Approximating $\mathcal{ALCN}$-Concept Descriptions[*]

Sebastian Brandt[†], Ralf Küsters[‡], and Anni-Yasmin Turhan[†]

[†] Theoretical Computer Science,
RWTH Aachen
{sbrandt,turhan}@cs.rwth-aachen.de

[‡] Theoretical Computer Science,
University of Kiel
kuesters@ti.informatik.uni-kiel.de

Abstract

Approximating a concept, defined in one DL, means to translate this concept to another concept, defined in a second typically less expressive DL, such that both concepts are as closely related as possible with respect to subsumption. In a previous work, we have provided an algorithm for approximating $\mathcal{ALC}$-concept descriptions by $\mathcal{ALE}$-concept descriptions. In the present paper, motivated by an application in chemical process engineering, we extend this result by taking number restrictions into account.

1 Introduction

Approximation is a new inference service in Description Logics (DLs) first mentioned by Baader, Küsters, and Molitor [2]. Approximating a concept, defined in one DL, means to translate this concept to another concept, defined in a second typically less expressive DL, such that both concepts are as closely related as possible with respect to subsumption. There are a number of different applications of this inference problem, some of which we will briefly mention here; see [4] for others, such as the translation of knowledge-bases, and knowledge-base vivification.

Non-standard inferences in expressive DLs. Non-standard inferences in DLs, such as computing the least common subsumer (lcs), matching, and unification of concepts, have been introduced to support the construction and maintenance of DL knowledge-bases (see [8, 6] for an overview). However, up to now they are mostly restricted to quite inexpressive DLs, for example to those that do not allow for concept disjunction. Approximation can be used to overcome this

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

problem to some extent. The general idea is to first approximate concepts given in an expressive DL, which yields concepts represented in a less expressive DL, and then apply the non-standard inferences to the approximations.

For example, the existing matching algorithms can be lifted up to handle more expressive DLs as follows: instead of directly matching concept patterns (defined in a small DL) against concepts (defined in a DL that can not be handled by existing matching algorithms), one can first approximate the concept (in the small DL) and then match against its approximation. Even though some information may be lost, e.g., the matcher is more general than the correct one, the accuracy of the result may still suffice.

Another example, which was in fact the main motivation for us to investigate approximation in the first place, is the *computation of commonalities of concepts.* This inference service is used in our chemical process engineering application [10] to support the bottom-up construction of knowledge-bases [1, 6]. Typically, the lcs is employed to accomplish this task. Formally, the lcs of two concepts, say C_1 and C_2, defined in some DL $\mathcal{L}$, is the most specific concept (w.r.t. subsumption) in $\mathcal{L}$ that subsumes both concepts. In case $\mathcal{L}$ provides concept disjunction, the lcs is just the disjunction of C_1 and C_2 ($C_1 \sqcup C_2$). Thus, the problem is that a user inspecting this concept does not learn anything about the commonalities between C_1 and C_2. By using approximation, however, one can make the commonalities explicit by first approximating C_1 and C_2 in a sublanguage of $\mathcal{L}$ which does not allow to express concept disjunction, and then computing the lcs of the approximations in this sublanguage.

Supporting frame-based user interfaces of DL systems. In the interaction with DL systems, users with little knowledge representation expertise may have difficulties to understand and make use of the full expressive power of the underlying DLs. To overcome this problem, some knowledge representation systems have been equipped with a simplified frame-based user interface built on top of a more powerful DL system. One Example for such a system is the ontology editor OilEd [3] built on top of the FaCT DL system [7]. On many occasions, these systems have to present concept descriptions to the user for editing, inspection, or as a solution of inference problems. Such concept descriptions, however, need not always fit into the restricted representation of the frame-based user interface or might overwhelm an inexperienced user. In such cases, approximation may be helpful as a means to represent concept descriptions in a simplified fashion suited to the user interface and the users level of expertise.

In [5], a first in-depth investigation of the approximation problem has been presented. Particularly, a double-exponential time algorithm has been devised to approximate $\mathcal{ALC}$-concepts by $\mathcal{ALE}$-concepts. Despite of the high complexity, our prototypical implementation showed a quite promising performance on runtime and concept sizes, see [4].

