Rewriting in Description Logics Using Terminologies

Franz Baader and Ralf Molitor
LuFg Theoretical Computer Science, RWTH Aachen
email: {baader,molitor}@informatik.rwth-aachen.de

Abstract

We consider the inference problem of computing (minimal) rewritings of concept descriptions using defined concepts from a terminology. We introduce a general framework for this problem and instantiate it with the small description logic $\mathcal{FL}_0$, which provides us with conjunction and value restrictions. We show that the decision problem induced by the minimal rewriting problem is NP-complete for $\mathcal{FL}_0$.

1 Motivation

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 "better" description D by using (some of) the names defined in $\mathcal{T}$? Better may mean shorter, but one can also imagine other optimality criteria. In the formal framework of rewriting introduced in Section 2 of this paper, we will not fix such an optimality criterion, and we will allow $\mathcal{T}$, C, and D to be built over different DLs. However, when instantiating this framework in Section 3, we will assume that $\mathcal{T}$, C, and D are built over the same DL $\mathcal{FL}_0$, and we will use the size of D as optimality criterion.

In the database area, the problem of rewriting queries using views is a well-known research topic [8]. It is closely related to the problem introduced in this paper since views can be regarded as TBox definitions and queries as concepts. However, our motivation for considering this new type of inference problem in DLs is quite different from the one in the DB area. There, one wants to optimize the runtime of queries by using cached views, and thus one wants to minimize the access to source relations. Our goal is to optimize the readability of concepts, and thus minimal length of the concept D appears to be a better optimality criterion.

More precisely, our interest in the rewriting problem stems from an application in chemical process engineering [5, 10]. Within this application, we try to support the bottom-up construction of KBs 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 [2, 1]. 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 [2, 1], 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. To overcome this problem, we want to employ rewriting of the computed concept description using $\mathcal{T}$ in order to obtain a shorter and better readable description.

2 A general framework of rewriting

Definition 1 *Let N_r be a set of role names and N_p a set of* primitive *concept names, and let $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ be three DLs. A* rewriting problem *is given by*

- *an $\mathcal{L}_1$-TBox $\mathcal{T}$ containing only role names from N_r and primitive concept names from N_p, and defining the concept names in N_d;*
- *an $\mathcal{L}_2$-concept description C using only the names from N_r and N_p.*

A rewriting of C using $\mathcal{T}$ *is an $\mathcal{L}_3$-concept description D built using names from N_r and $N_p \cup N_d$ such that C and D are equivalent modulo the TBox $\mathcal{T}$, i.e., $D^{\mathcal{I}} = C^{\mathcal{I}}$ for all models $\mathcal{I}$ of $\mathcal{T}$.*

Given an appropriate ordering $\preceq$ on $\mathcal{L}_3$-concepts built using names from N_r and $N_p \cup N_d$, a rewriting D is called $\preceq$-minimal *iff there does not exist a rewriting D' such that $D' \prec D$.*

It should be noted that there are cases in which there always exists a trivial rewriting; e.g., whenever $\mathcal{L}_1$ is a sublanguage of $\mathcal{L}_3$. However, in these cases finding a minimal rewriting is still a non-trivial task. In the next section, we will consider the case where all three DLs are equal to $\mathcal{FL}_0$, and where the ordering is induced by the size of (a normal form of) D.

In the remainder of this section, we will analyze the results of Section 3 in [8] within the framework introduced above. There are two differences between the rewriting problem considered there and the one introduced above. First, [8] is concerned with *maximally contained* rewritings, i.e., the case where one wants to determine a maximal concept D subsumed by the input concept C. It should be noted, however, that there exists a rewriting in our sense iff the maximally contained rewriting is equivalent to C. Second, [8] is concerned with *total* rewritings (i.e., D may only use defined concept names), whereas we allow for *partial* rewritings (i.e., D may still contain primitive concepts). Section 3 of [8] contains the following two results:

- For $\mathcal{L}_1 = \mathcal{L}_2 = \mathcal{ALCNR}$ and $\mathcal{L}_3 = \{\sqcap, \sqcup\}$, a maximally contained total rewriting is computable. Using the subsumption algorithm for $\mathcal{ALCNR}$, this can be used to decide whether there exists a total rewriting equivalent to the input concept C.
- If $\mathcal{ALCNR}$ is replaced by $\mathcal{ALN}$, then one can compute a maximally contained total rewriting in polynomial time, and existence of a total rewriting equivalent to C can also be decided in polynomial time.

