TBoxes do not yield a compact representation of least common subsumers*

Franz Baader and Anni-Yasmin Turhan
Theoretical Computer Science, RWTH Aachen, Germany
email: {baader,turhan}@cs.rwth-aachen.de

Abstract

For Description Logics with existential restrictions, the size of the least common subsumer (lcs) of concept descriptions may grow exponentially in the size of the input descriptions. This paper investigates whether the possibly exponentially large concept description representing the lcs can always be represented in a more compact way when using an appropriate (acyclic) TBox for defining this description. This conjecture was supported by our experience in a chemical process engineering application. Nevertheless, it turns out that, in general, TBoxes cannot always be used to obtain a polynomial size representation of the lcs.

1 Introduction

In an application in chemical process engineering [5; 9; 7], we support the bottom-up construction of Description Logic (DL) knowledge bases by computing most specific concepts (msc) of individuals and least common subsumers (lcs) of concepts: instead of directly defining a new concept, the knowledge engineer introduces several typical examples as individuals, which are then generalized into a concept description by using the msc and the lcs operation [1; 3; 6]. This description is offered to the knowledge engineer as a possible candidate for a definition of the concept.

Unfortunately, due to the nature of the algorithms for computing the lcs and the msc proposed in [1; 3; 6], these algorithms yield concept descriptions that do not contain defined concept names, even if the descriptions of the individuals use concepts defined in a TBox $\mathcal{T}$. In addition, due to the inherent complexity of the lcs and the msc operation, these descriptions may be quite large (exponentially

*Partially supported by the Deutsche Forschungsgemeinschaft, DFG Project BA 1122/4-1.

large in the size of the unfolded input descriptions). To be more precise, in [3; 2] we considered the lcs in DLs with existential restrictions. For the small DL $\mathcal{EL}$, which allows for conjunctions ($\sqcap$), existential restrictions ($\exists r.C$), and the top concept ($\top$), the binary lcs operation is still polynomial, but the n-ary one is already exponential. For the DL $\mathcal{ALE}$, which extends $\mathcal{EL}$ by value restrictions ($\forall r.C$), primitive negation ($\neg A$, where A is a name of a primitive concept), and the bottom concept ($\bot$), already the binary lcs operation is exponential in the worst case.

To overcome the problem of large least common subsumers, we have employed rewriting of the computed concept description using the TBox $\mathcal{T}$ in order to obtain a shorter and better readable description [4]. Informally, the problem of rewriting a concept given a terminology can be stated as follows: given a TBox $\mathcal{T}$ and a concept description C that does not contain concept names defined in $\mathcal{T}$, can this description be rewritten into an equivalent smaller description D by using (some of) the names defined in $\mathcal{T}$? First results obtained in our process engineering application were quite encouraging: for a TBox with about 65 defined and 55 primitive names, source descriptions of size about 800 (obtained as results of the lcs computation) were rewritten into descriptions of size about 10 [7].

These positive empirical results led us to conjecture that maybe TBoxes can always be used to yield a compact representation of the lcs. More formally, this conjecture can be stated as follows. Let $\mathcal{L}$ be a DL for which the lcs operation (binary or n-ary) is exponential (like $\mathcal{EL}$ or $\mathcal{ALE}$). Given input descriptions $C_1, \ldots, C_n$ with lcs D, does there always exist a TBox $\mathcal{T}$ whose size is polynomial in the size of $C_1, \ldots, C_n$ and a defined concept name A in $\mathcal{T}$ such that $A \equiv_{\mathcal{T}} D$, i.e., the TBox defines A such that it is equivalent to the lcs D of $C_1, \ldots, C_n$? A closer look at the worst-case examples for $\mathcal{EL}$ and $\mathcal{ALE}$ given in [2] also supports this conjecture: the exponentially large least common subsumers constructed there can easily be represented using polynomially large TBoxes.