Since most applications (like our chemical process engineering application),

Construct name	Syntax	Semantics		
prim. negation, $A \in N_C$	$\neg A$	$\Delta_{\mathcal{I}} \setminus A^{\mathcal{I}}$		
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	$\mathcal{A}$	
existential restrictions	$\exists r.C$	$\{x \in \Delta_{\mathcal{I}} \mid \exists y : (x,y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$	$\mathcal{L}$	$\mathcal{A}$
value restrictions	$\forall r.C$	$\{x \in \Delta_{\mathcal{I}} \mid \forall y{:}(x,y) \in r^{\mathcal{I}} \to y \in C^{\mathcal{I}}\}$	$\mathcal{E}$	$\mathcal{L}$
number restrictions,	$(\geq nr)$	$\{x \in \Delta_{\mathcal{I}} \mid \#\{y : (x,y) \in r^{\mathcal{I}}\} \geq n\}$,	$\mathcal{N}$	$\mathcal{C}$
$r \in N_R$, $n \in \mathbb{N}$	$(\leq nr)$	$\{x \in \Delta_{\mathcal{I}} \mid \#\{y : (x,y) \in r^{\mathcal{I}}\} \leq n\}$		$\mathcal{N}$
full negation	$\neg C$	$\Delta_{\mathcal{I}} \setminus C^{\mathcal{I}}$		
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$		

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

require number restrictions, in this paper we extend the results for $\mathcal{ALC}$ to $\mathcal{ALCN}$ and show how $\mathcal{ALCN}$-concepts can be approximated by concepts in $\mathcal{ALEN}$ or sublanguages thereof. It turns out that the approximation algorithm becomes much more involved.

The structure of the paper is as follows. In the next section, we define the DLs used and introduce the least common subsumer, a key operation in our approximation algorithm. We then (Section 3) formally define the notion of approximation and illustrate it by an example. Our approximation algorithm will be based on a so-called $\mathcal{ALCN}$-normal form, which is introduced in Section 3.1. The main difficulty in computing approximations is to extract induced concept descriptions from given $\mathcal{ALCN}$-concept descriptions. Section 3.2 is devoted to this problem. The proof of correctness of our algorithm uses a characterization of subsumption between $\mathcal{ALCN}$- and $\mathcal{ALEN}$-concept description. Such a characterization is provided in Section 3.3. The approximation algorithm is then presented in Section 3.4. We conclude in Section 4.

2 Preliminaries

Concept descriptions are defined inductively 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.* For the sake of simplicity, we assume N_R to be the singleton $\{r\}$. However, all definitions and results can easily be generalized to arbitrary sets of role names. In this work, we consider the DLs $\mathcal{ALEN}$ and $\mathcal{ALCN}$. Both of these DLs provide the top concept ($\top$), the bottom concept ($\bot$), conjunction ($C \sqcap D$), number restrictions ($(\geq n\ r)$, $(\leq n\ r)$), existential ($\exists r.C$) and value restrictions ($\forall r.C$). In addition, $\mathcal{ALEN}$ offers primitive negation, i.e., negation appears only in front of concept names ($\neg A$, for a concept name $A \in N_C$), $\mathcal{ALCN}$ offers disjunction ($C \sqcup D$) and full negation ($\neg C$). The semantics of $\mathcal{ALCN}$- and $\mathcal{ALEN}$-concept descriptions is defined in the usual model-theoretic way in terms of an *inter-*

pretation $\mathcal{I} = (\Delta_\mathcal{I}, \cdot^\mathcal{I})$. The extension of $\cdot^\mathcal{I}$ to arbitrary concept descriptions is defined inductively, as shown in 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 approximations of $\mathcal{ALCN}$-concept descriptions by $\mathcal{ALEN}$-concept descriptions, for this purpose we need to compute least common subsumers of $\mathcal{ALEN}$-concept descriptions.

Definition 1 (lcs) *Given $n \geq 1$ and $\mathcal{ALEN}$-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. It has been shown in [9] that for $\mathcal{ALEN}$ the lcs always exists and that it can be computed in double exponential time.

3 $\mathcal{ALEN}$-Approximation of $\mathcal{ALCN}$-Concepts

In this section, we show how $\mathcal{ALCN}$-concept descriptions can be approximated (from above) by $\mathcal{ALEN}$-concept descriptions. Let us first define the notion of an upper $\mathcal{ALEN}$-approximation formally.

Definition 2 *Let C be an $\mathcal{ALCN}$-concept description. An $\mathcal{ALEN}$-concept description D is an* (upper) $\mathcal{ALEN}$-approximation *of C ($approx_{\mathcal{ALEN}}(C)$) iff (i) $C \sqsubseteq D$, and (ii) D is minimal with this property, i.e., $C \sqsubseteq D'$ and $D' \sqsubseteq D$ implies $D' \equiv D$ for all $\mathcal{ALEN}$-concept descriptions D'.*

Analogously, an $\mathcal{ALCN}$-concept description can be approximated from below. Since we focus only on upper approximations in this paper, approximation in the following always means upper approximation. Since $\mathcal{ALEN}$ allows for concept conjunction it immediately follows that $\mathcal{ALEN}$-approximations are uniquely determined up to equivalence, if they exist: If D_1 and D_2 are two upper $\mathcal{ALEN}$-approximations of the same $\mathcal{ALCN}$-concept, then so is $D_1 \sqcap D_2$. But then, by definition of upper approximation, $D_1 \sqcap D_2 \sqsubseteq D_1$ and $D_1 \sqcap D_2 \sqsubseteq D_2$ implies $D_1 \sqcap D_2 \equiv D_1 \equiv D_2$.

