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

**Ralf Küsters**
Institut für Informatik und Praktische Mathematik
Christian-Albrechts-Universität zu Kiel
Germany
kuesters@ti.informatik.uni-kiel.de

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

## Abstract

Computing the least common subsumer (lcs) has proved to be useful in a variety of different applications. 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, thereby correcting previous algorithms proposed in the literature.

## 1 Introduction

Computing the least common subsumer (lcs) in description logics (DLs) is an inference task first introduced by Cohen, Borgida, and Hirsh [Cohen *et al.*, 1992] for sublanguages of CLASSIC. Since then it has found several applications: as a key operation in inductive learning algorithms [Cohen and Hirsh, 1994], as a means to measure the similarity of concepts for information retrieval [Möller *et al.*, 1998], and as an operation to support the bottom-up construction of DL-knowledge bases [Baader and Küsters, 1998; Baader *et al.*, 1999]. 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 [Cohen *et al.*, 1992; Cohen and Hirsh, 1994]. More recently, motivated by the bottom-up construction of knowledge bases in a chemical engineering application [Sattler, 1998; von Wedel and Marquardt, 2000], the lcs has been investigated for the DL $\mathcal{ALE}$ [Baader *et al.*, 1999], which allows for existential restrictions instead of number restrictions. Although first empirical results are encouraging [Baader and Molitor, 2000], 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 [Cohen *et al.*, 1992; Baader and Küsters, 1998; Baader *et al.*, 1999], 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 algorithm needs to be interleaved with the proof of soundness and completeness of the characterization of subsumption, making the proofs quite involved. In [Mantay, 1999], this approach is in fact pursued. However, in an attempt to avoid these interleaving proofs, Mantay 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

*This work was carried out while the author was still at the LuFG Theoretical Computer Science, RWTH Aachen.

| 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 for $r \in N_R$ | $\exists r.C$ | $\{x \in \Delta_\mathcal{I} \mid \exists y : (x,y) \in r^\mathcal{I} \wedge y \in C^\mathcal{I}\}$ |
| value restrictions for $r \in N_R$ | $\forall r.C$ | $\{x \in \Delta_\mathcal{I} \mid \forall y : (x,y) \in r^\mathcal{I} \rightarrow y \in C^\mathcal{I}\}$ |
| number restrictions for $r \in N_R$, $n \in \mathbb{N}$ | $(\geq nr)$ | $\{x \in \Delta_\mathcal{I} \mid \#\{y : (x,y) \in r^\mathcal{I}\} \geq n\}$ |
| | $(\leq nr)$ | $\{x \in \Delta_\mathcal{I} \mid \#\{y : (x,y) \in r^\mathcal{I}\} \leq n\}$ |

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

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. Due to space limitations, we cannot give all technical details. These details as well as complete proofs can be found in [Küsters and Molitor, 2000].

## 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*.

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 $\mathcal{ALEN}$-concept descriptions $C_1, \ldots, C_n$, $n \geq 2$, 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 descriptions 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:

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

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, $P \sqcap A_1 \sqcap A_2 \sqcap A_3 \equiv lcs(P \sqcap Q \sqcap A_1 \sqcap A_2 \sqcap A_3, P \sqcap \neg Q \sqcap A_1 \sqcap A_2 \sqcap A_3)$. However, as explained below, our characterization of subsumption avoids to explicitly use the lcs by employing the fact that

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

**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 $E_{ex} := (\leq 2r) \sqcap \exists r.(A_1 \sqcap A_2) \sqcap \exists r.(A_1 \sqcap A_3)$, 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, $A_1 \sqcap A_3 \equiv lcs(P \sqcap Q \sqcap A_1 \sqcap A_2 \sqcap A_3, \neg P \sqcap \neg Q \sqcap A_1 \sqcap A_2 \sqcap A_3, P \sqcap \neg Q \sqcap A_1 \sqcap A_2 \sqcap A_3, \neg P \sqcap Q \sqcap A_1 \sqcap A_2 \sqcap A_3)$. Again, using the equivalence (1), we avoid to explicitly compute the lcs in the characterization of subsumption.