The contribution of the present paper is to prove that the conjecture is nevertheless false, both for $\mathcal{EL}$ and for $\mathcal{ALE}$. This shows that, in general, rewriting cannot overcome the problem of large least common subsumers. Even though these are just negative results, we think that it is worth publishing them, if only to prevent other researchers from wasting their time on trying to prove the (at first sight quite intuitive) conjecture.

2 The least common subsumer in $\mathcal{EL}$ and $\mathcal{ALE}$

First, we introduce the DLs $\mathcal{EL}$ and $\mathcal{ALE}$ more formally. As usual, *concept descriptions* are inductively defined using a set of *constructors*, starting with a set N_C of *concept names* and a set N_R of *role names*. The constructors determine

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

Table 1: Syntax and semantics of concept descriptions.

the expressive power of the DL. In this paper, we consider concept descriptions built from the constructors shown in Table 1. In $\mathcal{EL}$, concept descriptions are formed using the constructors top concept ($\top$), conjunction ($C \sqcap D$) and existential restriction ($\exists r.C$). The DL $\mathcal{ALE}$ allows for all the constructors introduced in Table 1.

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

A *TBox* is a finite set of concept definitions of the form $A \doteq C$, where A is a concept name and C a concept description. In addition, we require that TBoxes are acyclic and do not contain multiple definitions (see, e.g., [8]). In TBoxes of the DL $\mathcal{ALE}$, negation may only be applied to concept names not occurring on the left-hand side of a concept definition. An interpretation $\mathcal{I}$ is a model of the TBox $\mathcal{T}$ iff it satisfies all its concept definitions, i.e., $A^{\mathcal{I}} = C^{\mathcal{I}}$ for all definitions $A \doteq C$ in $\mathcal{T}$.

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 w.r.t. the TBox $\mathcal{T}$ ($C \sqsubseteq_{\mathcal{T}} D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ holds for all models $\mathcal{I}$ of $\mathcal{T}$. The description C is *subsumed* by D ($C \sqsubseteq D$) iff it is subsumed by D w.r.t. the empty TBox (which has all interpretations as models). The concept descriptions C and D are *equivalent* (w.r.t. $\mathcal{T}$) iff they subsume each other (w.r.t. $\mathcal{T}$). We write $C \equiv_{\mathcal{T}} D$ if C and D are *equivalent* w.r.t. $\mathcal{T}$.

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

Definition 1 *Let $C_1, \ldots, C_n$ be concept descriptions in a DL $\mathcal{L}$. The $\mathcal{L}$-concept description C is a* least common subsumer (lcs) *of $C_1, \ldots, C_n$ in $\mathcal{L}$ iff*

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

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

In general (i.e., for an arbitrary DL $\mathcal{L}$), a given collection of n concept descriptions need not have an lcs. However, if an lcs exists, then it is unique up to equivalence. This justifies to talk about *the* lcs of $C_1, \ldots, C_n$ in $\mathcal{L}$.

In [3; 2] it was shown that, for the DLs $\mathcal{EL}$ and $\mathcal{ALE}$, the lcs always exists. The algorithms for computing the lcs are based on the product of *description trees*, i.e., the input concept descriptions are first transformed into a tree representation, and then the lcs is constructed by building the product tree. In the present paper, we cannot give an exact definition of these algorithms (see [3; 2] for details). Instead, we will illustrate them on two examples, which are the worst-case examples demonstrating that the n-ary lcs in $\mathcal{EL}$ and the binary lcs in $\mathcal{ALE}$ may lead to exponentially large concept descriptions.

2.1 The least common subsumer in $\mathcal{EL}$

For the DL $\mathcal{EL}$, a *description tree* is merely a graphical representation of the syntax of the concept description. Its nodes are labeled with sets of concept names (corresponding to concept names occurring in the description) and its edges are labeled with role names (corresponding to the existential restrictions occurring in the description). We call a node w reachable from a node v by an edge labeled with r an r-successor of v.