In [5], for the case of approximating $\mathcal{ALC}$- by $\mathcal{ALE}$-concept descriptions, it was shown that, naive approaches for computing the approximation return too general concept descriptions. For example one might think that every $\mathcal{ALCN}$-concept description D can be approximated by simply replacing every concept disjunction in D by the lcs operator and evaluating the lcs operators from inside out. However, the $\mathcal{ALCN}$-concept description

$$D = \big((\exists r.B \sqcap (\geq 2\ r)) \sqcup (\exists r.B \sqcap \exists r.A) \big) \sqcap \forall r.A,$$

with concept names A and B, illustrates that this is not the case: The obtained approximation would be

$$\begin{array}{rcl} approx_{\mathcal{ALEN}}(D) & \equiv & \mathsf{lcs}\big(\exists r.B \sqcap (\geq 2r),\, (\exists r.B \sqcap \exists r.A)\big) \sqcap \forall r.A \\ & \equiv & \exists r.B \sqcap \forall r.A. \end{array}$$

However, as one can easily check, $C \sqsubseteq \big(\exists r.(B \sqcap A) \sqcap \exists r.A \sqcap \forall r.A\big) \sqsubset \exists r.B \sqcap \forall r.A$. In fact, $\big(\exists r.(B \sqcap A) \sqcap \exists r.A \sqcap \forall r.A\big)$ is the correct $\mathcal{ALEN}$-approximation of C.

To avoid these effects it has already been shown in [5] for the approximation of an $\mathcal{ALC}$-concept description by an $\mathcal{ALE}$-concept description that in order to compute approximations, one needs to turn concepts into a kind of disjunctive normal form and make implicit facts explicit, i.e., compute induced concepts. For $\mathcal{ALC}$, the latter step is rather easy (from a conceptual point of view), since it suffices to make inconsistencies explicit and propagate value restrictions to existential restrictions. For $\mathcal{ALCN}$, things are much more involved since number restrictions can induce new value and existential restrictions. To illustrate the main difficulties one encounters, we consider the following running example.

Example 3 *Consider the $\mathcal{ALCN}$-concept description C_{ex} over the set of concept names $N_C := \{A_1, A_2, P, Q\}$ with*

$$\begin{array}{rlr} C_{ex} := & \big(\exists r.(P \sqcap A_1) \sqcap \exists r.(P \sqcap A_2) \sqcap \exists r.(\neg P \sqcap A_1) \sqcap \exists r.Q \sqcap (\leq 2\ r)\big) & (C_{\text{ex1}}) \\ \sqcup & \big(\forall r.((A_1 \sqcap A_2 \sqcap P) \sqcup (A_1 \sqcap A_2 \sqcap \neg P)) \sqcap (\geq 1\ r)\big). & (C_{\text{ex2}}) \end{array}$$

We want to compute the $\mathcal{ALEN}$-approximation of C_{ex}. To this end, we have to find the number restrictions, existential restrictions, and value restrictions common to both top-level disjuncts C_{ex1} and C_{ex2}.

Induced number restrictions. *As explicit number restrictions, we find $(\leq 2\ r)$ in C_{ex1} and $(\geq 1\ r)$ in C_{ex2}. Since C_{ex1} has one existential restriction with P and another with $\neg P$, we know that at least two r-successors must exist. Thus, C_{ex1}* induces *$(\geq 2\ r)$, and $approx_{\mathcal{ALEN}}(C_{ex})$ will have $(\geq 1\ r)$ as induced at-least restriction and no at-most restriction.*

Induced existential restrictions. *The first disjunct C_{ex1} has 4 existential restrictions while the number of r-successors is limited to 2. Hence, the 4 existential restrictions must be merged into 2 such that consistency is preserved.*

This can be done in two ways, yielding the possibilities

$$\exists r.(P \sqcap A_1 \sqcap A_2 \sqcap Q) \sqcap \exists r.(\neg P \sqcap A_1) \qquad \textit{or}$$
$$\exists r.(P \sqcap A_1 \sqcap A_2) \sqcap \exists r.(\neg P \sqcap A_1 \sqcap Q).$$

Although C_{ex2} has no explicit existential restrictions, the at-least restriction $(\geq 1\ r)$ implies one r-successor for which the value restriction holds. It is easy to see that the value restriction in C_{ex2} is approximated by $\forall r.(A_1 \sqcap A_2)$, so $\exists r.(A_1 \sqcap A_2)$ is an induced $\mathcal{ALEN}$-concept description of C_{ex2}. In both mappings, $A_1 \sqcap A_2$ occurs in one existential restriction. Hence, we know that $\exists r.(A_1 \sqcap A_2)$ will occur in the approximation of C_{ex}.