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

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

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

## 4 Structural characterization of subsumption

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 $C$;
- $\mathsf{min}_r(C) := max\{k \mid C \sqsubseteq (\geq k\, r)\}$;
- $\mathsf{max}_r(C) := min\{k \mid C \sqsubseteq (\leq k\, r)\}$;
- 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, $\mathsf{val}_r(C) := \top$;
- $\mathsf{exr}_r(C) := \{C' \mid \text{there exists } \exists r.C' \text{ on the top-level of } C\}$.

Note that $\mathsf{min}_r(C)$ is always finite, whereas $\mathsf{max}_r(C)$ may be $\infty$, namely if there exists no positive integer $k$ such that $C \sqsubseteq (\leq k\, r)$. Although these values 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 [Küsters and Molitor, 2000]. 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 $C \sqsubseteq D$.

[1] The structural subsumption algorithm introduced in [Mantay, 1999], 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 [Mantay, 1999] is incorrect: although the lcs of $E_{ex}$ and $\forall r.A_1$ is $\top$, the algorithm returns $\forall r.A_1$.

**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 $\alpha$ 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. $\bigsqcap_{j \in \alpha(i)} C_j \sqcap \mathsf{val}_r(C) \not\equiv \bot$ 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 $\mathsf{exr}_r(C) = \{C_1, \ldots, C_m\}$, $\alpha$ yields an $\mathcal{ALEN}$-concept description $C^\alpha$ 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

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

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, $\mathsf{min}_r(C) \geq 1$ 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 care of $\geq$-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 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 ready to state the structural characterization of subsumption in $\mathcal{ALEN}$.

**Theorem 2** *[Küsters and Molitor, 2000] 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*

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

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

where

- $(\le max\{\mathsf{max}_r(C), \mathsf{max}_r(D)\}\ r)$ 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 $\mathcal{ALEN}$.

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* $C' \in \mathsf{exr}_r(C)^*$.

As known from the literature, [Hemaspaandra, 1999], testing subsumption of $\mathcal{ALEN}$-concept descriptions is PSPACE-complete. However, our characterization is not intended to serve as a basis for a polynomial-space subsumption algorithm. It rather yields the formal basis for the lcs algorithm presented in the next section.

## 5 Computing the lcs in $\mathcal{ALEN}$

The recursive algorithm computing the lcs of two $\mathcal{ALEN}$-concept descriptions $C$ and $D$ is depicted in Figure 1. For a set $\mathcal{C}$ of $\mathcal{ALEN}$-concept descriptions, $\mathsf{c\text{-}lcs}(\mathcal{C})$ is computed by iteratively applying the binary $\mathsf{c\text{-}lcs}$-operation, i.e., $\mathsf{c\text{-}lcs}(\{C_1, \ldots, C_n\}) := \mathsf{c\text{-}lcs}(C_1, \mathsf{c\text{-}lcs}(C_2, \cdots, \mathsf{c\text{-}lcs}(C_{n-1}, C_n) \cdots))$ for $n > 2$.

Since the role depth of $C$ and $D$ is finite, the recursive computation of $\mathsf{c\text{-}lcs}(C, D)$ always terminates. The following theorem states correctness of the lcs algorithm depicted in Figure 1. As an immediate consequence, we obtain that the lcs of two $\mathcal{ALEN}$-concept descriptions always exists. The theorem is proved by induction on the role depth of $\mathsf{c\text{-}lcs}(C, D)$ and makes heavy use of the structural characterization of subsumption given in Theorem 2.

**Theorem 3** *[Küsters and Molitor, 2000] Let $C, D$ be $\mathcal{ALEN}$-concept descriptions. Then,* $\mathsf{c\text{-}lcs}(C, D) \equiv \mathsf{lcs}(C, D)$.