For example, the trees depicted in the upper half of Figure 1 were obtained from the concept descriptions

$$\begin{aligned}
C_1^3 &:= \exists r.(P \sqcap \exists r.(P \sqcap Q \sqcap \exists r.(P \sqcap Q))) \ \sqcap \\
& \exists r.(Q \sqcap \exists r.(P \sqcap Q \sqcap \exists r.(P \sqcap Q))), \\
C_2^3 &:= \exists r.(P \sqcap Q \sqcap \exists r.(P \sqcap \exists r.(P \sqcap Q)) \sqcap \exists r.(Q \sqcap \exists r.(P \sqcap Q))), \\
C_3^3 &:= \exists r.(P \sqcap Q \sqcap \exists r.(P \sqcap Q \sqcap \exists r.P \sqcap \exists r.Q))).
\end{aligned}$$

The *product* $\mathcal{G}_1 \times \cdots \times \mathcal{G}_n$ of n $\mathcal{EL}$-description trees $\mathcal{G}_1, \ldots, \mathcal{G}_n$ is defined by induction on the depth of the trees. Let $v_{0,1}, \ldots, v_{0,n}$ respectively be the roots of the trees $\mathcal{G}_1, \ldots, \mathcal{G}_n$ with labels $\ell_1(v_{0,1}), \ldots, \ell_n(v_{0,n})$. Then the product $\mathcal{G}_1 \times \cdots \times \mathcal{G}_n$ has the root $(v_{0,1}, \ldots, v_{0,n})$ with label $\ell_1(v_{0,1}) \cap \ldots \cap \ell_n(v_{0,n})$. For each role r and for each n-tuple $v_1, \ldots, v_n$ of r-successors of $v_{0,1}, \ldots, v_{0,n}$, the root $(v_{0,1}, \ldots, v_{0,n})$ has an r-successor $(v_1, \ldots, v_n)$, which is the root of the product of the subtrees of $\mathcal{G}_1, \ldots, \mathcal{G}_n$ with roots $v_1, \ldots, v_n$. The lower half of Figure 1 depicts the tree obtained as the product of the description trees corresponding to the descriptions C_1^3, C_2^3, C_3^3.

