RWTH
LTCS–Report

Aachen University of Technology
Research group for
Theoretical Computer Science

Computing Least Common Subsumers in $\mathcal{ALEN}$

Ralf Küsters
Institute for Computer Science
and Applied Mathematics
Christian-Albrechts-University of Kiel
kuesters@ti.informatik.uni-kiel.de

Ralf Molitor*
Swiss Life
IT Research and Development Group
8022 Zürich, Switzerland
ralf.molitor@swisslife.ch

LTCS-Report 00-07

(*) This research was carried out while the second author was still at the LuFG Theoretical Computer Science, RWTH Aachen.

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

Ahornstr. 55
52074 Aachen
Germany

Abstract

Computing the least common subsumer (lcs) in description logics is an inference task first introduced for sublanguages of CLASSIC. Roughly speaking, the lcs of a set of concept descriptions is the most specific concept description that subsumes all of the input descriptions. As such, the lcs allows to extract the commonalities from given concept descriptions, a task essential for several applications like, e.g., inductive learning, information retrieval, or the bottom-up construction of KR-knowledge bases.

Previous work on the lcs has concentrated on description logics that either allow for number restrictions or for existential restrictions. Many applications, however, require to combine these constructors. In this work, we present an lcs algorithm for the description logic $\mathcal{ALEN}$, which allows for both constructors (as well as concept conjunction, primitive negation, and value restrictions). The proof of correctness of our lcs algorithm is based on an appropriate structural characterization of subsumption in $\mathcal{ALEN}$ also introduced in this paper.

1 Introduction

Computing the least common subsumer (lcs) in description logics (DLs) is an inference task first introduced by Cohen, Borgida, and Hirsh [4] for sublanguages of Classic. Since then, it has found several applications: as a key operation in inductive learning algorithms [5], as a means to measure the similarity of concepts for information retrieval [11], and as an operation to support the bottom-up construction of DL-knowledge bases [1, 2]. Roughly speaking, the lcs of a set of concepts is the most specific concept description (among a possibly infinite number of concept descriptions) that subsumes all of the input descriptions, and as such allows to extract the commonalities from given concept descriptions, a task essential for all the mentioned applications.

The first lcs algorithms proposed in the literature were applicable to sublanguages of Classic, more precisely, DLs that in particular allow for number restrictions [4, 5]. More recently, motivated by the bottom-up construction of knowledge bases in a chemical engineering application [12, 13], the lcs has been investigated for the DL $\mathcal{ALE}$ [2], which allows for existential restrictions instead of number restrictions. Although first empirical results are encouraging [3], they also show that this application asks for a more expressive DL, one that allows to combine number restrictions and existential restrictions. Such a logic can, for example, be used to describe a reactor with cooling jacket and exactly two inlet valves by $\mathsf{Reactor} \sqcap \exists \mathsf{is\text{-}coupled\text{-}to.Cooling\text{-}Jacket} \sqcap (=2\ \mathsf{has\text{-}inlet}) \sqcap \forall \mathsf{has\text{-}inlet.Valve}$.

In this work, we propose an algorithm for computing the lcs of $\mathcal{ALEN}$-concept descriptions. The DL $\mathcal{ALEN}$ allows for conjunction, a restricted form of negation, value restrictions, existential restrictions, and number restrictions. Similar to previous approaches [4, 1, 2], our lcs algorithm builds on a structural characterization of subsumption.

Typically, such a characterization works in two steps. First, concept descriptions are turned into a structural normal form, which makes all facts implicitly represented in the description explicit. Second, the subsumer and the subsumee, given in structural normal form, are compared syntactically. A sound and complete characterization then ensures that the structural normal form indeed contains all implied facts.

Now, given that the structural normal form of concept descriptions can be computed effectively, the lcs of concept descriptions can be obtained by first computing their structural normal forms and then extracting the "common facts" present in these normal forms.

For $\mathcal{ALEN}$, however, computing the structural normal form already requires to (inductively) compute the lcs (see our running example in Section 3). Consequently, the proof of correctness of the lcs

Construct name	Syntax	Semantics
top-concept	$\top$	$\Delta_\mathcal{I}$
bottom-concept	$\bot$	$\emptyset$
concept name $P \in N_C$	P	$P^\mathcal{I} \subseteq \Delta_\mathcal{I}$
primitive negation, $P \in N_C$	$\neg P$	$\Delta_\mathcal{I} \setminus P^\mathcal{I}$
conjunction	$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
existential restrictions, $r \in N_R$	$\exists r.C$	$\{x \in \Delta_\mathcal{I} \mid \exists y : (x,y) \in r^\mathcal{I} \wedge y \in C^\mathcal{I}\}$
value restrictions, $r \in N_R$	$\forall r.C$	$\{x \in \Delta_\mathcal{I} \mid \forall y : (x,y) \in r^\mathcal{I} \rightarrow y \in C^\mathcal{I}\}$
number restrictions $r \in N_R$, $n \in \mathbb{N}$	$(\geq nr)$ $(\leq nr)$	$\{x \in \Delta_\mathcal{I} \mid \#\{y : (x,y) \in r^\mathcal{I}\} \geq n\}$, $\{x \in \Delta_\mathcal{I} \mid \#\{y : (x,y) \in r^\mathcal{I}\} \leq n\}$

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

algorithm needs to be interleaved with the proof of soundness and completeness of the characterization of subsumption, making the proofs quite involved. In [10], this approach is in fact pursued. However, in an attempt to avoid these interleaving proofs, the author made unproved assumptions concerning the existence and other properties of the lcs. Moreover, the lcs algorithm presented there is incorrect in that the computed concept description not necessarily subsumes the input descriptions.

In this work, we devise a more relaxed notion of structural normal form, which does not involve the lcs computation and therefore allows to decouple the characterization of subsumption from the lcs computation. Instead of a single $\mathcal{ALEN}$-concept description, our normal form consists of a set of $\mathcal{ALEN}$-concept descriptions, where some of the implicit facts are not made explicit.

The outline of our paper is as follows: In Section 2, we formally introduce the language $\mathcal{ALEN}$ and the lcs operation. Section 3 contains a running example and a discussion of the main difficulties that occur when computing the lcs. In subsequent sections, this example is used to illustrate the notions introduced. Section 4 then covers the characterization of subsumption and Section 5 the lcs algorithm. Finally, in Section 6, we briefly discuss the results obtained.

2 Preliminaries

Concept descriptions are inductively defined with the help of a set of *constructors*, starting with a set N_C of *concept names* and a set N_R of *role names*. In this work, we consider the DL $\mathcal{ALEN}$, i.e., concept descriptions built from the constructors shown in Table 1, subsequently called *$\mathcal{ALEN}$-concept descriptions*. Later on we will need the notion of the (role) depth of a concept description. Given an $\mathcal{ALEN}$-concept description C its depth, $\mathsf{depth}(C)$, is inductively defined as follows:

- $\mathsf{depth}(\bot) := \mathsf{depth}(\top) := \mathsf{depth}(P) := \mathsf{depth}(\neg P) := 0$;
- $\mathsf{depth}(\leq n\, r) := \mathsf{depth}(\geq n\, r) := 1$;
- $\mathsf{depth}(\exists r.C) := \mathsf{depth}(\forall r.C) := 1 + \mathsf{depth}(C)$; and
- $\mathsf{depth}(C \sqcap D) := max(\mathsf{depth}(C), \mathsf{depth}(D))$.

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

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 $C^\mathcal{I} \subseteq D^\mathcal{I}$ holds for all interpretations $\mathcal{I}$. The concept descriptions C and D are *equivalent* ($C \equiv D$) iff they subsume each other.

In this paper, we are interested in the computation of least common subsumers.

Definition 1 *Given $n \geq 2$ $\mathcal{ALEN}$-concept descriptions $C_1, \ldots, C_n$, the $\mathcal{ALEN}$-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 $C \sqsubseteq C'$.*

Depending on the DL under consideration, the lcs of two or more descriptions need not always exist, but if it exists, then it is unique up to equivalence. The main contribution of this paper is to show that in $\mathcal{ALEN}$ the lcs always exists and that it can be computed effectively.

For the sake of simplicity, we consider $\mathcal{ALEN}$-concept descriptions over a set N_C of concept names and assume N_R to be the singleton $\{r\}$. However, all definitions and results can easily be generalized to arbitrary sets of role names. Furthermore, w.l.o.g., we assume all $\mathcal{ALEN}$-concept descriptions to be in the following *normal form*: Each conjunction in an $\mathcal{ALEN}$-concept description contains

1. at most one number restriction of the form $(\geq n\ r)$
   (this is w.l.o.g. due to $(\geq m\ r) \sqcap (\geq n\ r) \equiv (\geq n\ r)$ if $n \geq m$);
2. at most one number restriction of the form $(\leq n\ r)$
   (this is w.l.o.g. due to $(\leq m\ r) \sqcap (\leq n\ r) \equiv (\leq n\ r)$ if $n \leq m$);
3. at most one value restriction of the form $\forall r.C$
   (this is w.l.o.g. due to $\forall r.C \sqcap \forall r.D \equiv \forall r.(C \sqcap D)$.

3 Running example

In order to highlight the main problems to be solved in the structural characterization of subsumption and the computation of the lcs in $\mathcal{ALEN}$, we will use the following $\mathcal{ALEN}$-concept descriptions:

$$\begin{aligned} C_{ex} &:= \exists r.(P \sqcap A_1) \sqcap \exists r.(P \sqcap A_2) \sqcap \exists r.(\neg P \sqcap A_1) \sqcap \\ &\quad\ \exists r.(Q \sqcap A_3) \sqcap \exists r.(\neg Q \sqcap A_3) \sqcap (\leq 2\, r), \text{ and} \\ D_{ex} &:= (\geq 3\, r) \sqcap \forall r.(A_1 \sqcap A_2 \sqcap A_3). \end{aligned}$$

The key point in the characterization of subsumption and the lcs computation is to describe the "non-trivial" concept descriptions, say C', subsuming a given concept description, say C, where "non-trivial" means that C' does not occur as a conjunct on the top-level of C. Subsequently, these concept descriptions are called *induced.* It suffices to only consider induced concept description that are minimal w.r.t. subsumption.

In what follows, we describe the concept descriptions induced by C_{ex} and D_{ex}. It turns out that some of the concept descriptions induced by C_{ex} correspond to the lcs of certain subdescriptions in C_{ex}. As we will see, given the induced concept descriptions of C_{ex} and D_{ex}, it is easy to determine the lcs of C_{ex} and D_{ex}. We start with the concept descriptions induced by C_{ex}.

Number restrictions: Because of the existential restrictions on the top-level of C_{ex}, e.g. $\exists r.(P \sqcap A_1)$ and $\exists r.(\neg P \sqcap A_1)$, we know $C_{ex} \sqsubseteq (\geq 2\, r)$, i.e., $(\geq 2\, r)$ is induced by C_{ex}. Conversely, there is no induced $\leq$-restriction, i.e., the most specific $\leq$-restriction subsuming C_{ex} is the $\leq$-restriction explicitly present on the top-level of C_{ex}.

