description graph, Theorem 37 is applicable and the concept description corresponding to the unraveled graph can be obtained by the conc function to read a $\mathcal{FLE}^+$-concept description from the simple description graph. We are now ready to prove the main theorem of this paper.

Theorem 50 *Let C, D be $\mathcal{FLE}^+$-concept descriptions and $\mathcal{G}_C, \mathcal{G}_D$ their corresponding simple description graphs, then* $\mathrm{conc}(\mathrm{unravel}(\mathcal{G}_C \times \mathcal{G}_D)) \equiv \mathsf{lcs}(C, D)$.

Proof. Let $L = \mathrm{conc}(\mathrm{unravel}(\mathcal{G}_C \times \mathcal{G}_D))$. We have to show that (1) $C \sqsubseteq L$ and $D \sqsubseteq L$ and (2) if there exist another $\mathcal{FLE}^+$-concept E with $E \sqsubseteq L$, $C \sqsubseteq E$, and $D \sqsubseteq E$ then $L \sqsubseteq E$.

Proof of (1): It is sufficient to show $C \sqsubseteq L$. Lemma 48 implies that there exists a simulation $Z \colon \mathrm{unravel}(\mathcal{G}_C \times \mathcal{G}_D) \rightleftarrows \mathcal{G}_C$. Applying Lemma 49 to the unraveled graph and by the definition of $\mathcal{G}_C$ we know that unravel$(\mathcal{G}_C \times \mathcal{G}_D)$ and $\mathcal{G}_C$ are both simple description graphs. Thus Lemma 26 implies that

$\mathcal{G}_C \sqsubseteq \text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$ since there is a simulation. From Theorem 37 it follows that $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D) \equiv \text{conc}(\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D))$. Since $\mathcal{G}_C$ is a simple description graph, Lemma 30 and Lemma 31 can be applied and we can conclude that $\mathcal{G}_C \equiv C \sqsubseteq \text{conc}(\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)) \equiv \text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$.

Proof of (2): By contradiction: assume $C \sqsubseteq E$, $D \sqsubseteq E$, $E \sqsubseteq L$ and $L \not\sqsubseteq E$. Let $\mathcal{G}_A := (V_A, E_A, v_0^A, \ell_V^A, \ell_E^A)$ where $A \in \{C, D, E, L\}$. From $C \sqsubseteq E$ and $D \sqsubseteq E$ follows by Theorem 44 that there exist simulations $Z_C : \mathcal{G}_E \rightleftarrows \mathcal{G}_C$ and $Z_D : \mathcal{G}_E \rightleftarrows \mathcal{G}_D$. Thus it holds by definition of simulations: $\forall v \in V_E$:

- $\forall v_F \in V_F$: If $v_F \in Z_F(v)$ then $\ell_V^E(v) \subseteq \ell_V^F(v_F)$, and
- $\forall (v\, Qr\, w) \in E_E$ there exist $v_F, w_F \in V_F$ s.t. $\{v_F\} \in Z_F(v), \{w_F\} \in Z_F(w)$ and $(v_F\, Qr\, w_F) \in E_F$,

where $F \in \{C, D\}$. From the existence of both simulation relations and from the Definition of product graphs follows that for all $v \in V_E$:

- If $v_C \in Z_C(v)$ and $v_D \in Z_D(v)$ for $v_C \in V_C, (v_C\, Qr\, w_C) \in E_C$ and for $v_D \in V_D, (v_D\, Qr\, w_D) \in E_D$ then there exist the vertices $\{(v_C, v_D), (w_C, w_D)\} \in V_{\mathcal{G}_C \times \mathcal{G}_D}$ and $((v_C, v_D)\, Qr\, (w_C, w_D)) \in E_{\mathcal{G}_C \times \mathcal{G}_D}$.
- Since $\ell_V^E(v) \subseteq \ell_V^C(v_C) \cap \ell_V^D(v_D) = \ell_V^{\mathcal{G}_C \times \mathcal{G}_D}((v_C, v_D))$

Thus there exists a simulation relation $Z_L : \mathcal{G}_E \rightleftarrows \mathcal{G}_C \times \mathcal{G}_D$, where $Z_L(v) = \{(v'v'') \in V_{\mathcal{G}_C \times \mathcal{G}_D} \mid v' \in Z_C(v), v'' \in Z_D(v)\}$. By Lemma 48 then there also must exist a simulation $Z'_L : \mathcal{G}_E \rightleftarrows \text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$. Since $\mathcal{G}_E$ and $\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$ are simple description graphs, Lemma 26 implies $\mathcal{G}_E \sqsubseteq \text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$. From this we obtain by means of Lemma 30, Lemma 31, and Lemma 49, that $\mathcal{G}_E \equiv E \sqsubseteq \text{conc}(\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)) \equiv \text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)$. This is a contradiction to our initial assumption that $L \not\sqsubseteq E$. Thus we can conclude that $\text{conc}(\text{unravel}(\mathcal{G}_C \times \mathcal{G}_D)) \equiv \mathsf{lcs}(C, D)$. ∎

In case the n-ary lcs is to be computed from a set of concepts, the product of all corresponding simple description graphs should be computed first and then the unravel and the conc function should be applied only once.