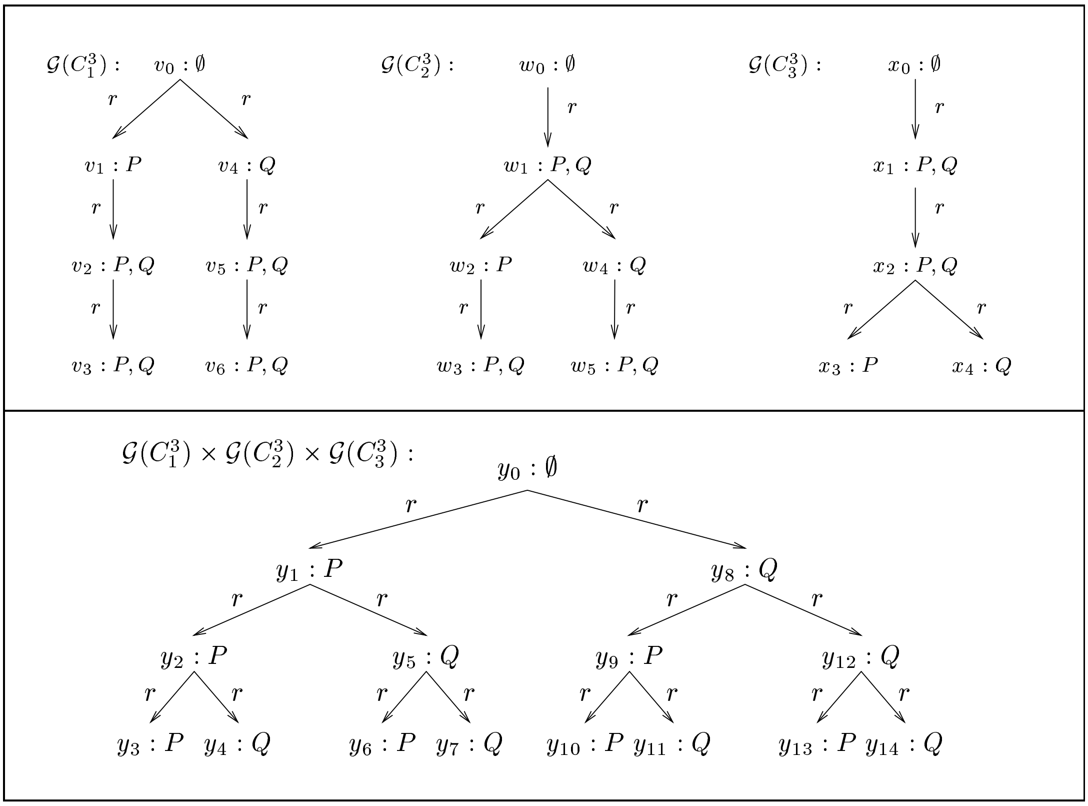

Figure 1: Description trees of C_1^3, C_2^3, C_3^3 and their product.

This example can be generalized to an example that demonstrates that the lcs of n $\mathcal{EL}$-concept descriptions of size linear in n may be exponential in n [2].

Example 2 *We define for each $n \geq 1$ a sequence $\{C_1^n, \ldots, C_n^n\}$ of $\mathcal{EL}$-concept descriptions. For $n \geq 0$ let*

$$D_n := \begin{cases} \top, & n = 0 \\ \exists r.(P \sqcap Q \sqcap D_{n-1}), & n > 0 \end{cases}$$

and for $n \geq 1$ and $1 \leq i \leq n$ we define

$$C_i^n := \begin{cases} \exists r.(P \sqcap D_{n-1}) \sqcap \exists r.(Q \sqcap D_{n-1}), & i = 1 \\ \exists r.(P \sqcap Q \sqcap C_{i-1}^{n-1}), & 1 < i \leq n. \end{cases}$$

It is easy to see that each C_i^n is linear in the size of n. The product of the corresponding description trees is a full binary tree of depth n, where the nodes reached by going to the left are labeled with P and the ones reached by going to the right are labeled with Q. Obviously, the size of this tree is exponential in n. What is less obvious, but can also be shown (see [2]), is that there is no smaller description tree representing the same concept (modulo equivalence).

2.2 The least common subsumer in $\mathcal{ALE}$

$\mathcal{ALE}$*-description trees* are very similar to $\mathcal{EL}$-description trees. The value restrictions just lead to another type of edges, which are labeled by $\forall r$ instead of simply r. However, the concept descriptions must first be normalized before they can be transformed into description trees. On the one hand, there are normalization rules dealing with negation and the bottom concept. Here we will ignore them since neither negation nor bottom is used in our examples. On the other hand, there are normalization rules dealing with value restrictions and their interaction with existential restrictions:

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

The first rule conjoins all value restrictions for the same role into a single value restriction. The second rule is problematic since it duplicates subterms, and thus may lead to an exponential blow-up of the description. The following is a well-known example that demonstrates this effect.

Example 3 *We define the following sequence* $C_1, C_2, C_3, \ldots$ *of* $\mathcal{ALE}$ *concept descriptions:*

$$C_n := \begin{cases} \exists r.P \ \sqcap \ \exists r.Q, & n = 1 \\ \exists r.P \ \sqcap \ \exists r.Q \ \sqcap \ \forall r.C_{n-1}, & n > 1. \end{cases}$$

Obviously, the size of C_n *is linear in* n*. However, applying the second normalization rule to* C_n *yields a description of size exponential in* n*. If one ignores the value restrictions (and everything occurring below a value restriction), then the description tree corresponding to the normal form of* C_n *is again a full binary tree of depth* n*, where the nodes reached by going to the left are labeled with* P *and the ones reached by going to the right are labeled with* Q*. Figure 2 shows the* $\mathcal{ALE}$ *-description tree of the normal form of* C_3*.*

Given the description trees of normalized $\mathcal{ALE}$ -concept descriptions, one can again obtain the lcs as the product of these trees. In this product, the bottom concept requires a special treatment, but we ignore this issue since it is irrelevant for our examples.