Existential restrictions: Due to the $\leq$-restriction $(\leq 2\, r)$ on the top-level of C_{ex}, each instance of C_{ex} has at most two r-successors. Consequently, some existential restrictions have to be "merged" to a single existential restriction, where "merging" means conjoining the concept descriptions occurring in the existential restrictions. For C_{ex}, there are several ways to merge the five existential restrictions on the top-level of C_{ex} into two existential restrictions. The merging process gives rise to new (derived) concept descriptions, where the only consistent ones are:

$$\begin{aligned} C^1_{ex} \;&:=\; \exists r.(P \sqcap Q \sqcap A_1 \sqcap A_2 \sqcap A_3) \;\sqcap \\ & \exists r.(\neg P \sqcap \neg Q \sqcap A_1 \sqcap A_3) \sqcap (\leq 2\, r), \text{ and} \\ C^2_{ex} \;&:=\; \exists r.(P \sqcap \neg Q \sqcap A_1 \sqcap A_2 \sqcap A_3) \;\sqcap \\ & \exists r.(\neg P \sqcap Q \sqcap A_1 \sqcap A_3) \sqcap (\leq 2\, r). \end{aligned}$$

It is clear that $C_{ex} \equiv C^1_{ex} \sqcup C^2_{ex}$. The existential restrictions $\exists r.(P \sqcap A_1 \sqcap A_2 \sqcap A_3)$, $\exists r.(\neg P \sqcap A_1 \sqcap A_3)$, $\exists r.(Q \sqcap A_1 \sqcap A_3)$, and $\exists r.(\neg Q \sqcap A_1 \sqcap A_3)$ subsume both C^1_{ex} and C^2_{ex}. From this it can be concluded that these restrictions are induced by C_{ex}.

As we will see, the induced existential restrictions can be obtained by picking one existential restriction from each of the (consistent) derived concept descriptions and applying the lcs operation to them. In our example, we have, for instance, $P \sqcap A_1 \sqcap A_2 \sqcap A_3 \equiv \mathit{lcs}(P \sqcap Q \sqcap A_1 \sqcap A_2 \sqcap A_3, P \sqcap \neg Q \sqcap A_1 \sqcap A_2 \sqcap A_3)$. However, as explained below, our characterization of subsumption avoids to explicitly use the lcs by employing the fact that

$$\mathit{lcs}(C_1, \ldots, C_n) \sqsubseteq D \text{ iff } C_i \sqsubseteq D \text{ for all } 1 \leq i \leq n. \tag{1}$$

Value restrictions: In view of $C_{ex} \equiv C^1_{ex} \sqcup C^2_{ex}$, it not only follows that every instance of C_{ex} has exactly two r-successors but that these r-successors must satisfy the existential restrictions given in C^1_{ex} and C^2_{ex}. In either case, all r-successors belong to $A_1 \sqcap A_3$, and thus, the value restriction $\forall r.(A_1 \sqcap A_3)$ is induced by C_{ex}.

There are two things that should be pointed out here: First, note that there is an induced value restriction only if the number of successors induced by existential restrictions coincides with the number in the $\leq$-restriction, because only in this case, we have "full" information about *all* r-successors of an instance of C. For example, if we consider the concept description $E_{ex} := (\leq 2\, r) \sqcap \exists r.(A_1 \sqcap A_2) \sqcap \exists r.(A_1 \sqcap A_3)$, no value restriction is induced.[1]

Second, if $\forall r.C'$ is the most specific value restriction induced by C, then C' corresponds to the lcs of all concept descriptions occurring in the merged existential restrictions. In the example, $A_1 \sqcap A_3 \equiv \mathit{lcs}(P \sqcap Q \sqcap A_1 \sqcap A_2 \sqcap A_3, \neg P \sqcap \neg Q \sqcap A_1 \sqcap A_2 \sqcap A_3, P \sqcap \neg Q \sqcap A_1 \sqcap A_2 \sqcap A_3, \neg P \sqcap Q \sqcap A_1 \sqcap A_2 \sqcap A_3)$. Again, using the equivalence (1), we avoid to compute the lcs explicitly in the characterization of subsumption.

It is easy to see that the only (minimal) concept description induced by D_{ex} is $\exists r.(A_1 \sqcap A_2 \sqcap A_3)$.

[1] The structural subsumption algorithm introduced in [10], however, computes $\forall r.A_1$ as a value restriction induced by E_{ex} and thus, is incorrect. For the same reason, the lcs-algorithm presented in [10] is incorrect: although the lcs of E_{ex} and $\forall r.A_1$ is $\top$, the algorithm returns $\forall r.A_1$.

Now, given the concept descriptions induced by C_{ex} and D_{ex} it is not hard to verify that the lcs of C_{ex} and D_{ex} can be stated as

$$(\geq 2\, r) \sqcap \forall r.(A_1 \sqcap A_3) \sqcap \exists r.(A_1 \sqcap A_2 \sqcap A_3).$$

4 A structural characterization of subsumption in $\mathcal{ALEN}$

In what follows, let C, D be $\mathcal{ALEN}$-concept descriptions in the normal form introduced in Section 2. Since both $C \equiv \bot$ and $D \equiv \top$ trivially imply $C \sqsubseteq D$, our characterization of subsumption explicitly checks these equivalences. Otherwise, roughly speaking, each conjunct in D, i.e., each (negated) concept name, number restriction, existential restriction, and value restriction occurring on the top-level of D, is compared with the corresponding conjuncts in C. For the existential restrictions and value restrictions, however, it will be necessary to resort to the concept descriptions derived from C by merging existential restrictions (C^1_{ex} and C^2_{ex} in the example). If all the comparisons have succeeded, it follows $C \sqsubseteq D$. In the remainder of this section, the structural comparison between C and D is further explained. Finally, Theorem 2 establishes the complete characterization of subsumption.

We first need some notation to access the different parts of the concept descriptions.

- $\mathsf{prim}(C)$ denotes the set of all (negated) concept names occurring on the top-level of C;
- $\mathsf{min}_r(C) := max\{k \mid C \sqsubseteq (\geq k\, r)\}$ (Note that $\mathsf{min}_r(C)$ is always finite);
- $\mathsf{max}_r(C) := min\{k \mid C \sqsubseteq (\leq k\, r)\}$; if there exists no k with $C \sqsubseteq (\leq k\, r)$, then $\mathsf{max}_r(C) := \infty$;
- if there exists a value restriction of the form $\forall r.C'$ on the top-level of C, then $\mathsf{val}_r(C) := C'$; otherwise, $\mathsf{val}_r(C) := \top$;
- $\mathsf{exr}_r(C) := \{C' \mid$ there exists $\exists r.C'$ on the top-level of $C\}$;

Although the values $\mathsf{min}_r(C)$ and $\mathsf{max}_r(C)$ need not be explicitly present in number restrictions of C, they can be computed in polynomial time in the size of C using an oracle for subsumption of $\mathcal{ALEN}$-concept descriptions. For $\mathsf{min}_r(C)$, if there exists a $\geq$-restriction $(\geq m\, r)$ on the top-level of C, then $m \leq \mathsf{min}_r(C) \leq max\{m, |\mathsf{exr}_r(C)|\}$. Thus, $\mathsf{min}_r(C)$ is the maximum k with $m \leq k \leq max\{m, |\mathsf{exr}_r(C)|\}$ and $C \sqsubseteq (\geq k\, r)$. Otherwise, if C does not have a $\geq$-restriction as top-level conjunct, it suffices to check the k's between 0 and $|\mathsf{exr}_r(C)|$. For $\mathsf{max}_r(C)$, it holds that if $\mathsf{val}_r(C) \equiv \bot$, then $\mathsf{max}_r(C) = 0$; otherwise if there exists a $\leq$-restriction $(\leq m\, r)$ on the top-level of C, then $\mathsf{max}_r(C) = m$, else $\mathsf{max}_r(C) = \infty$. In our example, we get $\mathsf{min}_r(C_{ex}) = 2$, $\mathsf{max}_r(C_{ex}) = 2$, $\mathsf{min}_r(D_{ex}) = 3$, and $\mathsf{max}_r(D_{ex}) = \infty$.

The structural comparison between the different parts of C and D can now be stated as follows (assuming $C \not\equiv \bot$ and $D \not\equiv \top$):

(Negated) concept names and number restrictions: (cf Theorem 2, 1.–3.) In order for $C \sqsubseteq D$ to hold, it is obvious that the following conditions need to be satisfied: $\mathsf{prim}(D) \subseteq \mathsf{prim}(C)$, $\mathsf{max}_r(C) \leq \mathsf{max}_r(D)$, $\mathsf{min}_r(C) \geq \mathsf{min}_r(D)$. Otherwise, it is easy to construct a counter-model for $C \sqsubseteq D$.

Existential restrictions: (cf Theorem 2, 4.) Given $D' \in \mathsf{exr}_r(D)$, then for $C \sqsubseteq D$ to hold one at first might expect that there need to exist a $C' \in \mathsf{exr}_r(C)$ with $\mathsf{val}_r(C) \sqcap C' \sqsubseteq D'$. Although this works for $\mathcal{ALE}$-concept descriptions, it fails for $\mathcal{ALEN}$, as $C_{ex} \sqsubseteq \exists r.(P \sqcap A_1 \sqcap A_2 \sqcap A_3)$ shows (see Section 3). The reason is that in $\mathcal{ALEN}$ $\leq$-restrictions may require to merge existential restrictions, yielding new (implicit) existential restrictions. To deal with this phenomenon, sets of derived concept descriptions are considered with certain existential restrictions merged (see C^1_{ex} and C^2_{ex} in our running example).

The result of merging existential restrictions is described by so-called *existential mappings*

$$\alpha : \{1, \ldots, n\} \longrightarrow 2^{\{1,\ldots,m\}},$$

where $n := min\{\mathsf{max}_r(C), |\mathsf{exr}_r(C)|\}$ and $m := |\mathsf{exr}_r(C)|$. We require α to obey the following conditions:

1. $\alpha(i) \neq \emptyset$ for all $1 \leq i \leq n$;
2. $\bigcup_{1 \leq i \leq n} \alpha(i) = \{1, \ldots, m\}$ and $\alpha(i) \cap \alpha(j) = \emptyset$ for all $1 \leq i < j \leq n$.
3. $\bigsqcap\limits_{j \in \alpha(i)} C_j \sqcap \mathsf{val}_r(C) \not\equiv \bot$ for all $1 \leq i \leq n$.

Although the first two conditions are not essential for soundness and completeness of the characterization of subsumption, they reduce the number of existential mappings that need to be considered.

Given $\mathsf{exr}_r(C) = \{C_1, \ldots, C_m\}$, α yields an $\mathcal{ALEN}$-concept description C^α obtained from C by substituting all existential restrictions on the top-level of C with

$$\bigsqcap_{1 \leq i \leq n} \exists r. \bigsqcap_{j \in \alpha(i)} C_j.$$

The set of all existential mappings on C satisfying the conditions (1)–(3) is denoted by $\Gamma_r(C)$, where $\Gamma_r(C) := \emptyset$ if $\mathsf{exr}_r(C) = \emptyset$. It is in fact sufficient to consider $\Gamma_r(C)$ modulo permutations, i.e., modulo the equivalence

$$\alpha \cong \alpha' \quad \text{iff} \quad \begin{array}{l} \text{there exists a permutation } \pi \text{ on } \{1, \ldots, n\} \\ \text{s.t. } \alpha(i) = \alpha'(\pi(i)) \text{ for all } 1 \leq i \leq n. \end{array}$$

In the sequel, for the sake of simplicity, we stay with $\alpha \in \Gamma_r(C)$ instead of $[\alpha]_{\cong} \in \Gamma_r(C)/_{\cong}$, though.

In our running example, let $\mathsf{exr}_r(C_{ex}) = \{C_{ex,1}, \ldots, C_{ex,5}\}$ with $C_{ex,1} = P \sqcap A_1$, $C_{ex,2} = P \sqcap A_2$, etc. Then, $\Gamma_r(C_{ex})$ consists of the two mappings

$$\begin{aligned} \alpha_1 &= \{1 \mapsto \{1,2,4\}, 2 \mapsto \{3,5\}\}; \\ \alpha_2 &= \{1 \mapsto \{1,2,5\}, 2 \mapsto \{3,4\}\}, \end{aligned}$$

and it is $C^{\alpha_i}_{ex} = C^i_{ex}$, $i = 1, 2$ (see Section 3).