Induced value restrictions. *The first disjunct C_{ex1} has no explicit value restriction. Nevertheless, as seen above, C_{ex1} has exactly 2 r-successors and every consistent merging has A_1 in every existential restriction. Hence, $\forall r.A_1$ is induced as a value restriction for C_{ex1}. As the disjunction in C_{ex2} yields $\forall r.(A_1 \sqcap A_2)$, the resulting value restriction for C_{ex} is $\forall r.A_1$.*

Summing up, we obtain $(\geq 1\ r) \sqcap \exists r.(A_1 \sqcap A_2) \sqcap \forall r.A_1$ as the approximation of the concept description C_{ex}.

Our approximation algorithm is based on a normal form for $\mathcal{ALCN}$-concept descriptions, that will be defined next.

3.1 $\mathcal{ALCN}$-Normal Form

We call a concept description *top-level $\sqcup$-free*, if it is in negation normal form (i.e., negation is pushed inwards until in front of a concept name) and does not contain any disjunction on top-level (w.r.t. role-depth). Some notation is needed to access the different parts of an $\mathcal{ALEN}$-concept description or a top-level $\sqcup$-free $\mathcal{ALCN}$-concept description C:

- $\mathit{prim}(C)$ denotes the set of all (negated) concept names and the bottom concept occurring on the top-level of C;
- $\mathsf{val}_r(C) := C_1 \sqcap \cdots \sqcap C_n$, if there exist value restrictions of the form $\forall r.C_1, \ldots, \forall r.C_n$ on the top-level of C; otherwise, $\mathsf{val}_r(C) := \top$;
- $\mathsf{ex}_r(C) := \{C' \mid \text{there exists } \exists r.C' \text{ on the top-level of } C\}$;
- $\mathsf{min}_r(C) := \mathit{max}\{k \mid C \sqsubseteq (\geq k\ r)\}$ (Note that $\mathsf{min}_r(C)$ is always finite.);
- $\mathsf{max}_r(C) := \mathit{min}\{k \mid C \sqsubseteq (\leq k\ r)\}$; if there exists no k with $C \sqsubseteq (\leq k\ r)$, then $\mathsf{max}_r(C) := \infty$.

Note that $\mathsf{min}_r(C)$ and $\mathsf{max}_r(C)$ can be computed in polynomial time with an oracle for subsumption. Also, these numbers do not necessarily refer to number restrictions explicitly represented in C, but rather to those induced by C.

Definition 4 *Let C be an $\mathcal{ALCN}$-concept description. C is in $\mathcal{ALCN}$-*normal form *iff $C = \bot$, $C = \top$, or C is of the form $C = C_1 \sqcup \ldots \sqcup C_n$ with $C_i :=$*

$$\bigsqcap_{A \in prim(C_i)} A \sqcap \bigsqcap_{C' \in \mathsf{ex}_r(C_i)} \exists r.C' \sqcap \forall r.\mathsf{val}_r(C_i) \sqcap (\geq \mathsf{min}_r(C_i)\ r) \sqcap (\leq \mathsf{max}_r(C_i)\ r),$$

for all $i = 1, \ldots n$, where the concepts $\mathsf{val}_r(C_i)$ and C' again are in $\mathcal{ALCN}$-normal form; C_i is removed from the disjunction in case $C_i \equiv \bot$.

It is easy to see that every $\mathcal{ALCN}$-concept description can be turned into an equivalent concept description in $\mathcal{ALCN}$-normal form. Note, however, that because disjunctions need to be distributed over conjunctions, the resulting normal form may be of size exponential in the size of the given concept description; for example, the $\mathcal{ALCN}$-normal form of $(A_1 \sqcap A_2) \sqcup \cdots \sqcup (A_{2n-1} \sqcap A_{2n})$ is of size exponential in n.