For each tuple of nodes, existential restrictions and value restrictions are treated symmetrically, i.e., for a role r the r-successors are combined with r-successors in all possible combinations (as before) and the (unique) $\forall r$-successors are combined with each other. Note that r-successor are not combined with $\forall r$-successors. The following example is taken from [2].

Example 4 *For* $n \geq 1$*, we consider the concept descriptions* C_n *introduced in Example 3 and the concept descriptions* D_n *defined in Example 2. By building*

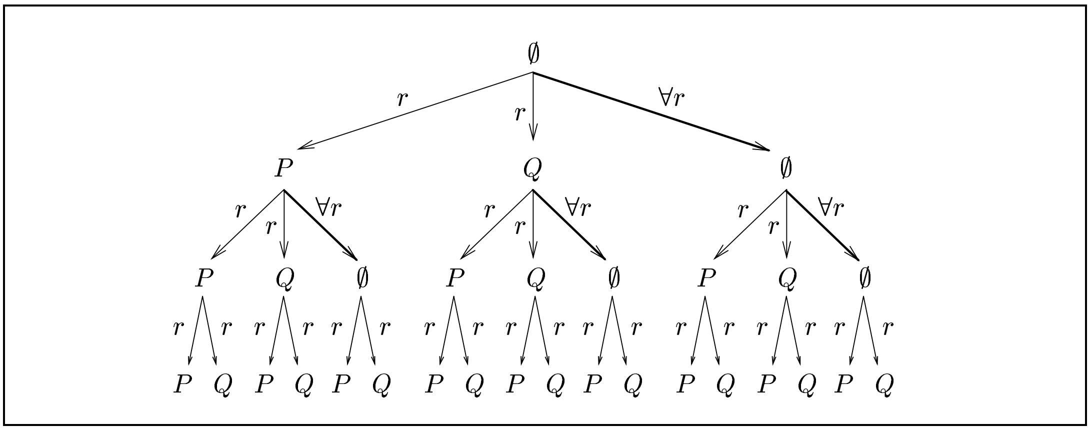

Figure 2: The $\mathcal{ALE}$ -description tree of the normal form of C_3 from Example 3.

the product of the description trees corresponding to the normal forms of C_n and D_n, one basically removes the value restrictions from the normal form of C_n. Thus, one ends up with an lcs that agrees with the one we obtained in Example 2. Again, it can be shown that there is no smaller $\mathcal{ALE}$ -concept description equivalent to this lcs.

3 Using TBoxes to compress the lcs

The exponentially larger lcs E_n constructed in Examples 2 and 4 had as its description tree the full binary tree of depth n, where the nodes reached by going to the left were labeled with P and the ones reached by going to the right were labeled with Q. This concept can be defined in a TBox of size linear in n.

Example 5 *Consider the following TBox $\mathcal{T}_n$:*

$$\{A_1 \doteq \exists r.P \sqcap \exists r.Q\} \;\; \cup$$
$$\{A_i \doteq \exists r.(P \sqcap A_{i-1}) \sqcap \exists r.(Q \sqcap A_{i-1}) \mid 1 < i \leq n\}.$$

It is easy to see that the size of $\mathcal{T}_n$ is linear in n and that $A_n \equiv_{\mathcal{T}_n} E_n$, i.e., the TBox $\mathcal{T}_n$ provides us with a compact representation of E_n.

In general, however, such a compact representation is not possible. We will first give a counterexample for the n-ary lcs in $\mathcal{EL}$, and then for the binary lcs in $\mathcal{ALE}$. The main idea underlying both counterexamples is to generate description trees having exponentially many leaves that are all labeled by sets of concept names that are incomparable w.r.t. set inclusion. To this purpose, we consider the set of concept names $N_n := \{A_j^0, A_j^1 \mid 1 \leq j \leq n\}$, and define $A^{\mathbf{i}} := A_1^{i_1} \sqcap \ldots \sqcap A_n^{i_n}$ for each n-tuple $\mathbf{i} = (i_1, \ldots, i_n) \in \{0,1\}^n$.