For the characterization of subsumption, we will use the following notation:

$$\mathsf{exr}_r(C)^\alpha := \{\bigsqcap_{j \in \alpha(i)} C_j \mid 1 \leq i \leq n\}.$$

Now, in case $\mathsf{exr}_r(C) \neq \emptyset$, $C \sqsubseteq D$ implies that for each $D' \in \mathsf{exr}_r(D)$ the following holds: for each $\alpha \in \Gamma_r(C)$, there exists $C'_\alpha \in \mathsf{exr}_r(C)^\alpha$ such that $C'_\alpha \sqcap \mathsf{val}_r(C) \sqsubseteq D'$. This is what is stated in Theorem 2, 4. Note that, using the equivalence (1) and provided that the lcs of $\mathcal{ALEN}$-concept descriptions always exists, this condition could be rewritten as: there exists a set $M := \{C'_\alpha \in \mathsf{exr}_r(C)^\alpha \mid \alpha \in \Gamma_r(C)\}$ with $lcs(\{C' \sqcap \mathsf{val}_r(C) \mid C' \in M\}) \sqsubseteq D'$. However, since the existence of the lcs is not guaranteed a priori, the lcs cannot be used in the structural characterization of subsumption, unless the characterization and the lcs computation are interleaved.

If $\mathsf{exr}_r(C) = \emptyset$, then for all $D' \in \mathsf{exr}_r(D)$ it must hold, $\min_r(C) \geq 1$ and $\mathsf{val}_r(C) \sqsubseteq D'$.

In our running example, $C_{ex} \sqsubseteq \exists r.(P \sqcap A_1 \sqcap A_2 \sqcap A_3)$ illustrates the case where $\mathsf{exr}_r(C) \neq \emptyset$ and $D_{ex} \sqsubseteq \exists r.(A_1 \sqcap A_2 \sqcap A_3)$ illustrates $\mathsf{exr}_r(C) = \emptyset$.

Value restrictions: (cf Theorem 2, 5.) Value restrictions can only be induced for two reasons. First, if $\mathsf{max}_r(C) = 0$, then $C \sqsubseteq \forall r.\bot$, and thus, $C \sqsubseteq \forall r.C'$ for all concept descriptions C'.

Second, the merging of existential restrictions may induce value restrictions. In contrast to induced existential restrictions, however, one further needs to take into account $\geq$-restrictions induced by "incompatible" existential restrictions:

$$\kappa_r(C) := \mathsf{min}_r(\forall r.\mathsf{val}_r(C) \sqcap \textstyle\bigsqcap_{C' \in \mathsf{exr}_r(C)} \exists r.C').$$

If $\mathsf{exr}_r(C) = \emptyset$, we define $\kappa_r(C) := 0$. In our example, $\kappa_r(C_{ex}) = 2$, $\kappa_r(D_{ex}) = 0$, and $\kappa_r(E_{ex}) = 1$.

Now, only if $\kappa_r(C) = \mathsf{max}_r(C)$, value restrictions can be induced, since only then we "know" all the r-successors of instances of C. In our example, $\kappa_r(C_{ex}) = \mathsf{max}_r(C_{ex}) = 2$, which accounts for $C_{ex} \sqsubseteq \forall r.(A_1 \sqcap A_3)$. Conversely, $E_{ex} \not\sqsubseteq \forall r.A_1$ since $1 = \kappa_r(E_{ex}) < \mathsf{max}_r(E_{ex}) = 2$.

To formally state the comparison between value restrictions of C and D, we use the following notation:

$$\mathsf{exr}_r(C)^* := \bigcup_{\alpha \in \Gamma_r(C)} \mathsf{exr}_r(C)^\alpha.$$

One can show that $C \sqsubseteq D$ implies $\mathsf{val}_r(C) \sqsubseteq \mathsf{val}_r(D)$ provided that $\kappa_r(C) < \mathsf{max}_r(C)$. In case $0 < \kappa(C) = \mathsf{max}_r(C)$, however, it suffices if the value restriction on the top-level of D satisfies $\mathsf{val}_r(C) \sqcap C' \sqsubseteq \mathsf{val}_r(D)$ for all $C' \in \mathsf{exr}_r(C)^*$. Again, using the equivalence (1) and provided that the lcs of $\mathcal{ALEN}$-concept descriptions always exists, this condition can be restated as $\mathsf{val}_r(C) \sqcap lcs(\mathsf{exr}_r(C)^*) \sqsubseteq D'$. For reasons already mentioned, we have not employed this variant.

We are now equipped for the structural characterization of subsumption in $\mathcal{ALEN}$.

Theorem 2 *Let C, D be two $\mathcal{ALEN}$-concept descriptions with $\mathsf{exr}_r(C) = \{C_1, \ldots, C_m\}$. Then $C \sqsubseteq D$ iff $C \equiv \bot$, $D \equiv \top$, or the following holds:*

1. $\mathsf{prim}(D) \subseteq \mathsf{prim}(C)$;
2. $\mathsf{max}_r(C) \leq \mathsf{max}_r(D)$;
3. $\mathsf{min}_r(C) \geq \mathsf{min}_r(D)$;
4. *for all $D' \in \mathsf{exr}_r(D)$ it holds that*
   - (a) $\mathsf{exr}_r(C) = \emptyset$, $\mathsf{min}_r(C) \geq 1$, *and* $\mathsf{val}_r(C) \sqsubseteq D'$; *or*
   - (b) $\mathsf{exr}_r(C) \neq \emptyset$ *and for each* $\alpha \in \Gamma_r(C)$, *there exists* $C' \in \mathsf{exr}_r(C)^\alpha$ *such that* $C' \sqcap \mathsf{val}_r(C) \sqsubseteq D'$; *and*
5. *if* $\mathsf{val}_r(D) \not\equiv \top$, *then*

(a) $\mathsf{max}_r(C) = 0$; *or*

(b) $\kappa_r(C) < \mathsf{max}_r(C)$ *and* $\mathsf{val}_r(C) \sqsubseteq \mathsf{val}_r(D)$; *or*

(c) $0 < \kappa_r(C) = \mathsf{max}_r(C)$ *and* $\mathsf{val}_r(C) \sqcap C' \sqsubseteq \mathsf{val}_r(D)$ *for all* $C' \in \mathsf{exr}_r(C)^*$.

In the remainder of this section, we provide a formal proof of the above theorem. For the proof of soundness, i.e., the if-direction of Theorem 2, we need the following lemma.

Lemma 3 *Let C be an $\mathcal{ALEN}$-concept description with $\mathsf{exr}_r(C) \neq \emptyset$. Then, for each model $\mathcal{I}$ of C and each $x \in C^{\mathcal{I}}$, there exists $\alpha \in \Gamma_r(C)$ such that for each $C' \in \mathsf{exr}_r(C)^{\alpha}$, there exists $y \in \Delta_{\mathcal{I}}$ with $(x,y) \in r^{\mathcal{I}}$ and $y \in C'^{\mathcal{I}}$.*

Proof: By induction on $n := |\mathsf{exr}_r(C)| - \mathsf{max}_r(C)$.

Let $n \leq 0$: Then w.l.o.g. $\Gamma_r(C)$ is given by the singleton $\{id\}$, where id denotes the mapping $id(i) := \{i\}$, $1 \leq i \leq m$. We get $\mathsf{exr}_r(C)^{id} = \mathsf{exr}_r(C)$. Since $x \in C^{\mathcal{I}}$, for each $C' \in \mathsf{exr}_r(C)$ there exists an r-successor y of x in $\mathcal{I}$ with $y \in C'^{\mathcal{I}}$.

Let $n > 0$: Since $x \in C^{\mathcal{I}}$, there exist at most $\mathsf{max}_r(C)$ r-successors of x in $\mathcal{I}$. Let $\mathsf{exr}_r(C) = \{C_1, \ldots, C_m\}$. Since $|\mathsf{exr}_r(C)| > \mathsf{max}_r(C)$, there exist $C_{i_1}, C_{i_2} \in \mathsf{exr}_r(C)$, $i_1 \neq i_2$, such that there exists an r-successor y of x in $\mathcal{I}$ with $y \in C_{i_1}^{\mathcal{I}}$ and $y \in C_{i_2}^{\mathcal{I}}$. W.l.o.g. let $i_1 = m-1$ and $i_2 = m$.

Now, let D be the $\mathcal{ALEN}$-concept description obtained from C by removing $\exists r.C_{m-1}$ and $\exists r.C_m$ from the top-level of C and conjoining $\exists r.(C_{m-1} \sqcap C_m)$ instead. Again w.l.o.g., let $\mathsf{exr}_r(D) = \{D_1, \ldots, D_{m-1}\}$ with $D_{m-1} = C_{m-1} \sqcap C_m$ and $D_i = C_i$ for all $1 \leq i \leq m-2$. Since $|\mathsf{exr}_r(D)| - \mathsf{max}_r(D) < n$, we get by induction that there exists a $\beta \in \Gamma_r(D)$ such that

$$\text{for each } D' \in \mathsf{exr}_r(D)^{\beta}, \text{ there exists } y \in \Delta_{\mathcal{I}} \text{ with } (x,y) \in r^{\mathcal{I}} \text{ and } y \in D'^{\mathcal{I}}. \qquad (*)$$

We extend β to a mapping $\alpha \in \Gamma_r(C)$ as follows:

$$\alpha(i) \quad := \quad \begin{cases} \beta(i), & \text{if } m-1 \notin \beta(i); \\ \beta(i) \cup \{m\}, & \text{if } m-1 \in \beta(i). \end{cases}$$

It is easy to see that $\alpha \in \Gamma_r(C)$. Now let $C' \in \mathsf{exr}_r(C)^{\alpha}$. Then there exists $i_0 \in \{1, \ldots, \mathsf{max}_r(C)\}$ with $C' = \bigsqcap_{j \in \alpha(i_0)} C_j$. If $m \notin \alpha(i_0)$, then $\alpha(i_0) = \beta(i_0)$ and by (*), we get that there exists an r-successor y of x in $\mathcal{I}$ with $y \in C'^{\mathcal{I}}$. If $m \in \alpha(i_0)$, then by definition of α, we get that $m-1 \in \beta(i_0)$. Since $D_{m-1} = C_{m-1} \sqcap C_m$, we get $\bigsqcap_{j \in \alpha(i_0)} C_j \equiv \bigsqcap_{j \in \beta(i_0)} D_j$. Thus, (*) implies that there exists an r-successor y of x in $\mathcal{I}$ with $y \in C'^{\mathcal{I}}$. □

Proof of the if-direction of Theorem 2: If $C \equiv \bot$ or $D \equiv \top$, then $C \sqsubseteq D$. Assume $C \not\equiv \bot$ and $D \not\equiv \top$, and (1) − −(5) are satisfied. In order to show $C \sqsubseteq D$, we show that for any model $\mathcal{I}$ of C with $x \in C^{\mathcal{I}}$ also $x \in D^{\mathcal{I}}$. Obviously, it is sufficient to show $x \in D'^{\mathcal{I}}$ for each conjunct D' occurring on the top-level of D.

Let $D' \in \mathsf{prim}(D)$: Since $\mathsf{prim}(D) \subseteq \mathsf{prim}(C)$ and $x \in C^{\mathcal{I}}$, we obtain $x \in D'^{\mathcal{I}}$.