3.2 Extracting Induced Information from $\mathcal{ALCN}$-Concepts

Example 3 illustrates that we need to take care of induced concepts of the top-level $\sqcup$-free $\mathcal{ALCN}$-concept descriptions C_i (cf. Definition 4) in order to compute approximations. We now show how these induced concepts can be determined. In what follows, let C be a top-level $\sqcup$-free $\mathcal{ALCN}$-concept description. As mentioned above, the induced number restrictions of C are $(\geq \mathsf{min}_r(C)\ r)$ and $(\leq \mathsf{max}_r(C)\ r)$. In the example, we have obtained the induced number restrictions $(\geq 2\ \ r)$ and $(\leq 2\ \ r)$ for C_{ex1} and $(\geq 1\, r)$ for C_{ex2} (there does not exist an induced at-most restriction for C_{ex2}).

Induced existential and value restrictions are not that easy to obtain. We will need to compute the lcs of certain subconcepts on embedded role-levels. Such subconcepts are still $\mathcal{ALCN}$-concepts which must be approximated before applying the $\mathcal{ALEN}$-lcs.

Induced existential restrictions. We need to formalize the merging of existential restrictions, which we encountered for C_{ex1}. This is done by so-called *existential mappings* α. Intuitively, each α reflects one way to merge all explicit existential restrictions to exactly as many r-successors as allowed by the number restriction for r. Formally, α is defined as

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

where $n := min\{\mathsf{max}_r(C), |\mathsf{ex}_r(C)|\}$ and $m := |\mathsf{ex}_r(C)|$. Moreover, for every α we want to enforce that no trivial r-successors $(\exists r.\top)$ are produced and that every mapping α partitions the set $\mathsf{ex}_r(C)$ into exactly n sets. Furthermore, merging existential restrictions must not lead to inconsistencies. This leads to the following conditions on α:

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 $1 \leq i \leq n$ with $\mathsf{ex}_r(C) = \{C'_1, \ldots, C'_m\}$.

As we have seen, there may be several of these mappings for one concept description. 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 $\mathsf{ex}_r(C) = \emptyset$. Given an existential mapping α, the corresponding set of merged concept descriptions is denoted by

$$\mathsf{ex}_r(C)^\alpha := \{\textstyle\bigsqcap_{j\in\alpha(i)} C'_j \mid 1 \leq i \leq n\}$$

with $\mathsf{ex}_r(C) = \{C'_1, \ldots, C'_m\}$. With a given set of k mapping functions $\Gamma_r(C) = \{\alpha_1, \ldots, \alpha_k\}$ the set of induced existential restrictions $\mathsf{ind\text{-}ex}_r(C)$ of C is defined as follows:

- if $\mathsf{ex}_r(C) \neq \emptyset$, then
  $\mathsf{ind\text{-}ex}_r(C) := \big\{ \mathsf{lcs}(\{\mathit{approx}_{\mathcal{ALEN}}(C_l \sqcap \mathsf{val}_r(C)) \mid 1 \leq l \leq k\})$
  $\mid C_j \in \mathsf{ex}_r(C)^{\alpha_j}, 1 \leq j \leq k\big\}$;
- if $\mathsf{ex}_r(C) = \emptyset$ and $\mathsf{min}_r(C) \geq 1$, then $\mathsf{ind\text{-}ex}_r(C) := \{\mathsf{val}_r(C)\}$;
- otherwise, $\mathsf{ind\text{-}ex}_r(C) := \emptyset$.

In our running example, for C_{ex1} the first case applies. We have two existential mappings, say α_1 and α_2, with $\mathsf{ex}_r(C_{\text{ex1}})^{\alpha_1} = \{P \sqcap A_1 \sqcap A_2 \sqcap Q, \neg P \sqcap A_1\}$ and $\mathsf{ex}_r(C_{\text{ex1}})^{\alpha_2} = \{P \sqcap A_1 \sqcap A_2, \neg P \sqcap A_1 \sqcap Q\}$. Extracting the commonalities of all valid existential mappings yields:

$$\begin{aligned}\mathsf{ind\text{-}ex}_r(C_{\text{ex1}}) = \big\{&\mathsf{lcs}\{P \sqcap A_1 \sqcap A_2 \sqcap Q,\ P \sqcap A_1 \sqcap A_2\},\\ &\mathsf{lcs}\{P \sqcap A_1 \sqcap A_2 \sqcap Q,\ \neg P \sqcap A_1 \sqcap Q\},\\ &\mathsf{lcs}\{\neg P \sqcap A_1,\ P \sqcap A_1 \sqcap A_2\},\\ &\mathsf{lcs}\{\neg P \sqcap A_1,\ \neg P \sqcap A_1 \sqcap Q\}\big\}\\ = \{&P \sqcap A_1 \sqcap A_2,\ A_1 \sqcap Q,\ A_1,\ \neg P \sqcap A_1\}.\end{aligned}$$

For C_{ex2}, the second case applies: $\mathsf{ind\text{-}ex}_r(C_{\text{ex2}}) = \{(A_1 \sqcap A_2 \sqcap P) \sqcup (A_1 \sqcap A_2 \sqcap \neg P)\}$.