6 Conclusion and Outlook

We have shown how the existing lcs algorithms for the DLs $\mathcal{EL}$ and $\mathcal{FL}_0$ can be extended to transitive roles with comparatively little effort. In case of

$\mathcal{EL}^+$ concept descriptions, the effect of transitive roles could simply be made explicit by adding a number of certain existential restrictions to the concept. For $\mathcal{FL}_0^+$, the representation of concept descriptions by formal languages could be extended by means of an operator for the transitive closure of formal languages.

For the DL $\mathcal{FLE}^+$ we have introduced a sound and complete algorithm for the effective computation of the lcs. In particular, the lcs of a finite set of $\mathcal{FLE}^+$-concept descriptions always exists and is uniquely determined up to equivalence. As a key utility for the lcs computation we have proposed description graphs as a finite representation of $\mathcal{FLE}^+$-concept descriptions in which all restrictions additionally imposed by transitive roles are made explicit. On this basis the lcs could be defined by means of the graph product of the description graphs of the input concepts.

It is easy to see that the lcs algorithm can be optimized in several ways to produce smaller output concept descriptions. Firstly, the blocking conditions used to generate description graphs out of concept descriptions so far only allow for blocking w.r.t. ancestors. This might be replaced by a more general blocking strategy capable of blocking between arbitrary vertices. Secondly, it seems expedient to reduce redundancies possibly produced by the function conc. In particular, it is not always necessary to apply the acc-function once for every edge in the description graph. A thorough investigation of the computational complexity of the lcs computation in $\mathcal{FLE}^+$ remains future work. Nevertheless, already for then non-transitive language $\mathcal{FLE}$ it is known that the lcs may be exponentially large in the input size.

Acknowledgments

We would like to express our thanks to Ulrike Sattler and Carsten Lutz for contributing many helpful ideas to the results presented here.

References

[1] F. Baader. Least common subsumers, most specific concepts, and role-value-maps in a description logic with existential restrictions and terminological cycles. LTCS-Report LTCS-02-07, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2002.

[2] F. Baader and R. Küsters. Unification in a description logic with inconsistency and transitive closure of roles. In I. Horrocks and S. Tessaris, editors, *Proceedings of the 2002 International Workshop on Description Logics*, Toulouse, France, 2002.

[3] F. Baader, R. Küsters, and R. Molitor. Computing least common subsumers in description logics with existential restrictions. LTCS-Report LTCS-98-09, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 1998.

[4] F. Baader, R. Küsters, and R. Molitor. Computing least common subsumer in description logics with existential restrictions. In T. Dean, editor, *Proc. of IJCAI-99*, pages 96–101, Stockholm, Sweden, 1999. Morgan Kaufmann.

[5] F. Baader and P. Narendran. Unification of concept terms in description logics. In H. Prade, editor, *ECAI-98*, pages 331–335. John Wiley & Sons Ltd, 1998.

[6] Sean Bechhofer, Ian Horrocks, Carole Goble, and Robert Stevens. OilEd: a Reason-able Ontology Editor for the Semantic Web. In *KI-01*.

[7] A. Borgida and D. L. McGuinness. Asking queries about frames. In Luigia C. Aiello, John Doyle, and Stuart C. Shapiro, editors, *Proc. of KR-96*, pages 340–349, Cambridge, MA, 1996. Morgan Kaufmann.

[8] S. Brandt and A.-Y. Turhan. Using non-standard inferences in description logics — what does it buy me? In *Proceedings of the KI-2001 Workshop on Applications of Description Logics (KIDLWS'01)*, number 44 in CEUR-WS, Vienna, Austria, September 2001. RWTH Aachen.

[9] Martin Buchheit, Francesco M. Donini, and Andrea Schaerf. Decidable reasoning in terminological knowledge representation systems. *Journal of Artificial Intelligence Research*, 1:109–138, 1993.

[10] W. W. Cohen, A. Borgida, and H. Hirsh. Computing least common subsumers in description logics. In W. Swartout, editor, *Proc. of AAAI-92*, pages 754–760, San Jose, CA, 1992. AAAI Press/The MIT Press.

[11] F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in description logics. In G. Brewka, editor, *Foundation of Knowledge Representation*. CSLI Publication, Cambridge University Press, 1996.

[12] V. Haarslev and R. Möller. Expressive abox reasoning with number restrictions, role hierarchies, and transitively closed roles. In *Proc. of KR-00*, 2000.

[13] I. Horrocks and U. Sattler. A description logic with transitive and inverse roles and role hierarchies. *J. of Logic and Computation*, 9(3):385–410, 1999.

[14] Deborah L. McGuinness and Peter Patel-Schneider. Usability issues in knowledge representation systems. In *Proc. of AAAI-98*.