Let $D' = (\leq n\, r)$**:** By condition (2), it is $\mathsf{max}_r(C) \leq \mathsf{max}_r(D) \leq n$. Since $x \in C^{\mathcal{I}}$, it follows $x \in (\leq \mathsf{max}_r(C)\, r)^{\mathcal{I}} \subseteq (\leq n\, r)^{\mathcal{I}}$.

Let $D' = (\geq n\, r)$**:** By condition (3), it is $\mathsf{min}_r(C) \geq \mathsf{min}_r(D) \geq m$. Since $x \in C^{\mathcal{I}}$, it follows $x \in (\geq \mathsf{min}_r(C)\, r)^{\mathcal{I}} \subseteq (\geq m\, r)^{\mathcal{I}}$.

Let $D' = \exists r.D_1$**:** Assume $\mathsf{exr}_r(C) = \emptyset$. Then (4.$a$) implies $\mathsf{min}_r(C) \geq 1$ and $\mathsf{val}_r(C) \sqsubseteq D_1$. Hence, there exists an r-successor y of x in $\mathcal{I}$ with $y \in \mathsf{val}_r(C)^{\mathcal{I}} \subseteq D_1^{\mathcal{I}}$. Thus, $x \in (\exists r.D_1)^{\mathcal{I}}$.

Assume $\mathsf{exr}_r(C) \neq \emptyset$. By Lemma 3, there exists $\alpha' \in \Gamma_r(C)$ such that for each $C' \in \mathsf{exr}_r(C)^{\alpha'}$ there exists an r-successor y of x in $\mathcal{I}$ with $y \in C'^{\mathcal{I}}$. By condition (4.b), there exists $C' \in \mathsf{exr}_r(C)^{\alpha'}$ with $C' \sqcap \mathsf{val}_r(C) \sqsubseteq D_1$. Thus, there is an r-successor y' of x in $\mathcal{I}$ with $y' \in (C' \sqcap \mathsf{val}_r(C))^{\mathcal{I}} \subseteq D_1^{\mathcal{I}}$. Hence, $x \in (\exists r.D_1)^{\mathcal{I}}$.

Let $D' = \forall r.D_1$**:** If $\mathsf{max}_r(C) = 0$, then there exists no r-successor of x in $\mathcal{I}$, and hence, $x \in (\forall r.D_1)^{\mathcal{I}}$.

Assume $\kappa_r(C) < \mathsf{max}_r(C)$. Let y be an arbitrary r-successor of $x \in \mathcal{I}$. Then $x \in C^{\mathcal{I}}$ implies $y \in \mathsf{val}_r(C)^{\mathcal{I}}$ and since $\mathsf{val}_r(C) \sqsubseteq \mathsf{val}_r(D)$, we get $y \in \mathsf{val}_r(D)^{\mathcal{I}}$. Thus, $x \in (\forall r.D_1)^{\mathcal{I}}$.

Finally, assume $0 < \kappa_r(C) = \mathsf{max}_r(C)$. Let y be an arbitrary r-successor of x in $\mathcal{I}$. Define $\mathcal{C}(y) := \{C' \in \mathsf{exr}_r(C) \mid y \in C'^{\mathcal{I}}\}$. Since $0 < \kappa_r(C) = \mathsf{max}_r(C)$, we know that there exist exactly $\kappa_r(C)$ r-successors of x in $\mathcal{I}$, and since there exist $\kappa_r(C)$ disjoint existential restrictions on the top-level of C, we get $\mathcal{C}(y) \neq \emptyset$. We even get $\bigsqcap_{E \in \mathcal{C}(y)} E \sqsubseteq C'$ for some $C' \in \mathsf{exr}_r(C)$, because otherwise, $\mathcal{I}$ would not be a model of C. Now, $\mathsf{val}_r(C) \sqcap C' \sqsubseteq \mathsf{val}_r(D)$ implies $y \in \mathsf{val}_r(D)^{\mathcal{I}}$ and hence, $x \in (\forall r.D_1)^{\mathcal{I}}$. □

For the proof of completeness of the characterization of subsumption we will have to construct certain models of $\mathcal{ALEN}$-concept descriptions. The following lemma ensures the existence of these models. It easily follows from the definition of $\mathsf{min}_r(C)$ and the (finite) tree model property of $\mathcal{ALEN}$-concept descriptions [8].

Lemma 4 *Let C be an $\mathcal{ALEN}$-concept description, $C \not\equiv \bot$. Then there exists a tree model $\mathcal{I}$ with root x_0 of C such that $x_0 \in C^{\mathcal{I}}$ and x_0 has exactly $\mathsf{min}_r(C)$ different r-successors in $\mathcal{I}$.*

Proof of the only-if-direction of Theorem 2: If $C \equiv \bot$ or $D \equiv \top$, nothing has to be shown. Assume $C \sqsubseteq D$, $C \not\equiv \bot$, and $D \not\equiv \top$. Then we have to show that C, D satisfy items (1)–(5) in Theorem 2.

Ad (1)**:** Assume that there exists $Q \in \mathsf{prim}(D) \setminus \mathsf{prim}(C)$. Let $\mathcal{I} = (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a tree model with root x_0 of C with $x_0 \in C^{\mathcal{I}}$ (see Lemma 4).

If $Q \in N_C$, then define $\mathcal{J} := (\Delta_{\mathcal{J}}, \cdot^{\mathcal{J}})$ with

- $P^{\mathcal{J}} := \begin{cases} P^{\mathcal{I}} \setminus \{x_0\}, & P = Q; \\ P^{\mathcal{I}}, & P \neq Q; \end{cases}$
- $r^{\mathcal{J}} := r^{\mathcal{I}}$.

If $Q = \neg P'$ for some $P' \in N_C$, then define $\mathcal{J} := (\Delta_{\mathcal{J}}, \cdot^{\mathcal{J}})$ with

- $P^{\mathcal{J}} := \begin{cases} P^{\mathcal{I}} \cup \{x_0\}, & P = P'; \\ P^{\mathcal{I}}, & P \neq P'; \end{cases}$

- $r^{\mathcal{J}} := r^{\mathcal{I}}$.

In either case, $\mathcal{J}$ is a model of C with $x_0 \in C^{\mathcal{J}}$, but $x_0 \notin D^{\mathcal{J}}$ in contradiction to $C \sqsubseteq D$.

Thus, $\mathsf{prim}(D) \subseteq \mathsf{prim}(C)$.

Ad (2)**:** Assume $\mathsf{max}_r(C) > \mathsf{max}_r(D)$. In particular, $\mathsf{max}_r(D)$ is some nonnegative integer. By the definition of $\mathsf{max}_r(C)$ there exists a model $\mathcal{I}$ and $x_0 \in \Delta_{\mathcal{I}}$ with $x_0 \in C^{\mathcal{I}}$ such that x_0 has $\mathsf{max}_r(D) + 1$ r-successors; otherwise we would have $\mathsf{max}_r(C) \leq \mathsf{max}(D)$. Obviously, $x_0 \notin D^{\mathcal{I}}$. Thus, $\mathsf{max}_r(C) \leq \mathsf{max}_r(D)$.

Ad (3)**:** Assume $\mathsf{min}_r(C) < \mathsf{min}_r(D)$. By Lemma 4, there exists a (tree) model $\mathcal{I}$ with $x_0 \in C^{\mathcal{I}}$ having exactly $\mathsf{min}_r(C)$ r-successors. Obviously, $x_0 \notin D^{\mathcal{I}}$.

Ad (4)**:** Let $D' \in \mathsf{exr}_r(D)$ and assume $\mathsf{exr}_r(C) = \emptyset$. If $\mathsf{min}_r(C) = 0$, then there exists a tree model $\mathcal{I} = (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ with root x_0 of C with $x_0 \in C^{\mathcal{I}}$ and x_0 has 0 r-successors in $\mathcal{I}$, i.e., $r^{\mathcal{I}} = \emptyset$. Obviously, this implies $x_0 \notin D^{\mathcal{I}}$ in contradiction to $C \sqsubseteq D$. Hence, $\mathsf{min}_r(C) \geq 1$. Let $\mathcal{I} = (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ be a tree model of C with root x_0 and $x_0 \in C^{\mathcal{I}}$. Assume $\mathsf{val}_r(C) \not\sqsubseteq D'$, i.e., there exists a tree model $\mathcal{I}' = (\Delta_{\mathcal{I}'}, \cdot^{\mathcal{I}'})$ with root y_0 such that $y_0 \in \mathsf{val}_r(C)^{\mathcal{I}'}$ and $y_0 \notin D'^{\mathcal{I}'}$. Let $\mathcal{I}_i$, $1 \leq i \leq \mathsf{min}_r(C)$, be pairwise disjoint copies of $\mathcal{I}'$ with roots y_i, which are also pairwise disjoint with $\mathcal{I}$. Define $\mathcal{J} := (\Delta_{\mathcal{J}}, \cdot^{\mathcal{J}})$ as follows:

- $\Delta_{\mathcal{J}} := \Delta_{\mathcal{I}} \cup \bigcup_{1 \leq i \leq \mathsf{min}_r(C)} \Delta_{\mathcal{I}_i}$;
- $P^{\mathcal{J}} := P^{\mathcal{I}} \cup \bigcup_{1 \leq i \leq \mathsf{min}_r(C)} P^{\mathcal{I}_i}$ for all $P \in N_C$;
- $r^{\mathcal{J}} := \bigcup_{1 \leq i \leq \mathsf{min}_r(C)} r^{\mathcal{I}_i} \cup \{(x_0, y_i) \mid 1 \leq i \leq \mathsf{min}_r(C)\}$.

It is easy to see that $\mathcal{I}$ is a model of C with $x_0 \in C^{\mathcal{I}}$, but since there does not exist an r-successor y of x_0 with $y \in D'^{\mathcal{J}}$, it is $x_0 \notin D^{\mathcal{I}}$ in contradiction to $C \sqsubseteq D$. Thus, $\mathsf{val}_r(C) \sqsubseteq D'$.

Now assume $\mathsf{exr}_r(C) \neq \emptyset$. We have to show that for each $\alpha \in \Gamma_r(C)$, there exists $C' \in \mathsf{exr}_r(C)^\alpha$ with $C' \sqcap \mathsf{val}_r(C) \sqsubseteq D'$. Assume that there exists an $\alpha \in \Gamma_r(C)$ such that $C' \sqcap \mathsf{val}_r(C) \not\sqsubseteq D'$ for all $C' \in \mathsf{exr}_r(C)^\alpha$. Let C_α be the $\mathcal{ALEN}$-concept description obtained from C by removing all existential restrictions from the top-level of C and conjoining all existential restrictions from $\mathsf{exr}_r(C)^\alpha$. We define a tree model $\mathcal{I}$ with root x_0 such that $x_0 \in C_\alpha^{\mathcal{I}} \subseteq C^{\mathcal{I}}$ and $x_0 \notin D^{\mathcal{I}}$ as follows:

- Define $\mathcal{I}_0 := (\{x_0\}, \cdot^{\mathcal{I}_0})$ with
  - $P^{\mathcal{I}_0} := \begin{cases} \{x_0\}, & P \in \mathsf{prim}(C); \\ \emptyset, & P \notin \mathsf{prim}(C) \end{cases}$ for all $P \in N_C$;
  - $r^{\mathcal{I}_0} := \emptyset$.
- For each $C' \in \mathsf{exr}_r(C)^\alpha$, let $\mathcal{I}(C') := (\Delta_{\mathcal{I}(C')}, \cdot^{\mathcal{I}(C')})$ be a tree model with root $y(C')$ such that $y(C') \in (C' \sqcap \mathsf{val}_r(C))^{\mathcal{I}(C')}$ and $y(C') \notin D'^{\mathcal{I}(C')}$.
- If $\mathsf{min}_r(C) > |\mathsf{exr}_r(C)^\alpha|$, then let $\mathcal{I}_1, \ldots, \mathcal{I}_k$, $k := \mathsf{min}_r(C) - |\mathsf{exr}_r(C)^\alpha|$, be copies of $\mathcal{I}(C')$, as defined above, for some C' with root y_i, $1 \leq i \leq k$.