3 Rewriting in $\mathcal{FL}_0$

Under rewriting in $\mathcal{FL}_0$ we understand the instance of the framework introduced above where (i) all three DLs are the language $\mathcal{FL}_0$, which allows for value restrictions and conjunction; and (ii) the TBox is of the usual form (i.e., acyclic and without multiple definitions).

For the DL $\mathcal{FL}_0$, a *concept-based normal form* has turned out to be quite convenient for various purposes [4]: any $\mathcal{FL}_0$-concept description can be written in the form $\forall L_1.P_1 \sqcap \ldots \sqcap \forall L_k.P_k$, where $P_1, \ldots, P_k$ are concept names and the L_i are finite sets of words over the alphabet of role names. This normal form can be obtained by (i) distributing value restrictions over conjunctions; (ii) writing $\forall r_1 \ldots r_n.P_i$ instead of $\forall r_1. \cdots \forall r_n.P_i$; and finally (iii) collecting the words w occurring in a value restriction ending with P_i in the set L_i. For example, the normal form of the $\mathcal{FL}_0$-concept description $P \sqcap \forall r.(\forall r.P \sqcap \forall r.Q)$ is given by $\forall\{\varepsilon, rr\}.P \sqcap \forall\{rr\}.Q$.

Using this normal form, *equivalence of $\mathcal{FL}_0$-concept descriptions* can be characterized as follows [4]. Let C, D be $\mathcal{FL}_0$-concept descriptions with normal forms $C \equiv \forall L_1.P_1 \sqcap \ldots \sqcap \forall L_k.P_k$ and $D \equiv \forall M_1.P_1 \sqcap \ldots \sqcap \forall M_k.P_k$. Then $C \equiv D$ iff $L_i = M_i$ for $i = 1, \ldots, k$. (Note that L_i (M_i) is empty if C (D) does not contain a value restriction ending with P_i.)

With the help of this characterization, the *rewriting problem in $\mathcal{FL}_0$* can be translated into a *formal language problem*. Let $(C, \mathcal{T})$ be an instance of the $\mathcal{FL}_0$ rewriting problem, let $P_1, \ldots, P_k$ be the available primitive concepts, and let $A_1, \ldots, A_\ell$ be the concept names defined in $\mathcal{T}$. We assume that C has the normal form $C \equiv \forall L_1.P_1 \sqcap \ldots \sqcap \forall L_k.P_k$, and that the *unfolded* concept description C_j assigned by $\mathcal{T}$ to the name A_j has the normal form $C_j \equiv \forall N_{j,1}.P_1 \sqcap \ldots \sqcap \forall N_{j,k}.P_k$. Then, the $\mathcal{FL}_0$-concept description D is a rewriting of C using the TBox $\mathcal{T}$ iff its normal form is of the form

$$D \equiv \forall M_1.A_1 \sqcap \ldots \sqcap \forall M_\ell.A_\ell \sqcap \forall K_1.P_1 \sqcap \ldots \sqcap \forall K_k.P_k,$$

where the assignment $X_1 := M_1, \ldots, X_\ell := M_\ell, Y_1 := K_1, \ldots, Y_k := K_k$ solves the system of formal language equations

$$(*) \quad L_i \;=\; X_1 \cdot N_{1,i} \cup \ldots \cup X_\ell \cdot N_{\ell,i} \cup Y_i \quad (1 \leq i \leq k).$$

As an example, we consider the $\mathcal{FL}_0$-concept description $C \equiv \forall\{\varepsilon, rr\}.P_1 \sqcap \forall\{rr\}.P_2$ and the $\mathcal{FL}_0$-TBox

$$\begin{aligned}\mathcal{T} \;:=\; \{&A_1 \doteq P_1 \sqcap \forall r.\forall r.P_2,\ A_2 \doteq \forall r.P_1,\\ &A_3 \doteq \forall r.\forall r.P_2 \sqcap \forall r.A_2\}.\end{aligned}$$