3.1 The counterexample for $\mathcal{EL}$

For all $n \geq 1$ we define a sequence $C_1, \ldots, C_n$ of n $\mathcal{EL}$-concept descriptions whose size is linear in n:

$$C_j := \exists r. \bigsqcap_{B \in N_n \setminus \{A_j^0\}} B \quad \sqcap \quad \exists r. \bigsqcap_{B \in N_n \setminus \{A_j^1\}} B.$$

Since each of the concepts C_j contains two existential restrictions, the lcs of $C_1, \ldots, C_n$ contains 2^n existential restrictions. The concept descriptions occurring under these restrictions are obtained by intersecting the corresponding concept descriptions under the existential restrictions of the concept descriptions C_j. It is easy to see that these are exactly the 2^n concept descriptions $A^{\mathbf{i}}$ for $\mathbf{i} \in \{0,1\}^n$ introduced above. Since the descriptions $A^{\mathbf{i}}$ are pairwise incomparable w.r.t. subsumption, it is clear that there is no smaller $\mathcal{EL}$-concept description equivalent to this lcs. We will show that a TBox cannot be used to obtain a smaller representation.

Recall that acyclic TBoxes can be unfolded by replacing defined names by their definitions until no more defined names occur on the right-hand sides [8]. If the defined name A represents the lcs of $C_1, \ldots, C_n$ w.r.t. a TBox, then the description defining A in the unfolded TBox is equivalent to this lcs.

Obviously, to get a more compact representation of the lcs using a TBox, one needs duplication of concept names on the right-hand side of the TBox. During unfolding of the TBox, this would, however, lead to duplication of subconcepts. Since the (description tree of the) lcs we have constructed here has 2^n different leaves, such duplication does not help, since it can only duplicate leaves with the same label, but not generate leaves with different labels.

3.2 The counterexample for $\mathcal{ALE}$

For $n \geq 1$ we define concept descriptions C_n of size quadratic in n. For $n \geq 1$, let $F_j^i := \forall r. \cdots \forall r. A_{j+1}^i$ be the concept description consisting of j nested value restrictions followed by the concept name A_{j+1}^i. We define

$$\begin{aligned} C_1 &:= \exists r. A_1^0 \sqcap \exists r. A_1^1, \\ C_n &:= \exists r. F_{n-1}^0 \ \sqcap\ \exists r. F_{n-1}^1 \ \sqcap\ \forall r. C_{n-1} \qquad \text{for } n > 1. \end{aligned}$$

Figure 3 shows the description tree corresponding to C_3.

Applying the normalization rule $\forall r.E \sqcap \exists r.F \longrightarrow \forall r.E \sqcap \exists r.(E \sqcap F)$ to C_n yields a normalized concept description whose size is exponential in n. If one ignores the value restrictions (and everything occurring below a value restriction), then the description tree corresponding to this normal form of C_n is a full binary tree of depth n whose 2^n leaves are labeled by the 2^n concept descriptions $A^{\mathbf{i}}$ for $\mathbf{i} \in \{0,1\}^n$.

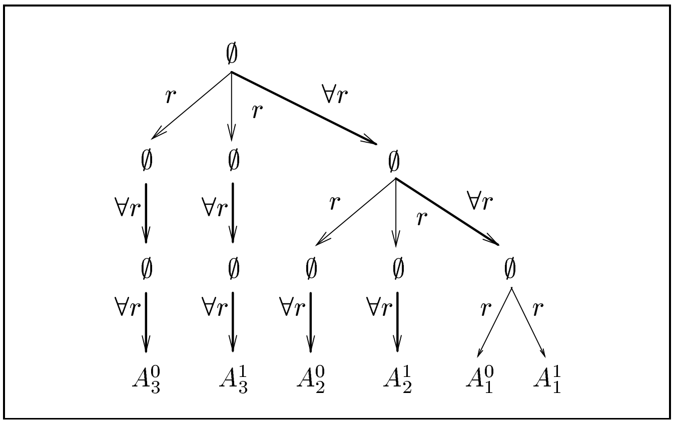

Figure 3: The $\mathcal{ALE}$-description tree corresponding to C_3.