W.l.o.g. let all these interpretations be pairwise disjoint. Now, define $\mathcal{I} = (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ by

- $\Delta_{\mathcal{I}} := \{x_0\} \cup \bigcup_{1 \leq i \leq k} \Delta_{\mathcal{I}_i} \cup \bigcup_{C' \in \mathsf{exr}_r(C)^\alpha} \Delta_{\mathcal{I}(C')}$;

- $P^{\mathcal{I}} := P^{\mathcal{I}_0} \cup \bigcup_{1\le i\le k} P^{\mathcal{I}_i} \cup \bigcup_{C'\in \mathsf{exr}_r(C')^\alpha} P^{\mathcal{I}(C')}$ for all $P \in N_C$;
- $r^{\mathcal{I}} := \{(x_0, y(C')) \mid C' \in \mathsf{exr}_r(C)^\alpha\} \cup \{(x_0, y_i) \mid 1 \le i \le k\} \cup \bigcup_{1\le i\le k} r^{\mathcal{I}_i} \cup \bigcup_{C'\in \mathsf{exr}_r(C)^\alpha} r^{\mathcal{I}(C')}$.

By construction, $\mathcal{I}$ is a model of C_α and hence of C with $x_0 \in C^{\mathcal{I}}$. But since there does not exist an r-successor y of x_0 with $y \in D'^{\mathcal{I}}$, it follows $x_0 \notin D^{\mathcal{I}}$ in contradiction to $C \sqsubseteq D$. Thus, for each $\alpha \in \Gamma_r(C)$ there exists $C' \in \mathsf{exr}_r(C)^\alpha$ with $C' \sqcap \mathsf{val}_r(C) \sqsubseteq D'$.

Ad (5)**:** Assume $\mathsf{val}_r(D) \not\equiv \top$ and $\mathsf{max}_r(C) > 0$ (otherwise, nothing has to be shown). We have to distinguish two cases:

1. $\kappa_r(C) < \mathsf{max}_r(C)$. We have to show $\mathsf{val}_r(C) \sqsubseteq \mathsf{val}_r(D)$. Proof by contraposition: assuming $\mathsf{val}_r(C) \not\sqsubseteq \mathsf{val}_r(D)$, we define a tree model $\mathcal{I}$ with root x_0 of C such that $x_0 \in C^{\mathcal{I}}$ and $x_0 \notin D^{\mathcal{I}}$.

   In order to define $\mathcal{I}$, we distinguish two cases:

   (a) there exists a $\geq$-restriction of the form $(\geq m\, r)$ on the top-level of C with $m > \kappa_r(C)$, and

   (b) there exists no such number restriction.

   Ad (a): Consider the $\mathcal{ALEN}$-concept description C' obtained from C by removing $(\geq m\, r)$ from the top-level of C. Obviously, it is $\kappa_r(C) = \kappa_r(C')$. Let $\mathcal{I}'$ be a tree model of C' with root x_0 such that $x_0 \in C'^{\mathcal{I}}$ and x_0 has exactly $\kappa_r(C')$ r-successors in $\mathcal{I}'$ (see Lemma 4). By assumption, there exists a model $\mathcal{I}''$ of $\mathsf{val}_r(C)$ with root y_0 such that $y_0 \in \mathsf{val}_r(C)^{\mathcal{I}''}$ and $y_0 \notin \mathsf{val}_r(D)^{\mathcal{I}''}$. Let $\mathcal{I}_i$, $1 \le i \le m - \kappa_r(C)$, be disjoint copies of $\mathcal{I}''$ with roots y_i, which are w.l.o.g. also pairwise disjoint with $\mathcal{I}'$. Define $\mathcal{I} := (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ with

   - $\Delta_{\mathcal{I}} := \Delta'_{\mathcal{I}} \cup \bigcup_{1\le i\le m-\kappa_r(C)} \Delta_{\mathcal{I}_i}$;
   - $P^{\mathcal{I}} := P^{\mathcal{I}'} \cup \bigcup_{1\le i\le m-\kappa_r(C)} P^{\mathcal{I}_i}$ for all $P \in N_C$;
   - $r^{\mathcal{I}} := r^{\mathcal{I}'} \cup \bigcup_{1\le i\le m-\kappa_r(C)} r^{\mathcal{I}_i} \cup \{(x_0, y_i) \mid 1 \le i \le m - \kappa_r(C)\}$.

   It is easy to see that $\mathcal{I}$ is a model of C with $x_0 \in C^{\mathcal{I}}$, but since x_0 has at least the r-successor y_1 with $y_1 \notin \mathsf{val}_r(D)^{\mathcal{I}}$, we get $x_0 \notin D^{\mathcal{I}}$ in contradiction to $C \sqsubseteq D$. Thus, $\mathsf{val}_r(C) \sqsubseteq \mathsf{val}_r(D)$.

   Ad (b): Let $\mathcal{I}'$ be a tree model of C' with root x_0 such that $x_0 \in C^{\mathcal{I}}$ and x_0 has exactly $\kappa_r(C)$ r-successors in $\mathcal{I}'$ (see Lemma 4). By assumption, there exists a model $\mathcal{I}''$ of $\mathsf{val}_r(C)$ with root y_0 such that $y_0 \in \mathsf{val}_r(C)^{\mathcal{I}''}$ and $y_0 \notin \mathsf{val}_r(D)^{\mathcal{I}''}$. Define $\mathcal{I} := (\Delta_{\mathcal{I}}, \cdot^{\mathcal{I}})$ with

   - $\Delta_{\mathcal{I}} := \Delta'_{\mathcal{I}} \cup \Delta_{\mathcal{I}''}$;
   - $P^{\mathcal{I}} := P^{\mathcal{I}'} \cup P^{\mathcal{I}''}$ for all $P \in N_C$;
   - $r^{\mathcal{I}} := r^{\mathcal{I}'} \cup r^{\mathcal{I}''} \cup \{(x_0, y_0)\}$.

   Since by assumption, $\kappa_r(C) < \mathsf{max}_r(C)$, $\mathcal{I}$ is a model of C with $x_0 \in C^{\mathcal{I}}$, but since x_0 has at least the r-successor y_0 with $y_0 \notin \mathsf{val}_r(D)^{\mathcal{I}}$, we get $x_0 \notin D^{\mathcal{I}}$ in contradiction to $C \sqsubseteq D$. Thus, $\mathsf{val}_r(C) \sqsubseteq \mathsf{val}_r(D)$.

2. $\kappa_r(C) = \mathsf{max}_r(C)$. Assume that there exists $C' \in \mathsf{exr}_r(C)^*$ with $\mathsf{val}_r(C) \sqcap C' \not\sqsubseteq \mathsf{val}_r(D)$. Then there exists $\alpha' \in \Gamma_r(C)$ such that $C' \in \mathsf{exr}_r(C)^{\alpha'}$. Let $C_{\alpha'}$ be the $\mathcal{ALEN}$-concept description obtained from C by removing all existential restrictions from the top-level of C and conjoining all existential restrictions from $\mathsf{exr}_r(C)^{\alpha'}$. Obviously, $C_{\alpha'} \sqsubseteq C$. Since, using $C \not\equiv \bot$, from the definition of existential mappings it immediately follows that $C_{\alpha'} \not\equiv \bot$,

Let C, D be two $\mathcal{ALEN}$-concept descriptions.
If $C \sqsubseteq D$, then $\mathsf{c\text{-}lcs}(C, D) := D$, and if $D \sqsubseteq C$, then $\mathsf{c\text{-}lcs}(C, D) := C$.
Otherwise, let $\Gamma_r(C) = \{\alpha_1^C, \ldots, \alpha_{n(C)}^C\}$ and $\Gamma_r(D) = \{\alpha_1^D, \ldots, \alpha_{n(D)}^D\}$, and define for $E \in \{C, D\}$

$$\mathcal{C}_r(E) := \begin{cases} \{\mathsf{c\text{-}lcs}(\{E_1 \sqcap \mathsf{val}_r(E), \ldots, E_{n(E)} \sqcap \mathsf{val}_r(E)\}) \mid E_i \in \mathsf{exr}_r(E)^{\alpha_i^E}, 1 \le i \le n(E)\}, & \Gamma_r(E) \neq \emptyset, \\ \{\mathsf{val}_r(E)\}, & \Gamma_r(E) = \emptyset \wedge \mathsf{min}_r(E) \ge 1, \\ \emptyset, & \mathsf{min}_r(E) = 0; \end{cases}$$

$$E_r^* := \begin{cases} \mathsf{val}_r(E), & 0 \le \kappa_r(E) < \mathsf{max}_r(E), \\ \bot, & \mathsf{max}_r(E) = 0, \\ \mathsf{c\text{-}lcs}(\{\mathsf{val}_r(E) \sqcap E' \mid E' \in \mathsf{exr}_r(E)^*\}), & 0 < \kappa_r(E) = \mathsf{max}_r(E). \end{cases}$$

Define

$$\begin{aligned} \mathsf{c\text{-}lcs}(C, D) \quad := \quad & \textstyle\bigsqcap_{Q \in \mathsf{prim}(C) \cap \mathsf{prim}(D)} Q \ \sqcap \\ & (\le \mathit{max}\{\mathsf{max}_r(C), \mathsf{max}_r(D)\}\ r) \sqcap (\ge \mathit{min}\{\mathsf{min}_r(C), \mathsf{min}_r(D)\}\ r) \ \sqcap \\ & \textstyle\bigsqcap_{C' \in \mathcal{C}_r(C), D' \in \mathcal{C}_r(D)} \exists r.\mathsf{c\text{-}lcs}(C', D') \ \sqcap \\ & \forall r.\mathsf{c\text{-}lcs}(C_r^*, D_r^*), \end{aligned}$$

where

- $(\le \mathit{max}\{\mathsf{max}_r(C), \mathsf{max}_r(D)\}\ r)$ is omitted if $\mathsf{max}_r(C) = \infty$ or $\mathsf{max}_r(D) = \infty$, and
- $\bigsqcap_{C' \in \mathcal{C}_r(C), D' \in \mathcal{C}_r(D)} \exists r.\mathsf{c\text{-}lcs}(C', D') := \top$ if $\mathcal{C}_r(C) = \emptyset$ or $\mathcal{C}_r(D) = \emptyset$.

Figure 1: The recursive computation of the lcs in $\mathcal{ALEN}$.

there exists a tree model $\mathcal{I}$ of $C_{\alpha'}$ with root $x_0 \in C_{\alpha'}^{\mathcal{I}}$. Moreover, since $\mathsf{max}_r(C) = \kappa_r(C)$, for every $C'' \in \mathsf{exr}_r(C)^{\alpha'}$, there exists exactly one r-successor y of x_0 in $\mathcal{I}$ with $y \in C''^{\mathcal{I}}$; x_0 has no other r-successors. In particular, there exists an r-successor y of x_0 such that $y \in (\mathsf{val}_r(C) \sqcap C')^{\mathcal{I}}$. By assumption, there exists a tree model $\mathcal{I}' = (\Delta_{\mathcal{I}'}, \cdot^{\mathcal{I}'})$ with root y_0 such that $y_0 \in (\mathsf{val}_r(C) \sqcap C')^{\mathcal{I}'}$ and $y_0 \notin \mathsf{val}_r(D)^{\mathcal{I}'}$. Define $\mathcal{J} = (\Delta_{\mathcal{J}}, \cdot^{\mathcal{J}})$ with

- $\Delta_{\mathcal{J}} := \Delta_{\mathcal{I}} \cup \Delta_{\mathcal{I}'}$;
- $P^{\mathcal{J}} := P^{\mathcal{I}} \cup P^{\mathcal{I}'}$ for all $P \in N_C$;
- $r^{\mathcal{J}} := (r^{\mathcal{I}} \setminus \{(x_0, y)\}) \cup r^{\mathcal{I}'} \cup \{(x_0, y_0)\}$.