Induced value restrictions. New 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 C'. Second, the merging of existential restrictions in combination with at-most restrictions may induce value restrictions. In contrast to induced existential restrictions, however, one further needs to take care of at-least restrictions induced by "incompatible" existential restrictions. To this end we need to know the number of r-successors induced by existential restrictions:

$$\kappa_r(C) := \begin{cases} \mathsf{min}_r(\forall r.\mathsf{val}_r(C) \sqcap \sqcap_{C' \in \mathsf{ex}_r(C)} \exists r.C') & \text{if } \mathsf{ex}_r(C) \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

In our running example, we have $\kappa_r(C_{\mathrm{ex1}}) = 2$. Now, only if $\kappa_r(C) = \mathsf{max}_r(C)$, value restrictions can be induced, since only then we "know" all r-successors of instances of C. With all merged existential restrictions obtained from all existential mappings $\alpha \in \Gamma_r(C)$ collected in the set $\mathsf{ex}_r(C)^* = \bigcup_{\alpha \in \Gamma_r(C)} \mathsf{ex}_r(C)^\alpha$ the induced value restriction $\mathsf{ind\text{-}val}_r(C)$ of C is defined as follows:

- if $\mathsf{max}_r(C) = 0$, then $\mathsf{ind\text{-}val}_r(C) := \bot$;
- if $0 < \kappa_r(C) < \mathsf{max}_r(C)$, then $\mathsf{ind\text{-}val}_r(C) := \mathsf{val}_r(C)$;
- if $0 < \kappa_r(C) = \mathsf{max}_r(C)$, then
  $\mathsf{ind\text{-}val}_r(C) := \mathsf{lcs}(\{\mathit{approx}_{\mathcal{ALEN}}(\mathsf{val}_r(C) \sqcap C') \mid C' \in \mathsf{ex}_r(C)^*\})$

In our example, we have $\kappa(C_{\mathrm{ex1}}) = \mathsf{max}_r(C_{\mathrm{ex1}}) = 2$ and $\mathsf{ex}_r(C_{\mathrm{ex1}})^* = \{P \sqcap A_1 \sqcap A_2 \sqcap Q, \neg P \sqcap A_1, P \sqcap A_1 \sqcap A_2, \neg P \sqcap A_1 \sqcap Q\}$. Thus, $\mathsf{ind\text{-}val}_r(C_{\mathrm{ex1}}) = A_1$. For C_{ex2}, no new value restriction is induced since $\kappa_r(C_{\mathrm{ex2}}) = 0 < \infty = \mathsf{max}_r(C_{\mathrm{ex2}})$.

3.3 Characterization of Subsumption

In the previous subsections, we have introduced a normal form for $\mathcal{ALCN}$-concept descriptions and presented means to deal with induced concept descriptions. With these methods at hand, we can provide a structural characterization of subsumption for the case of an $\mathcal{ALCN}$-concept description C subsumed by an $\mathcal{ALEN}$-concept description D. This characterization is later used to prove correctness of our approximation algorithm. Assuming C in $\mathcal{ALCN}$-normal form it is easy to see that C is subsumed by D if and only if every conjunct C_i in C is subsumed. For a single conjunct C_i, the subsumption $C_i \sqsubseteq D$ can be characterized similarly to the case where both concepts come from $\mathcal{ALEN}$ [9].

Theorem 5 *Let C be an $\mathcal{ALCN}$-concept description in $\mathcal{ALCN}$-normal form with n disjuncts $C_1, \dots, C_n$ and let D be an $\mathcal{ALEN}$-concept description in $\mathcal{ALEN}$-normal form. Then, $C \sqsubseteq D$ iff $C \equiv \bot$, $D \equiv \top$, or for every $i \in \{1, \dots, n\}$ it holds that*

1. *$\mathit{prim}(D) \subseteq \mathit{prim}(C_i)$, and*
2. *$\mathsf{max}_r(C_i) \leq \mathsf{max}_r(D)$, and*
3. *$\mathsf{min}_r(C_i) \geq \mathsf{min}_r(D)$, and*