The rewriting problem $(C, \mathcal{T})$ translates into the system of formal language equations

$$\begin{aligned}\{\varepsilon, rr\} &= X_1 \cdot \{\varepsilon\} \cup X_2 \cdot \{r\} \cup X_3 \cdot \{rr\} \cup Y_1,\\ \{rr\} &= X_1 \cdot \{rr\} \cup X_2 \cdot \emptyset \cup X_3 \cdot \{rr\} \cup Y_2.\end{aligned}$$

It is easy to see that the assignment $X_1 := \{\varepsilon\}$, $X_2 := \{r\}$ and $X_3 := Y_1 := Y_2 := \emptyset$ solves this system. This solution yields the rewriting $D := A_1 \sqcap \forall r.A_2$. It should be noted that there are also other solutions of the system; e.g., $X_1 := X_2 := X_3 := \emptyset$ and $Y_1 := \{\varepsilon, rr\}$, $Y_2 := \{rr\}$ is also a solution, which yields the trivial rewriting $D' := C$.

Intuitively, D is a better rewriting than D' since it is shorter, and thus better to read and comprehend. This leads us to the definition of minimal rewritings. As ordering on the rewriting concepts D we choose the size of the concept description obtained after step (i) of the normalization process (since the other steps just introduce a different representation of this description). We can also define this size directly on the concept-based normal form: For a finite set L of words, we define $\|L\| := \sum_{w \in L}(|w| + 1)$, where $|w|$ denotes the length of w. The *size* of an $\mathcal{FL}_0$-concept description $C \equiv \forall L_1.P_1 \sqcap \ldots \sqcap \forall L_k.P_k$ is now defined as $\|C\| := \sum_{1 \leq i \leq k} \|L_i\|$. In the sequel, we are interested in $\preceq$-minimal rewritings where the partial ordering $\preceq$ is defined as $D \preceq D'$ iff $\|D\| \leq \|D'\|$. We call such a rewriting a *cb-minimal rewriting* since the partial ordering $\preceq$

is based on the size of the concept-based normal form of $\mathcal{FL}_0$-concept descriptions.

In our example, it can be shown that $D'' := P_1 \sqcap A_3$ is the unique cb-minimal rewriting of C using $\mathcal{T}$. This rewriting is induced by the solution $Y_1 := X_3 := \{\varepsilon\}$ and $X_1 := X_2 := Y_2 := \emptyset$ of the system of formal language equations.

Although the DL $\mathcal{FL}_0$ is rather small, cb-minimal rewritings cannot be computed by a polynomial-time algorithm (unless P = NP) since the decision problem induced by the minimal rewriting problem is NP-complete.

Theorem 2 *Let C be an $\mathcal{FL}_0$-concept description, $\mathcal{T}$ an $\mathcal{FL}_0$-TBox, and $\kappa \in \mathbb{N}$.*

Deciding whether there exists a rewriting D of C using $\mathcal{T}$ that is in concept-based normal form and of size $\leq \kappa$ is NP-complete.

Outline of the proof. First, we reduce the problem of solving the system of k formal language equations $(*)$ to solving a single formal language equation. Then, we determine a maximal solution of the single equation (cf. Lemma 3). Unfortunately, computing this maximal solution may take time exponential in the size of the TBox. Thus, in order to obtain a non-deterministic *polynomial* algorithm, we introduce a set of possible solutions that (a) contains all solutions (in particular, the maximal solution), and (b) allows us to "guess" one of its members in non-deterministic polynomial time. For the possible solution obtained this way, deciding whether it is in fact a solution and whether its size is $\leq \kappa$ takes time polynomial in the size of C and $\mathcal{T}$. This yields the required NP-decision procedure.

In order to show that the problem is NP-hard, we will give a polynomial reduction of the NP-complete problem SETCOVER [7] to the decision problem induced by the minimal rewriting problem.

Proof that the problem is in NP. For a language L and a word w we define $L{\cdot}w := \{vw \mid v \in L\}$ and $L{\cdot}w^{-1} := \{v \mid vw \in L\}$.