Then $\mathcal{J}$ is a model of C with $x_0 \in C^{\mathcal{J}}$. But since the r-successor y_0 of x_0 in $\mathcal{J}$ is not an instance of $\mathsf{val}_r(D)^{\mathcal{J}}$, it follows $x_0 \notin D^{\mathcal{J}}$ in contradiction to $C \sqsubseteq D$. Thus, $\mathsf{val}_r(C) \sqcap C' \sqsubseteq \mathsf{val}_r(D)$ for all $C' \in \mathsf{exr}_r(C)^*$. $\square$

For the sake of completeness, we should mention that, as shown by Hemaspaandra, [7], subsumption checking in $\mathcal{ALEN}$ is PSPACE-complete. However, our characterization is not intended to yield a PSPACE-algorithm. It rather provides a formal basis for the lcs algorithm presented in the next section.

5 Computing the LCS in $\mathcal{ALEN}$

We now present a recursive algorithm computing the lcs of two $\mathcal{ALEN}$-concept descriptions C and D. It is depicted in Figure 1 and called c-lcs. Although, the algorithm is presented as binary operation (working on C and D), it can be generalized to be applicable to (arbitrary) sets of concept description in the obvious way: $\text{c-lcs}(C_1, \ldots, C_n) := \text{c-lcs}(C_1, \text{c-lcs}(C_2, \cdots, \text{c-lcs}(C_{n-1}, C_n) \cdots))$. In fact, within the algorithm this is used. Since the maximum role depth of the concept descriptions occurring in the recursive invocations of the algorithm decreases, c-lcs always terminates. The following theorem states correctness of the lcs algorithm depicted in Figure 1. Its proof is by induction on the maximum role depth of the input concept descriptions and makes heavy use of the structural characterization of subsumption given in Theorem 2. As an immediate consequence, we obtain that the lcs of two $\mathcal{ALEN}$-concept descriptions always exists.

Theorem 5 *Let C, D be $\mathcal{ALEN}$-concept descriptions. Then,* $\text{c-lcs}(C, D) \equiv \text{lcs}(C, D)$.

Before we prove Theorem 5, we illustrate the definition of $\text{c-lcs}(C, D)$ in case that $C \not\sqsubseteq D$ and $D \not\sqsubseteq C$ (since the special cases $C \sqsubseteq D$, $D \sqsubseteq C$ are trivial), using our running example introduced in Section 3. The conjuncts occurring on the top-level of $\text{c-lcs}(C, D)$ can, as before, be divided into three parts, namely (1) (negated) concept names and number restrictions, (2) the existential restrictions, and (3) the value restriction. These conjuncts are defined in such a way that

(a) the conditions 1.–5. in Theorem 2 for $C \sqsubseteq \text{c-lcs}(C, D)$ and $D \sqsubseteq \text{c-lcs}(C, D)$ are satisfied, and

(b) $\text{c-lcs}(C, D)$ is the least concept description (w.r.t. $\sqsubseteq$) satisfying (a).

For the conditions 1.–3., this is quite obvious, since, for $E \in \{C, D\}$, we obtain

- $\text{prim}(\text{c-lcs}(C, D)) = \text{prim}(C) \cap \text{prim}(D) \subseteq \text{prim}(E)$,
- $\text{min}_r(\text{c-lcs}(C, D)) = min\{\text{min}_r(C), \text{min}_r(D)\} \leq \text{min}_r(E)$, and
- $\text{max}_r(\text{c-lcs}(C, D)) = max\{\text{max}_r(C), \text{max}_r(D)\} \geq \text{max}_r(E)$.

In our running example, we obtain $\text{prim}(\text{c-lcs}(C_{ex}, D_{ex})) = \emptyset$, $\text{min}_r(\text{c-lcs}(C_{ex}, D_{ex})) = 2$, as well as $\text{max}_r(\text{c-lcs}(C_{ex}, D_{ex})) = \infty$.

Things are more complicated for existential and value restrictions. Let us first consider the definition of $\text{exr}_r(\text{c-lcs}(C, D))$, i.e., the existential restrictions obtained from the sets $\mathcal{C}_r(C)$ and $\mathcal{C}_r(D)$. Roughly speaking, if $\Gamma_r(E) \neq \emptyset$, then $\mathcal{C}_r(E)$ contains all (minimal) concept descriptions occurring in an existential restriction induced by E for $E \in \{C, D\}$. Each such concept description is obtained as the recursively computed lcs of a set of concept descriptions consisting of one concept description from $\text{exr}_r(E)^\alpha$ (conjoined with $\text{val}_r(E)$) for each $\alpha \in \Gamma_r(E)$. In our running example, each pair of concept descriptions occurring in the existential restrictions on the top-level of C_{ex}^1 and C_{ex}^2, respectively, yields such an lcs, and thus $\mathcal{C}_r(C_{ex}) = \{P \sqcap A_1 \sqcap A_2 \sqcap A_3, Q \sqcap A_1 \sqcap A_3, \neg Q \sqcap A_1 \sqcap A_3, \neg P \sqcap A_1 \sqcap A_3\}$ (see Section 3).

If $\text{min}_r(E) = 1$ and $\Gamma_r(E) = \emptyset$, we set $\mathcal{C}_r(E) = \{\text{val}_r(E)\}$ since then $\text{exr}_r(E) = \emptyset$, and $\exists r.\text{val}_r(E)$ is the unique minimal existential restriction induced by E. This case is illustrated by D_{ex} in our running example, where $\text{min}_r(D_{ex}) = 3$, $\text{exr}_r(D_{ex}) = \emptyset$, and $\mathcal{C}_r(D_{ex}) = \{A_1 \sqcap A_2 \sqcap A_3\}$.

Finally, if $\text{min}_r(E) = 0$, i.e., there exists no existential restriction subsuming E, then obviously, no existential restriction can occur on the top-level of a common subsumer of C and D. Therefore, we set $\mathcal{C}_r(E) := \emptyset$.

Given $\mathcal{C}_r(C)$ and $\mathcal{C}_r(D)$, the lcs of each pair $C' \in \mathcal{C}_r(C)$ and $D' \in \mathcal{C}_r(D)$ gives rise to an existential restriction on the top-level of the lcs of C and D. In our example, we obtain $\mathsf{exr}_r(\mathsf{c\text{-}lcs}(C_{ex}, D_{ex})) = \{A_1 \sqcap A_2 \sqcap A_3, A_1 \sqcap A_3\}$, giving rise to the existential restrictions $\exists r.(A_1 \sqcap A_2 \sqcap A_3)$ and $\exists r.(A_1 \sqcap A_3)$ (where the latter one can be omitted).

It remains to comment on $\forall r.\mathsf{c\text{-}lcs}(C_r^*, D_r^*)$. Intuitively, $\forall r.E_r^*$ is the most specific value restriction subsuming E for $E \in \{C, D\}$. (Thus, $\forall r.\mathsf{c\text{-}lcs}(C_r^*, D_r^*)$ is the most specific value restriction subsuming both C and D.) If E has no induced value restriction, E_r^* coincides with $\mathsf{val}_r(E)$. This is the case for D_{ex}, where $(D_{ex})_r^* = \mathsf{val}_r(D_{ex}) = A_1 \sqcap A_2 \sqcap A_3$. If $\mathsf{max}_r(E) = 0$, the fact that $E \sqsubseteq \forall r.\bot$ is made explicit by defining $E_r^* := \bot$. Finally, if $0 < \kappa_r(E) = \mathsf{max}_r(E)$, then the induced value restriction is again made explicit by recursively computing the lcs of the merged existential restrictions (each conjoined with $\mathsf{val}_r(E)$). In our running example, this case is illustrated by C_{ex}: there, $\mathsf{val}_r(C_{ex}) = \top$, but since $\kappa_r(C_{ex}) = \mathsf{max}_r(C_{ex}) = 2$, we obtain $(C_{ex})_r^* = A_1 \sqcap A_3$ which is the recursively computed lcs of $\mathsf{exr}_r(C_{ex})^* = \mathsf{exr}_r(C_{ex}^1) \cup \mathsf{exr}_r(C_{ex}^2)$ (see Section 3).

Proof of Theorem 5: It is sufficient to show that

i) $C \sqsubseteq \mathsf{c\text{-}lcs}(C, D)$ and $D \sqsubseteq \mathsf{c\text{-}lcs}(C, D)$, and

ii) for all E with $C, D \sqsubseteq E$, it follows $\mathsf{c\text{-}lcs}(C, D) \sqsubseteq E$.

Proof by induction on $max\{\mathsf{depth}(C), \mathsf{depth}(D)\}$. Let $L := \mathsf{c\text{-}lcs}(C, D)$.

Ad i): Obviously, it is sufficient to show $C \sqsubseteq \mathsf{c\text{-}lcs}(C, D)$. If $C \sqsubseteq D$, $C \equiv \bot$ or $L \equiv \top$, nothing has to be shown. Assume $C \not\equiv \bot$ and $L \not\equiv \top$. We show that C and L satisfy the conditions (1)–(5) in Theorem 2.

By definition of L,

(1) $\mathsf{prim}(L) \subseteq \mathsf{prim}(C)$,

(2) $\mathsf{min}_r(C) \geq \mathsf{min}_r(L)$, and

(3) $\mathsf{max}_r(C) \leq \mathsf{max}_r(L)$.

To show (5), assume $\mathsf{val}_r(L) \not\equiv \top$. If $\mathsf{max}_r(C) = 0$, nothing has to be shown. If $\kappa_r(C) < \mathsf{max}_r(C)$, then $C_r^* = \mathsf{val}_r(C)$. By induction, $\mathsf{val}_r(C) \sqsubseteq \mathsf{val}_r(L)$. Assume $0 < \kappa_r(C) = \mathsf{max}_r(C)$. Then, $C_r^* = \mathsf{c\text{-}lcs}(\{\mathsf{val}_r(C) \sqcap C' \mid C' \in \mathsf{exr}_r(C)^*\})$. By induction, $\mathsf{val}_r(C) \sqcap C' \sqsubseteq C_r^*$ for all $C' \in \mathsf{exr}_r(C)^*$, as well as $C_r^* \sqsubseteq \mathsf{val}_r(L) = \mathsf{c\text{-}lcs}(C_r^*.D_r^*)$. This implies $\mathsf{val}_r(C) \sqcap C' \sqsubseteq \mathsf{val}_r(L)$ for all $C' \in \mathsf{exr}_r(C)^*$.

It remains to show (4). Let $L' \in \mathsf{exr}_r(L)$. This implies $\mathcal{C}_r \neq \emptyset$ and $\mathcal{D}_r \neq \emptyset$.

Assume $\mathsf{exr}_r(C) = \emptyset$. Then $\mathcal{C}_r \neq \emptyset$ implies $\mathcal{C}_r = \{\{\mathsf{val}_r(C)\}\}$ and hence, $\mathsf{min}_r(C) \geq 1$. By induction, $\mathsf{val}_r(C) \sqsubseteq L'$.