4. *for all* $D' \in \mathsf{ex}_r(D)$,
   (a) $\mathsf{ex}_r(C_i) = \emptyset, \mathsf{min}_r(C_i) \geq 1$, *and* $\mathsf{val}_r(C_i) \sqsubseteq D'$, *or*
   (b) $\mathsf{ex}_r(C_i) \neq \emptyset$ *and for each* $\alpha \in \Gamma_r(C_i)$, *there exists* $C_i' \in \mathsf{ex}_r(C_i)^\alpha$ *such that* $C_i' \sqcap \mathsf{val}_r(C_i) \sqsubseteq D'$, *and*
5. *if* $\mathsf{val}_r(D) \not\equiv \top$, *then*
   (a) $\mathsf{max}_r(C_i) = 0$, *or*
   (b) $\kappa_r(C_i) < \mathsf{max}_r(C_i)$ *and* $\mathsf{val}_r(C_i) \sqsubseteq \mathsf{val}_r(D)$, *or*
   (c) $0 < \kappa_r(C_i) = \mathsf{max}_r(C_i)$ *and* $\mathsf{val}_r(C_i) \sqcap C_i' \sqsubseteq \mathsf{val}_r(D)$, *for all* $C_i' \in \mathsf{ex}_r(C_i)^*$.

Consider our example concept $C_{\text{ex}} = C_{\text{ex1}} \sqcup C_{\text{ex2}}$ and the $\mathcal{ALEN}$-concept description $D := \exists r.(A_1 \sqcap A_2) \sqcap (\geq 1\ r)$. In order to check whether C_{ex} is subsumed by D, we only have to check Conditions 3 and 4 because D has neither primitive concepts nor at-most or value restrictions. As seen in Section 3, C_{ex1} and C_{ex2} have $(\geq 2\ r)$ and $(\geq 1\ r)$ as at-least restrictions, respectively, so that Condition 3 is satisfied. Condition 4 holds for C_{ex1} because every existential mapping produces an existential restriction with $A_1 \sqcap A_2$ (thus satisfying 4b). For C_{ex2}, an appropriate existential restriction is induced by $(\geq 1\ r)$ and the value restriction (satisfying 4a).

3.4 The Approximation Algorithm for $\mathcal{ALCN}$-Concepts

Based on the recursive computation of induced number, value, and existential restrictions as well as the lcs operation, our approximation algorithm is defined as follows. It is quite similar to the lcs algorithm for $\mathcal{ALEN}$-concept descriptions presented in [9].

Definition 6 *Approximation of $\mathcal{ALCN}$ by $\mathcal{ALEN}$.*
Input: $\mathcal{ALCN}$-concept description C. Output: $\mathcal{ALEN}$-approximation of C.

1. *If* $C \equiv \bot$ *then* $\mathsf{c\text{-}approx}_{\mathcal{ALEN}}(C) := \bot$ *or*
   if $C \equiv \top$ *then* $\mathsf{c\text{-}approx}_{\mathcal{ALEN}}(C) := \top$.
2. *Otherwise, transform C into $\mathcal{ALCN}$-normal form $C_1 \sqcup \cdots \sqcup C_n$ and return*

$$
\begin{aligned}
&\mathsf{c\text{-}approx}_{\mathcal{ALEN}}(C) := \\
&\quad \textstyle\sqcap_{A \in \bigcap_i prim(C_i)} A \\
&\quad \sqcap\ (\geq min\{\mathsf{min}_r(C_i) \mid 1 \leq i \leq n\}\ r) \\
&\quad \sqcap\ (\leq max\{\mathsf{max}_r(C_i) \mid 1 \leq i \leq n\}\ r) \\
&\quad \sqcap \bigsqcap_{\substack{(C_1',\ldots,C_n') \in \\ \mathsf{ind\text{-}ex}_r(C_1) \times \cdots \times \mathsf{ind\text{-}ex}_r(C_n)}} \exists r.\mathsf{lcs}\{\mathsf{c\text{-}approx}_{\mathcal{ALEN}}(C_i' \sqcap \mathsf{val}_r(C_i)) \mid 1 \leq i \leq n\} \\
&\quad \sqcap\ \forall r.\mathsf{lcs}\{\mathsf{c\text{-}approx}_{\mathcal{ALEN}}(\mathsf{ind\text{-}val}_r(C_i)) \mid 1 \leq i \leq n\}
\end{aligned}
$$

Returning to our running example, it is easy to check that $\mathsf{c\text{-}approx}_{\mathcal{ALEN}}(C_{\text{ex}}) \equiv (\geq 1\ r) \sqcap \exists r.(A_1 \sqcap A_2) \sqcap \forall r.A_1 \equiv \mathit{approx}_{\mathcal{ALEN}}(C_{\text{ex}})$. More generally, we can show the following theorem. The proof combines the ideas from the proof of the correctness of the lcs algorithm for $\mathcal{ALEN}$-concept descriptions and the correctness of the algorithm for $\mathcal{ALE}$-approximations.