The system of formal language equations $(*)$ can be transformed into an equivalent single equation as follows. Let $R_1, \ldots, R_k$ be k distinct role names (i.e., one for each equation in $(*)$). We introduce the following abbreviations:

$$S_0 \;:=\; \bigcup_{1\leq i\leq k} L_i{\cdot}R_i \quad \text{and}$$
$$S_j \;:=\; \bigcup_{1\leq i\leq k} N_{j,i}{\cdot}R_i \quad \text{for } 1 \leq j \leq \ell.$$

Using these abbreviations, we can rewrite the system of equations $(*)$ into the single equation

$$(**) \quad S_0 \;=\; X_1{\cdot}S_1 \cup \ldots \cup X_\ell{\cdot}S_\ell \cup \bigcup_{1\leq i\leq k} Y_i{\cdot}R_i$$

It is easy to show (see Lemma 13 in [3]) that the assignment $X_1 := M_1, \ldots, X_\ell := M_\ell, Y_1 := K_1, \ldots, Y_k := K_k$ solves the system $(*)$ iff it solves the single equation $(**)$ (see Lemma 13 in [3]).

Because of the presence of the variables Y_i, the system $(*)$ (and thus also the equation $(**)$) always has a solution. The following lemma describes the maximal solution of $(**)$.

Lemma 3 *[3] Let $S_0, S_1, \ldots, S_\ell$ be defined as above, and*

$$\widehat{M_j} \;:=\; \bigcap_{w\in S_j} S_0{\cdot}w^{-1} \quad \textit{for } 1 \leq j \leq \ell, \textit{ and}$$
$$\widehat{K}_i \;:=\; L_i \quad \textit{for } 1 \leq i \leq k.$$

The assignment $X_1 := \widehat{M}_1, \ldots, X_\ell := \widehat{M}_\ell, Y_1 := \widehat{K}_1, \ldots, Y_k := \widehat{K}_k$

1. *solves the equation $(**)$, and*
2. *each solution $M_1, \ldots, M_\ell, K_1, \ldots, K_k$ of $(**)$ satisfies $M_j \subseteq \widehat{M_j}$ for all $1 \leq j \leq \ell$ and $K_i \subseteq \widehat{K}_i$ for all $1 \leq i \leq k$.*

Thus, the problem of guessing an appropriate solution of $(**)$ has been reduced to the problem of guessing appropriate subsets of the sets $\widehat{M}_1, \ldots, \widehat{M}_\ell, \widehat{K}_1, \ldots, \widehat{K}_k$. Unfortunately, the cardinalities of the languages $S_1, \ldots, S_\ell$ (which we need for computing the sets $\widehat{M_j}$) may be exponential in the size of the TBox $\mathcal{T}$. This is due to the fact that the TBox provides for a compact representation of the descriptions C_j: unfolding of $\mathcal{T}$ may lead to an exponential blow-up [9].

This problem can be avoided as follows. Instead of guessing subsets of the sets $\widehat{M}_1, \ldots, \widehat{M}_\ell, \widehat{K}_1, \ldots, \widehat{K}_k$, we guess subsets of a common superset of these sets. Let $\mathcal{X}$ be the set of all prefixes of words in S_0. Since S_0 depends only on the concept description C to be rewritten, and since a word w has $|w| + 1$ prefixes, the cardinality $|\mathcal{X}|$ and the size $\|\mathcal{X}\|$ of $\mathcal{X}$ is polynomial in the size of C, and this set can be computed in polynomial time. As an easy consequence of the definition of $\widehat{M_j}$ and $\widehat{K}_i$ we obtain

$$(***) \quad \widehat{M_j} \subseteq \mathcal{X} \;\text{ and }\; \widehat{K}_i \subseteq \mathcal{X}.$$

Now, the *non-deterministic* decision procedure works as follows:

1. Guess sets $M_j \subseteq \mathcal{X}$, $K_i \subseteq \mathcal{X}$, i.e., determine a possible solution of $(**)$.
2. Test whether the $\mathcal{FL}_0$-concept description D induced by these sets is equivalent to C w.r.t. $\mathcal{T}$.

3. Test whether $\|D\| \leq \kappa$.

Return "yes" (meaning that there is a rewriting of size $\leq \kappa$) if there exists a successful computation for Steps 1–3, i.e., there exist subsets M_j, K_i of $\mathcal{X}$ such that the induced $\mathcal{FL}_0$-concept description is equivalent to C w.r.t. $\mathcal{T}$ and has size $\leq \kappa$. Otherwise, return "no".