Assume $\mathsf{exr}_r(C) \neq \emptyset$. Let L' be obtained from $\{C_1 \sqcap \mathsf{val}_r(C), \ldots, C_k \sqcap \mathsf{val}_r(C)\} \in \mathcal{C}_r$ and some $\mathcal{D} \in \mathcal{D}_r$. Then, for each $\alpha \in \Gamma_r(C)$, there exists $i \in \{1, \ldots, k\}$ such that $C_i \in \mathsf{exr}_r(C)^\alpha$. By induction, $C_i \sqcap \mathsf{val}_r(C) \sqsubseteq L'$.

Thus, we have shown $C \sqsubseteq L$.

Ad ii): Let E be an $\mathcal{ALEN}$-concept description with $C, D \sqsubseteq E$. If $C \sqsubseteq D$ or $D \sqsubseteq C$, we get $\mathsf{c\text{-}lcs}(C, D) \sqsubseteq E$. Assume $C \not\sqsubseteq D$ and $D \not\sqsubseteq C$. In particular, this implies $C \not\equiv \bot$ and $D \not\equiv \bot$, and hence, $L \not\equiv \bot$.

If $E \equiv \top$, nothing has to be shown. Assume $E \not\equiv \top$. We show that L and E satisfy the conditions (1)–(5) in Theorem 2. By assumption, $C \not\equiv \bot$, $D \not\equiv \bot$, and $E \not\equiv \top$. Thus, by Theorem 2, $C \sqsubseteq E$

$(D \sqsubseteq E)$ implies that C (D) and E satisfy the conditions (1)–(5) in Theorem 2. In the following, we will use this fact without referring to Theorem 2 each time.

Ad (1): Since $\mathsf{prim}(E) \subseteq \mathsf{prim}(C)$ and $\mathsf{prim}(E) \subseteq \mathsf{prim}(D)$, it follows $\mathsf{prim}(E) \subseteq \mathsf{prim}(C) \cap \mathsf{prim}(D) = \mathsf{prim}(L)$.

Ad (2): Since $\mathsf{max}_r(C) \leq \mathsf{max}_r(E)$ and $\mathsf{max}_r(D) \leq \mathsf{max}_r(E)$, it follows $\mathsf{max}_r(L) = min\{\mathsf{max}_r(C), \mathsf{max}_r(D)\} \leq \mathsf{max}_r(E)$.

Ad (3): Since $\mathsf{min}_r(C) \geq \mathsf{min}_r(E)$ and $\mathsf{min}_r(D) \geq \mathsf{min}_r(E)$, it follows $\mathsf{min}_r(L) = max\{\mathsf{min}_r(C), \mathsf{min}_r(D)\} \geq \mathsf{min}_r(E)$.

Ad (5): Assume $\mathsf{val}_r(E) \not\equiv \top$. If $\mathsf{max}_r(L) = 0$, nothing has to be shown. According to condition (5) we have to distinguish two cases:

1. Assume $\kappa_r(L) < \mathsf{max}_r(L)$. We have to show $\mathsf{val}_r(L) \sqsubseteq \mathsf{val}_r(E)$. We show $C_r^* \sqsubseteq \mathsf{val}_r(E)$. Analogously, we can show $D_r^* \sqsubseteq \mathsf{val}_r(E)$. It also holds that $\mathsf{depth}(C_r^*) < \mathsf{depth}(C)$, and analogously, $\mathsf{depth}(D_r^*) < \mathsf{depth}(D)$: For the cases where $0 < \kappa_r(C) < \mathsf{max}(C)$ or $\mathsf{max}_r(C) = 0$ this is obvious. For $0 < \kappa_r(C) = \mathsf{max}_r(C)$, we use that $\mathsf{depth}(\mathsf{c\text{-}lcs}\{E_1, \ldots, E_k\}) \leq max\{\mathsf{depth}(E_i) \mid 1 \leq i \leq n\}$. This easily follows by an inductive argument from the definition of c-lcs. Consequently, we can then apply the induction hypothesis on $\mathsf{c\text{-}lcs}(C_r^*, D_r^*)$ and obtain $\mathsf{val}_r(L) = \mathsf{c\text{-}lcs}(C_r^*, D_r^*) \sqsubseteq \mathsf{val}_r(E)$.

   It remains to show $C_r^* \sqsubseteq \mathsf{val}_r(E)$.

   Assume $\mathsf{max}_r(C) = 0$. Then $C_r^* = \bot$ and hence, $C_r^* \sqsubseteq \mathsf{val}_r(E)$.

   Assume $\kappa_r(C) < \mathsf{max}_r(C)$. Then $\mathsf{val}_r(C) \sqsubseteq \mathsf{val}_r(E)$ and, since $C_r^* = \mathsf{val}_r(C)$, we obtain $C_r^* \sqsubseteq \mathsf{val}_r(E)$.

   Finally, assume $0 < \kappa_r(C) = \mathsf{max}_r(C)$. Then $\mathsf{val}_r(C) \sqcap C' \sqsubseteq \mathsf{val}_r(E)$ for all $C' \in \mathsf{exr}_r(C)^*$. By induction, $C_r^* = \mathsf{c\text{-}lcs}(\{\mathsf{val}_r(C) \sqcap C' \mid C' \in \mathsf{exr}_r(C)^*\}) \sqsubseteq \mathsf{val}_r(E)$.

2. Assume $0 < \kappa_r(L) = \mathsf{max}_r(L)$. We have to show $\mathsf{val}_r(L) \sqcap L' \sqsubseteq \mathsf{val}_r(E)$ for all $L' \in \mathsf{exr}_r(L)^*$. As in the previous case we show $C_r^* \sqsubseteq \mathsf{val}_r(E)$. Analogously, we also obtain $D_r^* \sqsubseteq \mathsf{val}_r(E)$. Then again, by induction we can conclude $\mathsf{val}_r(L) = \mathsf{c\text{-}lcs}(C_r^*, D_r^*) \sqsubseteq \mathsf{val}_r(E)$. In particular, $\mathsf{val}_r(L) \sqcap L' \sqsubseteq \mathsf{val}_r(E)$.

   Note that $\kappa_r(L) > 0$ implies $\mathsf{exr}_r(L) \neq \emptyset$, and thus, $\mathcal{C}_r(C) \neq \emptyset$.

   Let us first assume $\mathsf{exr}_r(C) = \emptyset$. Together with $\mathcal{C}_r(C) \neq \emptyset$, we know $\mathsf{min}_r(C) \geq 1$. Because $C \not\equiv \bot$ it also follows $\mathsf{max}_r(C) \geq \mathsf{min}_r(C) \geq 1$. Moreover, $\mathsf{exr}_r(C) = \emptyset$ yields $\kappa_r(C) = 0$. In particular, $\kappa_r(C) < \mathsf{max}_r(C)$. Then, $\mathsf{val}_r(E) \not\equiv \top$ and $C \sqsubseteq E$ imply $\mathsf{val}_r(C) \sqsubseteq \mathsf{val}_r(E)$. Since $C_r^* = \mathsf{val}_r(C)$, it follows $C_r^* \sqsubseteq \mathsf{val}_r(E)$.

   We now assume $\mathsf{exr}_r(C) \neq \emptyset$. Then, for all $L' \in \mathsf{exr}_r(L)$, there exist a tuple $(C_1, \ldots, C_k) \in \otimes_{\alpha \in \Gamma_r(C)} \mathsf{exr}_r(C)^\alpha$ such that L' is obtained from this tuple, which implies $C_i \sqcap \mathsf{val}_r(C) \sqsubseteq L'$ for all $1 \leq i \leq k$. Since there exist $\kappa_r(L)$ disjoint existential restrictions on the top-level of L, these subsumption relationships imply that there also exist at least $\kappa_r(L)$ disjoint existential restrictions on the top-level of C. In particular, $\kappa_r(C) \geq \kappa_r(L)$. Since C is consistent, it is $\mathsf{max}_r(C) \geq \kappa_r(C)$. By definition, it is $\mathsf{max}_r(L) \geq \mathsf{max}_r(C)$. Thus, $\mathsf{max}_r(C) \geq \kappa_r(C) \geq \kappa_r(L) = \mathsf{max}_r(L) \geq \mathsf{max}_r(C)$, i.e., $\mathsf{max}_r(C) = \kappa_r(C)$. Now, $C \sqsubseteq E$ implies $C' \sqcap \mathsf{val}_r(C) \sqsubseteq \mathsf{val}_r(E)$ for all $C' \in \mathsf{exr}_r(C)^*$. By induction, it follows $C_r^* \sqsubseteq \mathsf{val}_r(E)$.

Ad (4): Let $E' \in \mathsf{exr}_r(E)$.

Assume $\mathsf{exr}_r(L) = \emptyset$. This implies $\mathcal{C}_r = \emptyset$ or $\mathcal{D}_r = \emptyset$. W.l.o.g. let $\mathcal{C}_r = \emptyset$. This would imply $\mathsf{min}_r(C) = 0$ and $\mathsf{exr}_r(C) = \emptyset$ in contradiction to $C \sqsubseteq E$. Thus, it is $\mathsf{exr}_r(L) \neq \emptyset$ and we have to show that, for each $\gamma \in \Gamma_r(L)$, there exists $L' \in \mathsf{exr}_r(L)^\gamma$ with $L' \sqcap \mathsf{val}_r(L) \sqsubseteq E'$.

Since $C \sqsubseteq E$, it holds

(1) $\mathsf{exr}_r(C) = \emptyset$, $\mathsf{min}_r(C) \geq 1$, and $\mathsf{val}_r(C) \sqsubseteq E'$; or

(2) for each $\alpha \in \Gamma_r(C)$, there exists $C' \in \mathsf{exr}_r(C)^\alpha$ with $C' \sqcap \mathsf{val}_r(C) \sqsubseteq E'$;

and since $D \sqsubseteq E$, it holds

(1)' $\mathsf{exr}_r(D) = \emptyset$, $\mathsf{min}_r(D) \geq 1$, and $\mathsf{val}_r(D) \sqsubseteq E'$; or

(2)' for each $\beta \in \Gamma_r(D)$, there exists $D' \in \mathsf{exr}_r(D)^\beta$ with $D' \sqcap \mathsf{val}_r(D) \sqsubseteq E'$.

We have to consider the possible combinations of cases from $\{(1), (2)\}$ and $\{(1)', (2)'\}$.

Assume (1) and (1)': Then $C_r^* = \mathsf{val}_r(C)$ and $D_r^* = \mathsf{val}_r(D)$, i.e., $\mathsf{val}_r(L) = \mathsf{c\text{-}lcs}(\mathsf{val}_r(C), \mathsf{val}_r(D))$. By induction, we get $\mathsf{val}_r(L) \sqsubseteq E'$ and hence, for all $\gamma \in \Gamma_r(L)$, we can choose an arbitrary $L' \in \mathsf{exr}_r(L)^\gamma$ and get $L' \sqcap \mathsf{val}_r(L) \sqsubseteq E'$.[2]

Assume (1) and (2)': Then there exists $(D_1, \ldots, D_\ell) \in \otimes_{\beta \in \Gamma_r(D)} \mathsf{exr}_r(D)^\beta$ such that $D_j \sqcap \mathsf{val}_r(D) \sqsubseteq E'$ for all $1 \leq j \leq \ell$, and

$$L_0 := \mathsf{c\text{-}lcs}(\mathsf{val}_r(C), \mathsf{c\text{-}lcs}(\{D_j \sqcap \mathsf{val}_r(D) \mid 1 \leq j \leq \ell\})) \in \mathsf{exr}_r(L).$$

By induction, $L_0 \sqsubseteq E'$.