Let $D_n := \exists r. \cdots \exists r. \bigsqcap_{B \in N_n} B$ be the concept description consisting of n nested existential restrictions followed by the conjunction of all concept names in N_n. Again, by building the product of the description trees corresponding to the normal forms of C_n and D_n, one basically removes the value restrictions from the normal form of C_n. Thus, the lcs corresponds to the full binary tree of depth n whose leaves are labeled by the concept descriptions $A^{\mathbf{i}}$ for $\mathbf{i} \in \{0,1\}^n$.

By an argument similar to the one for $\mathcal{EL}$ one can show that there is no smaller $\mathcal{ALE}$-concept description equivalent to this lcs, and that a TBox cannot be used to obtain a smaller representation.

4 Conclusions and future work

The worst-case examples presented in this paper are quite contrived and not likely to occur in practice. Nevertheless, they show that, in principle, the exponential blow-up inherent to the lcs operation cannot be avoided, even if one can introduce "abbreviations" for subdescriptions. An intersting question for future research is to characterize situations in which this exponential blow-up cannot occur, and to check whether these situations are likely to occur in practice.

Another interesting question related to the lcs and TBoxes is the following. Let $\mathcal{L}_1$ be a "large" DL possibly containing disjunction (which makes the lcs operation trivial), and let $\mathcal{L}_2$ be a sublanguage not allowing for disjunction. Now assume that $\mathcal{T}$ is a TBox of $\mathcal{L}_1$ and that $C_1, \ldots, C_n$ are $\mathcal{L}_2$-concept descriptions possibly using names defined in $\mathcal{T}$. Can we compute the most specific $\mathcal{L}_2$-concept description (possibly containing names defined in $\mathcal{T}$) that subsumes $C_1, \ldots, C_n$ w.r.t. $\mathcal{T}$. This operation could, for example, be useful if naïve users of a DL system based on the complex DL $\mathcal{L}_1$ use a simple frame-like interface that only allows them to write descriptions of $\mathcal{L}_2$, but these descriptions can use names defined in a TBox of $\mathcal{L}_1$ (which has been designed by an expert user). The lcs operation modulo a TBox sketched above can now be used to support the definition of new concepts by naïve users.

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 *Proc. KI-98*, Springer LNAI 1504, pages 129–140, 1998.

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

[3] F. Baader, R. Küsters, and R. Molitor. Computing least common subsumer in description logics with existential restrictions. In *Proc. of the 16th Int. Joint Conf. on Artificial Intelligence (IJCAI-99)*, pages 96–101, 1999.

[4] F. Baader, R. Küsters, and R. Molitor. Rewriting concepts using terminologies. In *Proc. of the 7th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR-00)*, pages 297–308, 2000.

[5] F. Baader and U. Sattler. Knowledge representation in process engineering. In *Proceedings of the International Workshop on Description Logics*, 1996. AAAI Press/The MIT Press.

[6] R. Küsters and R. Molitor. Computing least common subsumers in $\mathcal{ALEN}$. In *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI'01)*, 2001. To appear.

[7] R. Molitor. *Unterstützung der Modellierung verfahrenstechnischer Prozesse durch Nicht-Standardinferenzen in Beschreibungslogiken.* PhD thesis, Department of Computer Science, RWTH Aachen, Germany, 2000. In German.

[8] B. Nebel. *Reasoning and Revision in Hybrid Representation Systems*, volume 422 of *Lecture Notes in Computer Science*. Springer–Verlag, 1990.

[9] U. Sattler. *Terminological Knowledge Representation Systems in a Chemical Engineering Application.* PhD thesis, Department of Computer Science, RWTH Aachen, Germany, 1998.