The correctness of this non-deterministic algorithm is an immediate consequence of the fact that all solutions of $(**)$ are among the possible solutions guessed in Step 1. This follows from $(***)$ and Lemma 3. It remains to be shown that the algorithm is indeed a non-deterministic *polynomial* algorithm. Since the cardinality of $\mathcal{X}$ is polynomial in the size of C, guessing (polynomially many) subsets of $\mathcal{X}$ can be realized by polynomially many binary decisions. Polynomiality of the equivalence test in Step 2 is less trivial since it must be done w.r.t. a TBox (and unfolding the TBox could lead to an exponential blow-up). Nevertheless, we were able to show (see [3], Theorem 7) that this step can be realized in time polynomial in the size of C and $\mathcal{T}$. Finally, the size of the induced $\mathcal{FL}_0$-concept description D is polynomial in the size of C and $\mathcal{T}$, and thus $\|D\| \leq \kappa$ can also be decided in polynomial time.

Proof of NP-hardness. We will use a reduction of the NP-complete problem SETCOVER. An instance of this problem is of the following form [7]:

Instance: A finite set $\mathcal{U} = \{u_1, \ldots, u_n\}$, a family $\mathcal{F} = \{F_i \subseteq \mathcal{U} \mid 1 \leq i \leq m\}$ of subsets of $\mathcal{U}$, and a number $\kappa \in \mathbb{N}$.

Question: Does there exist a subset $\{F_{i_1}, \ldots, F_{i_\kappa}\}$ of $\mathcal{F}$ of size $k \leq \kappa$ such that $F_{i_1} \cup \ldots \cup F_{i_\kappa} = \mathcal{U}$?

Obviously, we can restrict our attention to instances of the problem where (a) at least $\mathcal{F}$ itself covers $\mathcal{U}$, i.e., $F_1 \cup \ldots \cup F_n = \mathcal{U}$, and (b) $\kappa \leq n$.

For a given instance $(\mathcal{U}, \mathcal{F}, \kappa)$ of the SETCOVER problem, we view $\mathcal{U}$ as set of role names, and define the corresponding instance of the $\mathcal{FL}_0$-rewriting problem as follows:

$$\begin{aligned} C_{\mathcal{U}} &:= \forall \mathcal{U}.P \\ \mathcal{T}_{\mathcal{F}} &:= \{A_j \doteq \forall F_j.P \mid 1 \leq j \leq m\}, \end{aligned}$$

where P is the only primitive concept. Obviously, $C_{\mathcal{U}}$ and $\mathcal{T}_{\mathcal{F}}$ are polynomial in the size of $(\mathcal{U}, \mathcal{F}, \kappa)$.

Lemma 4 *There exists a* cb-minimal *rewriting D of $C_{\mathcal{U}}$ using $\mathcal{T}_{\mathcal{F}}$ with $\|D\| \leq \kappa$ iff there exists a cover of U with $k \leq \kappa$ sets $F_{i_1}, \ldots, F_{i_k}$ from $\mathcal{F}$.*

Proof. The maximal rewriting of $C_{\mathcal{U}}$ using $\mathcal{T}_{\mathcal{F}}$ is of the form $\widehat{D} = \forall\{\varepsilon\}.A_1 \sqcap \ldots \sqcap \forall\{\varepsilon\}.A_m \sqcap \forall \mathcal{U}.P$. Hence, each rewriting of $C_{\mathcal{U}}$ using $\mathcal{T}_{\mathcal{F}}$ is of the form $\forall M_1.A_1 \sqcap \ldots \sqcap \forall M_m.A_m \sqcap \forall K.P$, where $M_j = \emptyset$ or $M_j = \{\varepsilon\}$, $K \subseteq \mathcal{U}$, and $M_1 \cdot F_1 \cup \ldots \cup M_m \cdot F_m \cup K = \mathcal{U}$.

Assume that $F_{i_1}, \ldots, F_{i_k}$ is a cover of $\mathcal{U}$ of size $k \leq \kappa$. Then $D' := A_{i_1} \sqcap \ldots \sqcap A_{i_k}$ is a rewriting of $C_{\mathcal{U}}$ of size k, and thus the cb-minimal rewriting has size $\leq k \leq \kappa$.