Now, for each $\gamma \in \Gamma_r(L)$, there exists $L' \in \mathsf{exr}_r(L)^\gamma$ such that $L' \sqsubseteq L_0 \sqsubseteq E'$; in particular, $L' \sqcap \mathsf{val}_r(L) \sqsubseteq E'$.

Assume (2) and (1)': Analogously to the previous case.

Assume (2) and (2)': Again, there exists $(D_1, \ldots, D_\ell) \in \otimes_{\beta \in \Gamma_r(D)} \mathsf{exr}_r(D)^\beta$ such that $D_j \sqcap \mathsf{val}_r(D) \sqsubseteq E'$ for all $1 \leq j \leq \ell$. In addition, there exists $(C_1, \ldots, C_k) \in \otimes_{\alpha \in \Gamma_r(C)} \mathsf{exr}_r(C)^\alpha$ such that $C_i \sqcap \mathsf{val}_r(C) \sqsubseteq E'$ for all $1 \leq i \leq k$. Then,

$$L_0 := \mathsf{c\text{-}lcs}(\mathsf{c\text{-}lcs}(\{C_i \sqcap \mathsf{val}_r(C) \mid 1 \leq i \leq k\}), \mathsf{c\text{-}lcs}(\{D_j \sqcap \mathsf{val}_r(D) \mid 1 \leq j \leq \ell\})) \in \mathsf{exr}_r(L).$$

By induction, $L_0 \sqsubseteq E'$. As above, we get that, for each $\gamma \in \Gamma_r(L)$, there exists $L' \in \mathsf{exr}_r(L)^\gamma$ such that $L' \sqsubseteq L_0$ and hence, $L' \sqcap \mathsf{val}_r(L) \sqsubseteq E'$.

This completes the proof of Theorem 5. □

[2] Note that if (1) and (1)$'$ holds, then $\Gamma_r(C) = \Gamma_r(D) = \emptyset$; hence, $\mathcal{C}_r(C) = \{\mathsf{val}_r(C)\}$, $\mathcal{C}_r(D) = \{\mathsf{val}_r(D)\}$, and $\mathsf{exr}_r(L) = \{\mathsf{c\text{-}lcs}(\mathsf{val}_r(C), \mathsf{val}_r(D))\}$.

Complexity of the lcs algorithm: For the complexity analysis of our lcs algorithm, the algorithm is slightly modified: When computing $\mathsf{c\text{-}lcs}(C', D')$ (see the definition of $\mathsf{c\text{-}lcs}(C, D)$ in Figure 1), C', and analogously D', might be the result of applying c-lcs to the set $S := \{C_1 \sqcap \mathsf{val}_r(C), \ldots, C_{n(C)} \sqcap \mathsf{val}_r(C)\}$ as specified in $\mathcal{C}_r(C)$. However, using the fact that $lcs(lcs(C, D), E) \equiv lcs(C, D, E)$, we can omit the application of c-lcs to S. Thus, instead of $C' = \mathsf{c\text{-}lcs}(S)$, we simply take the set S. In particular, when computing $\mathsf{c\text{-}lcs}(C', D')$, c-lcs is merely applied to a set of subdescriptions of C and D. The same argument works for the concept description C_r^* (and D_r^*). Instead of $C_r^* = \mathsf{c\text{-}lcs}(\{\mathsf{val}_r(C) \sqcap C' \mid C' \in \mathsf{exr}_r(C)^*\})$, we just take the set $\{\mathsf{val}_r(C) \sqcap C' \mid C' \in \mathsf{exr}_r(C)^*\}$. Then again, when computing $\mathsf{c\text{-}lcs}(C_r^*, D_r^*)$, c-lcs is only applied to subdescriptions of C and D, or $\bot$. Throughout the remainder of this section, whenever we refer to c-lcs, we mean the algorithm modified as just explained.

We will show that c-lcs runs in double exponential time in the size of the input concept descriptions. We first investigate the size of the sets, involved in the computation.

Let E be an $\mathcal{ALEN}$-concept description. It is clear that $|\Gamma_r(E)|$ can be bounded exponentially in the size of E; $|\mathsf{exr}_r(E)^\alpha|$ as well as the size of the concept descriptions in $\mathsf{exr}_r(E)^\alpha$ are bounded by the size of E; $|\mathsf{exr}_r(E)^*|$ can be bounded exponentially in the size of E; finally, $|\mathcal{C}_r(E)|$ has an upper bound double exponential in the size of E.

Now, we look at $\mathsf{c\text{-}lcs}(C, D)$ when recursively expanding its definition. Such an expansion can be viewed as a labeled tree, where every node is labeled by the conjunction of (negated) concept names and number restriction, and the edges of the tree are labeled with $\exists r$ and $\forall r$. (Note that every node has exactly one edge labeled $\forall r$, but there may be several edges labeled $\exists r$.) Clearly, the depth of the tree is bounded by $m := max\{\mathsf{depth}(C), \mathsf{depth}(D)\}$.

It is obvious that on every level, c-lcs is only applied to subdescriptions of C and D of the form $\mathsf{val}_r(E)$, $E \sqcap \mathsf{val}_r(E)$, or $\bot$.

Let us consider the number of these concept descriptions, i.e., the number of arguments c-lcs is applied to on each level. We first concentrate on the invocation of c-lcs for computing existential restrictions. It is sufficient to only consider the case where $\Gamma_r(E) \neq \emptyset$, since this produces the largest number of concept descriptions. We know that the cardinality of the sets $C' \in \mathcal{C}_r(C)$ and $D' \in \mathcal{C}_r(D)$ is bounded exponentially in the size of C and D. (Recall that we work with sets S instead of $\mathsf{c\text{-}lcs}(S)$.) Thus, on the first level, for computing $\mathsf{c\text{-}lcs}(C', D')$, c-lcs is only applied to an exponential number of concept descriptions. But then, on the second level, an exponential number of sets $\mathcal{C}_r(E)$ need to be considered, where E is of the form explained above. Every element in $\mathcal{C}_r(E)$ might be a set of exponential cardinality. Hence, the number of arguments of c-lcs on the second level is the product of two exponential functions, thus still exponential. Analogously, on the third level, we obtain a product of three exponential functions. Iterating this argument, and since the depth of the tree is bounded by m, the number of arguments c-lcs is applied is exponential on every level of the tree. A similar, even simpler argument, works for the invocations of c-lcs for value restrictions.

Summing up, so far, we know that every invocation of c-lcs is applied to a set of concept descriptions with cardinality exponentially bounded in the size of C and D. Moreover, every concept description in such a set is a subdescription of C or D, or $\bot$. Thus, for every node of the tree, the (negated) concept names as well as the number restrictions can be computed in exponential time.

It remains to count the number of nodes of the tree. Of course, it suffices to only consider those nodes reached by edges labeled with $\exists r$, since every node of the tree has only one outgoing edge labeled $\forall r$. On the first level, since the cardinality of $\mathcal{C}_r(C)$ and $\mathcal{C}_r(D)$ is bounded double exponentially in the size of C and D, the number of outgoing edges from the root is at most double exponential. As explained above, on every other level, the number of sets $\mathcal{C}_r(E)$ to be considered is bounded exponentially in the size of C and D. Since the cardinality of all these sets can be bounded double exponentially, the outdegree of every node of the tree can be bounded double exponentially as well. Thus, since the depth of the tree is bounded by m, the number of its nodes is at most double exponential in the size of C and D.

Altogether, we have shown the following upper bound for computing the lcs in $\mathcal{ALEN}$.

Corollary 6 *The lcs of (a set of) $\mathcal{ALEN}$-concept descriptions can be computed in double exponential time.*

It is an open problem whether there also exists an exponential time algorithm, or whether the double exponential blow-up is unavoidable.

6 Conclusion and future work

We have presented an algorithm for computing the lcs in $\mathcal{ALEN}$. Its proof of correctness is based on the structural characterization of subsumption introduced in Section 4. In this characterization, we avoided to explicitly use the lcs in order to be able to decouple the characterization of subsumption and the lcs computation. Interleaving these two tasks caused previous work on the lcs in $\mathcal{ALEN}$ to be incorrect. Still, due to the interaction between number restrictions and existential restrictions the characterization of subsumption and the lcs computation became much more involved than for the sublanguages $\mathcal{ALE}$ [2] and $\mathcal{ALN}$ [5, 1].

For the size of the lcs of two $\mathcal{ALE}$-concept descriptions a tight exponential lower bound was shown in [2]. The same argument yields an exponential lower bound for $\mathcal{ALEN}$-concept descriptions. However, as yet it is not known whether this lower bound is tight. Nevertheless, since our lcs algorithm runs in double exponential time, we at least obtain a double exponential upper bound for the size of the lcs.

Since a prototype implementation of the exponential-time lcs algorithm for $\mathcal{ALE}$ behaves quite well in the chemical process engineering application [3], we believe that the lcs algorithm for $\mathcal{ALEN}$ proposed here will also work in realistic application situations. For evaluation, continuing the work on $\mathcal{ALE}$, a prototype of the algorithm for $\mathcal{ALEN}$ will be implemented using the DL-system FaCT [9] for deciding subsumption in $\mathcal{ALEN}$, which then is to be applied to the chemical engineering knowledge base.

References

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

[2] F. Baader, R. Küsters, and R. Molitor. Computing least common subsumers in description logics with existential restrictions. In T. Dean, editor, *Proceedings of the 16th International Joint Conference on Artificial Intelligence 1999 (IJCAI'99)*, pages 96–101. Morgan Kaufmann, 1999.

[3] F. Baader and R. Molitor. Building and structuring description logic knowledge bases using least common subsumers and concept analysis. In B. Ganter and G. Mineau, editors, *Proceedings of the 8th International Conference on Conceptual Structures (ICCS2000)*, volume 1867 of *Lecture Notes in Artificial Intelligence*, pages 292–305. Springer-Verlag, 2000.

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

[5] W.W. Cohen and H. Hirsh. Learning the CLASSIC description logic: Theoretical and experimental results. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 4th International Conference (KR'94)*, pages 121–132. Morgan Kaufmann, 1994.

[6] F.M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in description logics. In G. Brewka, editor, *Foundation of Knowledge Representation*, pages 191–236. CSLI-Publications, 1996.

[7] E. Hemaspaandra. The complexity of pure man's logic. In *Essays dedicated to Johan van Benthem on the occasion of his 50th birthday*, 1999.

[8] B. Hollunder and F. Baader. Qualifying number restrictions in concept languages. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 2nd International Conference (KR'91)*, pages 335–346. Morgan Kaufmann, 1991.

[9] I. Horrocks. Using an expressive description logic: FaCT or fiction? In A.G. Cohn, L. Schubert, and S.C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of 6th International Conference (KR'98)*, pages 636–647. Morgan Kaufmann, 1998.

[10] T. Mantay. Computing least common subsumers in expressive description logics. In N. Foo, editor, *Advanced Topis in Artificial Intelligence: Proceedings of the 12th Australian Joint Conference on Artificial Intelligence (AI'99)*, volume 1747 of *Lecture Notes in Artificial Intelligence*, pages 218–230. Springer-Verlag, 1999.

[11] R. Möller, V. Haaslev, and B. Neumann. Semantics-based information retrieval. In *Proceedings of the International Conference on Information Technology and Knowledge Systems (IT&KNOWS'98)*, 1998.

[12] U. Sattler. *Terminological knowledge representation systems in a process engineering application*. PhD thesis, RWTH Aachen, 1998. Available at http://www-lti.informatik.rwth-aachen.de/Forschung/Papers.html.

[13] L. von Wedel and W. Marquardt. ROME: A repository to support the integration of models over the lifecycle of model-based engineering processes. In *Proceedings of ESCAPE-10*, 2000.