Theorem 7 *Let C be an $\mathcal{ALCN}$-concept description in $\mathcal{ALCN}$-normal form. Then $\mathsf{c\text{-}approx}_{\mathcal{ALEN}}(C)$ is the upper $\mathcal{ALEN}$-approximation of C, i.e.,*

$$\mathit{approx}_{\mathcal{ALEN}}(C) \equiv \mathsf{c\text{-}approx}_{\mathcal{ALEN}}(C).$$

In particular, $\mathcal{ALEN}$-approximations of $\mathcal{ALCN}$-concept descriptions always exist and can be computed effectively.

One can prove $C \sqsubseteq \mathsf{c\text{-}approx}_{\mathcal{ALEN}}(C)$ by structural induction on C using Theorem 5, where $\mathsf{c\text{-}approx}_{\mathcal{ALEN}}(C)$ takes the place of D. In order to prove minimality of $\mathsf{c\text{-}approx}_{\mathcal{ALEN}}(C)$ (w.r.t. subsumption), one assumes another $\mathcal{ALEN}$-concept E subsuming C. Again, using Theorem 5, one can show that $\mathsf{c\text{-}approx}_{\mathcal{ALEN}}(C) \sqsubseteq E$.

By omitting some of the subconcepts computed by $\mathsf{c\text{-}approx}_{\mathcal{ALEN}}$, we obtain approximations in sublanguages of $\mathcal{ALEN}$. For example, if we discard the number restrictions computed by $\mathsf{c\text{-}approx}_{\mathcal{ALEN}}$, we obtain an $\mathcal{ALE}$-approximation of the given $\mathcal{ALCN}$-concept description.

As for the computational complexity of computing $\mathcal{ALEN}$-approximations, we note that the complexity of computing the lcs of $\mathcal{ALEN}$-concept descriptions yields a lower bound since the $\mathcal{ALEN}$-approximation of $C_1 \sqcup C_2$ for two $\mathcal{ALEN}$-concept descriptions C_1 and C_2 is exactly the lcs of C_1 and C_2 in $\mathcal{ALEN}$. For the lcs, a double-exponential time upper bound has been shown in [9]. However, it is not know whether this bound is tight, and whether tight complexity bounds for the lcs would carry over to the approximation problem.

4 Conclusion

We have devised an algorithm to approximate $\mathcal{ALCN}$ concepts by concepts in $\mathcal{ALEN}$ or sublanguages thereof. It remains to determine the exact complexity bound for this problem. From a practical point of view, an interesting question is whether, similar to the $\mathcal{ALC}$ case, our algorithm can be implemented efficiently.

Once one has given an approximation of a concept, a natural question regards the loss of information, i.e., what aspects of the approximated concept are not captured by its approximation. In [5], we proposed an algorithm for computing the difference between $\mathcal{ALC}$ and $\mathcal{ALE}$ concepts. Extending this algorithm to $\mathcal{ALCN}$ and $\mathcal{ALEN}$ remains future work.

References

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

[2] F. Baader, R. Küsters, and R. Molitor. Rewriting concepts using terminologies. In A.G. Cohn, F. Giunchiglia, and B. Selman, eds., *Proc. of KR-00*, pages 297–308, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[3] S. Bechhofer, I. Horrocks, C. Goble, and R. Stevens. Oiled: a reason-able ontology editor for the semantic web. In F. Baader, G. Brewka, and Th. Eiter, eds., *Proc. of KI 2001*, volume 2174 of *Lecture Notes in Artificial Intelligence*, p. 396–408, Vienna, Austria, 2001. Springer–Verlag.

[4] S. Brandt, R. Küsters, and A.-Y. Turhan. Approximation and difference in description logics. LTCS-Report 01-06, LuFG Theoretical Computer Science, RWTH Aachen, Germany, 2001.

[5] S. Brandt, R. Küsters, and A.-Y. Turhan. Approximation and difference in description logics. In D. Fensel, D. McGuiness, and M.-A. Williams, eds., *Proc. of KR-02*, San Francisco, CA, 2002. Morgan Kaufmann Publishers. To appear.

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

[7] I. Horrocks. Using an expressive description logic: FaCT or fiction? In A. G. Cohn, L. Schubert, and S. C. Shapiro, editors, *KR'98: Principles of Knowledge Representation and Reasoning*, pages 636–645. Morgan Kaufmann, San Francisco, California, 1998.

[8] R. Küsters. *Non-Standard Inferences in Description Logics*, volume 2100 of *Lecture Notes in Artificial Intelligence.* Springer-Verlag, 2001. Ph.D. thesis, RWTH Aachen.

[9] R. Küsters and R. Molitor. Computing Least Common Subsumers in $\mathcal{ALEN}$. In B. Nebel, ed., *IJCAI-01*, pages 219–224. Morgan Kaufman, 2001.

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