Conversely, assume that $D = \forall M_1.A_1 \sqcap \ldots \sqcap \forall M_m.A_m \sqcap \forall K.P$ is a cb-minimal rewriting of $C_{\mathcal{U}}$ using $\mathcal{T}_{\mathcal{F}}$, and $\|D\| \leq \kappa$. We show that $K = \emptyset$. This implies that D is of the form $A_{i_1} \sqcap \ldots \sqcap A_{i_k}$ for some $k \leq \kappa$, and hence $\{F_{i_1}, \ldots, F_{i_k}\}$ is a cover of $\mathcal{U}$ of size $\leq \kappa$.

Assume that $K \neq \emptyset$, and let $u \in K$. Then, $u \notin \bigcup_{M_i = \{\varepsilon\}} F_i$ since, otherwise, removing u from K would yield a smaller rewriting.

Since the whole family $\mathcal{F}$ covers $\mathcal{U}$, the fact that $u \notin \bigcup_{M_i = \{\varepsilon\}} F_i$ implies that there exists an index j such that $M_j = \emptyset$ and $u \in F_j$. This implies, however, that removing u from K and inserting ε into M_j yields a smaller rewriting, which is a contradiction. Thus, we have shown that $K = \emptyset$. □

It should be noted that, in the formulation of Theorem 2, we do not assume that the TBox $\mathcal{T}$ is unfolded. Since it is well-known [9] that the equivalence problem w.r.t. (not unfolded) $\mathcal{FL}_0$-TBoxes is a co-NP-complete problem, one might conjecture that this is the source of complexity for the rewriting problem. This is not true, however: on the one hand, the TBox $\mathcal{T}_{\mathcal{F}}$ defined in the reduction of SETCOVER to the rewriting problem is already unfolded; on the other hand, our NP-algorithm is based on the fact that testing whether a candidate rewriting D is equivalent to C can be realized in polynomial time, even if $\mathcal{T}$ is not assumed to be unfolded.

4 Minimal rewritings in role-based normal form

Above we have used the concept-based normal form of $\mathcal{FL}_0$-concept descriptions for tackling the minimal rewriting problem since it allowed us to reduce the rewriting problem to solving a system of formal language equations. However, the concept-based normal form of a given $\mathcal{FL}_0$-concept description C need not be of minimal size among all $\mathcal{FL}_0$-concept descriptions equivalent to C. For example, the concept $C := P_1 \sqcap \forall r.\forall r.P_1 \sqcap \forall r.\forall r.P_2$, which is in concept-based normal form, is of size 7, whereas the size of the equivalent concept $C' := P_1 \sqcap \forall r.\forall r.(P_1 \sqcap P_2)$ is only 5.[1] In order to obtain rewritings of minimal size among all equivalent $\mathcal{FL}_0$-concept descriptions, the so-called role-based normal form is more appropriate.

Recall that, for a given $\mathcal{FL}_0$-concept description C,

[1] As "size" of an $\mathcal{FL}_0$-concept description we take the sum of the number of $\forall$-constructors and the number of occurrences of concept names. For concepts in concept-based normal form, this coincides with the size we have defined before.

the corresponding concept-based normal form (more precisely, the description obtained after the first normalization step) can be obtained by exhaustively applying the rule $\forall r.(D \sqcap E) \longrightarrow \forall r.D \sqcap \forall r.E$. If we apply this rule in the other direction, we obtain the *role-based normal form* of C (as employed in structural subsumption algorithms such as the one used in CLASSIC [6]). In the above example, the concept $C' = P_1 \sqcap \forall r.\forall r.(P_1 \sqcap P_2)$ is in role-based normal form. The reason for this difference in the sizes of the two normal forms is that the role-based normal form "shares" common prefixes of words occurring in value restrictions, whereas the concept-based normal form does not. In our example, the word rr is shared, which explains the difference in the sizes.

It is easy to show that the role-based normal form of a given $\mathcal{FL}_0$-concept description C is of minimal size among all $\mathcal{FL}_0$-concept descriptions equivalent to C, and that it can be computed in polynomial time. Thus, if we are able to compute rewritings whose role-based normal form is minimal (*rb-minimal rewritings*), then we can compute rewritings that are minimal w.r.t. all equivalent $\mathcal{FL}_0$-concept descriptions.