In the remainder of this section, we illustrate the definition of $\mathsf{c\text{-}lcs}(C, D)$ 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 $\mathsf{c\text{-}lcs}(C, D)$ 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 $C \sqsubseteq \mathsf{c\text{-}lcs}(C, D)$ and $D \sqsubseteq \mathsf{c\text{-}lcs}(C, D)$ are satisfied, and

(b) $\mathsf{c\text{-}lcs}(C, D)$ 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

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

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

For existential and value restrictions things are more complicated. Let us first consider the definition of $\mathsf{exr}_r(\mathsf{c\text{-}lcs}(C, D))$, 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 $\mathsf{exr}_r(E)^\alpha$ (conjoined with $\mathsf{val}_r(E)$) 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 $C^1_{ex}$ and $C^2_{ex}$, 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 $\mathsf{min}_r(E) = 1$ and $\Gamma_r(E) = \emptyset$, we set $\mathcal{C}_r(E) = \{\mathsf{val}_r(E)\}$ since then $\mathsf{exr}_r(E) = \emptyset$, and $\exists r.\mathsf{val}_r(E)$ is the unique minimal existential restriction induced by $E$. This case is illustrated by $D_{ex}$ in our running example, where $\mathsf{min}_r(D_{ex}) = 3$, $\mathsf{exr}_r(D_{ex}) = \emptyset$, and $\mathcal{C}_r(D_{ex}) = \{A_1 \sqcap A_2 \sqcap A_3\}$.

Finally, if $\mathsf{min}_r(E) = 0$, 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 $\mathsf{exr}_r(C_{ex})^* = \mathsf{exr}_r(C^1_{ex}) \cup \mathsf{exr}_r(C^2_{ex})$ (see Section 3).

## 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.

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 $\mathcal{ALE}$ [Baader *et al.*, 1999] and $\mathcal{ALN}$ [Cohen and Hirsh, 1994; Baader and Küsters, 1998]. As an immediate consequence of results shown in [Baader *et al.*, 1999] for $\mathcal{ALE}$, we obtain an exponential lower bound for the size of the lcs of two $\mathcal{ALEN}$-concept descriptions. It is, however, not known whether this bound is tight. The lcs algorithm presented in Section 5 takes double exponential time [Küsters and Molitor, 2000]. Nevertheless, since a prototype implementation of the exponential-time lcs algorithm for $\mathcal{ALE}$ behaves quite well in the chemical process engineering application [Baader and Molitor, 2000], the hope is 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 [Horrocks, 1998] for deciding subsumption in $\mathcal{ALEN}$, which then is to be applied to the chemical engineering knowledge base.

## References


[Baader and Küsters, 1998] 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 *Proc. of KI'98*, LNAI 1504. 1998.

[Baader and Molitor, 2000] F. Baader and R. Molitor. Building and structuring description logic knowledge bases using least common subsumers and concept analysis. In *Proc. of ICCS2000*, LNAI 1867. 2000.

[Baader *et al.*, 1999] F. Baader, R. Küsters, and R. Molitor. Computing least common subsumers in description logics with existential restrictions. In *Proc. of IJCAI'99*. Morgan Kaufmann, 1999.

[Cohen and Hirsh, 1994] W.W. Cohen and H. Hirsh. Learning the CLASSIC description logic: Theoretical and experimental results. In *Proc. of KR'94*. Morgan Kaufmann, 1994.

[Cohen *et al.*, 1992] W.W. Cohen, A. Borgida, and H. Hirsh. Computing least common subsumers in description logics. In *Proc. of AAAI'92*. MIT Press, 1992.

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

[Horrocks, 1998] I. Horrocks. Using an expressive description logic: FaCT or fiction? In *Proc. of KR'98*. Morgan Kaufmann, 1998.

[Küsters and Molitor, 2000] Computing least common subsumers in $\mathcal{ALEN}$. LTCS-Report 00-07, RWTH Aachen, Germany, 2000. See http://www-lti.informatik.rwth-aachen.de/Forschung/Reports.html.

[Mantay, 1999] T. Mantay. Computing least common subsumers in expressive description logics. In *Proc. of AI'99*, LNAI 1747. 1999.

[Möller *et al.*, 1998] R. Möller, V. Haaslev, and B. Neumann. Semantics-based information retrieval. In *Proc. of IT&KNOWS*. 1998.

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

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