At first sight, the algorithm for computing cb-minimal rewritings introduced above cannot be used to solve this problem. In fact, the role-based normal form of a cb-minimal rewriting need *not* be rb-minimal, and vice versa.

For example, consider the concept description

$$C := \forall r.\forall r.\forall r.(P_1 \sqcap P_2 \sqcap P_3 \sqcap P_4 \sqcap \forall r.(P_1 \sqcap P_2)),$$

and the TBox

$$\begin{aligned}\mathcal{T} := \{&A_1 \doteq \forall r.\forall r.P_1, A_2 \doteq \forall r.\forall r.P_2, A_3 \doteq P_3 \sqcap P_4,\\ &A_4 \doteq P_1 \sqcap P_2 \sqcap \forall r.P_1, A_5 \doteq \forall r.P_2,\\ &A_6 \doteq \forall r.\forall r.\forall r.(P_1 \sqcap P_2)\}.\end{aligned}$$

The unique cb-minimal rewriting is

$$D_c := \forall r.A_1 \sqcap \forall r.A_2 \sqcap \forall r.A_6 \sqcap \forall r.\forall r.\forall r.A_3,$$

and the unique rb-minimal rewriting is

$$D_r := \forall r.\forall r.\forall r.(A_3 \sqcap A_4 \sqcap A_5).$$

However, the role-based normal form of D_c is

$$D_{cr} := \forall r.(A_1 \sqcap A_2 \sqcap A_6 \sqcap \forall r.\forall r.A_3),$$

and the concept-based normal form of D_r is

$$D_{rc} := \forall r.\forall r.\forall r.A_3 \sqcap \forall r.\forall r.\forall r.A_4 \sqcap \forall r.\forall r.\forall r.A_5.$$

Obviously, we have $\|D_c\| = 10 < 12 = \|D_{rc}\|$ and $\|D_r\| = 6 < 7 = \|D_{cr}\|$.

This example shows that it is not possible to compute an rb-minimal rewriting by first computing a cb-minimal rewriting, and then transforming the obtained concept into role-based normal form. Nevertheless, a simple modification of the non-deterministic algorithm introduced above can be used to treat concepts in role-based normal form.

Theorem 5 *Let C be an $\mathcal{FL}_0$-concept description, $\mathcal{T}$ an $\mathcal{FL}_0$-TBox, and $\kappa \in \mathbb{N}$.*

Deciding whether there exists a rewriting D of C using $\mathcal{T}$ that is in role-based normal form and of size $\leq \kappa$ is NP-complete.

The proof of this theorem is obtained through simple modifications of the proof of Theorem 2. The proof of NP-hardness can be used as it is since the cb-minimal rewriting considered there is just a conjunction of concept names. For concept descriptions of this form, the role-based normal form coincides with the concept-based normal form. The non-deterministic algorithm that shows that the decision problem is in NP must be modified only at Step 3. Instead of checking whether the concept-based normal form induced by the languages guessed in Step 1 is $\leq \kappa$, the modified algorithm computes the corresponding role-based normal form, and then checks whether it is $\leq \kappa$. Correctness of this algorithm is an immediate consequence of the fact that the languages guessed in Step 1 cover all possible rewritings, and thus also the rb-minimal ones.

5 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. of KI'98*, volume 1504 of LNCS, 1998.

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

[3] F. Baader and R. Molitor. Rewriting Concepts Using Terminologies. LTCS-Report 99-06. See http://www-lti.informatik.rwth-aachen.de/Forschung/Papers.html.

[4] F. Baader and P. Narendran. Unification of concept terms in description logics. In *Proc. of ECAI'98*, 1998.

[5] F. Baader and U. Sattler. Knowledge representation in process engineering. In *Proc. of DL'96*, 1996.

[6] A. Borgida and P. Patel-Schneider. A semantics and complete algorithm for subsumption in the CLASSIC description logic. In *JAIR*, 1:277–308, 1994.

[7] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* 1979.

[8] C. Beeri, A.Y. Levy, and M.-C. Rousset. Rewriting queries using views in description logics. In *Proc. of PODS'97*, 1997.

[9] B. Nebel. Terminological reasoning is inherently intractable. *AIJ*, 43(2):235–249, 1990.

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