Least common subsumer computation w.r.t. cyclic $\mathcal{ALN}$-terminologies

Franz Baader and **Ralf Küsters**
LuFG Theoretische Informatik, RWTH Aachen, Ahornstr. 55, 52074 Aachen, Germany
{baader,kuesters}@informatik.rwth-aachen.de

Abstract

Computing least common subsumers (lcs) and most specific concepts (msc) are inference tasks that can be used to support the "bottom up" construction of knowledge bases for KR systems based on description logic. For the description logic $\mathcal{ALN}$, the msc need not always exist if one restricts the attention to acyclic concept descriptions. In this paper, we extend the notions lcs and msc to cyclic descriptions, and show how they can be computed. Our approach is based on the automata-theoretic characterizations of fixed-point semantics for cyclic terminologies developed in previous papers.

1 Introduction

Traditionally, the knowledge base of a DL system is built by first formalizing the relevant concepts of the domain (its terminology, stored in the so-called TBox) by *concept descriptions.* In a second step, the concept descriptions are used to specify properties of objects and individuals occurring in the domain (the world description, stored in the so-called ABox). DL systems provide their users with inference services that support both steps: classification of concepts (subsumption) and individuals (instance) and testing for consistency.

This traditional "top down" approach for constructing a DL knowledge base is not always adequate, though. On the one hand, it need not be clear from the outset which are the relevant concepts in a particular application. On the other hand, even if it is clear which (intuitive) concepts should be introduced, it is in general not easy to come up with formal definitions of these concepts within the available description language. For example, in one of our applications in chemical process engineering [3], the process engineers prefer to construct the knowledge base (which consists of descriptions of standard building blocks of process models, such as reactors) in the following "bottom up" fashion: first, they introduce several "typical" examples of the standard building block as individuals in the ABox, and then they generalize (the descriptions of) these individuals into a concept description that (a) has all the individuals as instances, and (b) is the most specific description satisfying property (a).

The present paper is concerned with developing inference services that can support this "bottom up" approach of building knowledge bases. We split the task of computing descriptions satisfying (a) and (b) from above into two subtasks: computing the most specific concept of a single ABox individual, and computing the least common subsumer of two concepts. The *most specific concept* (msc) of an individual b (the *least common subsumer* (lcs) of two concept descriptions A, B) is the most specific concept description C (expressible in the given description language) that has b as an instance (that subsumes both A and B). For sub-languages of the DL used by the system CLASSIC [4], both tasks have already been considered in the literature [5, 7, 6]. However, the algorithms described in these papers only compute approximations of the msc of an individual. In fact, for ABoxes with cyclic dependencies between individuals, the msc of a given individual need not exist, unless one allows for *cyclic concept descriptions* (i.e., concepts defined by cyclic TBoxes, interpreted with greatest fixed-point semantics). Once one allows for cyclic concept descriptions, the algorithm for computing the lcs must also be able to deal with these descriptions.

As a first solution to these problems, we consider cyclic concept descriptions in the language $\mathcal{ALN}$ (which allows for conjunctions, value restrictions, number restrictions, and atomic negations), and show how (1) the lcs of two such descriptions and (2) the msc of an ABox individual can be computed. In (2) we allow for cyclic descriptions in the ABox, and the msc may also be a cyclic description. Our approach is based on the known automata-theoretic characterizations of subsumption w.r.t. cyclic terminologies with greatest fixed-point semantics [1, 8].

All technical details as well as complete proofs can be found in [2].

2 Definitions and notations

In this section, we introduce the notions msc and lcs more formally, and show how they can be generalized to cyclic $\mathcal{ALN}$-concept descriptions.

$\mathcal{ALN}$-concept descriptions allow for concept conjunctions ($C \sqcap D$), value restrictions ($\forall R.C$), number restrictions ($(\geq m\ R)$, $(\leq n\ R)$), and atomic negations ($\neg A$). The semantics of these operators as well as subsumption of $\mathcal{ALN}$-concept descriptions ($C \sqsubseteq D$) is defined as usual. In order to simplify the presentation of our results we assume $m \geq 1$. Furthermore, since atomic negation can be simulated within $\mathcal{FLN}$, by using $(\leq 0\ R_A)$ in place of A and $(\geq 1\ R_A)$ in place of $\neg A$, where R_A is a new role name only used for this purpose, we restrict our attention to the sub-language $\mathcal{FLN}$ of $\mathcal{ALN}$, which disallows atomic negation. In the following, we use $\perp$ to denote a concept description that is always interpreted by the empty set, such as $(\geq 2\ R) \sqcap (\leq 1\ R)$.

Definition 1 (lcs). *Let C, D, E be $\mathcal{FLN}$-concept descriptions. The concept E is a* least common subsumer *(lcs) of C, D iff it satisfies*

- *$C \sqsubseteq E$ and $D \sqsubseteq E$, and*
- *E is the least $\mathcal{FLN}$-concept description with this property, i.e., if E' is an $\mathcal{FLN}$-concept description satisfying $C \sqsubseteq E'$ and $D \sqsubseteq E'$, then $E \sqsubseteq E'$.*

As shown in [5], the lcs of two $\mathcal{FLN}$-concept descriptions always exists, and it can be computed in polynomial time. Things become less rosy, however, if we consider the most specific concept of ABox individuals.

Definition 2 ($\mathcal{FLN}$-ABoxes). *An $\mathcal{FLN}$-*ABox *$\mathcal{A}$ is a finite set of assertions of the form $R(a, b)$ (role assertion) or $C(a)$ (concept assertion), where a, b are individual names, R is a role name, and C is an $\mathcal{FLN}$-concept description.*

The semantics of ABoxes as well as the instance problem ($a \in_{\mathcal{A}} C$) are defined in the usual way. In particular, individuals are interpreted under the unique name assumption.

Definition 3 (msc). *Let $\mathcal{A}$ be an $\mathcal{FLN}$-ABox, a an individual name in $\mathcal{A}$, and C an $\mathcal{FLN}$-concept description. C is the* most specific concept *for a in $\mathcal{A}$ iff $a \in_{\mathcal{A}} C$ and C is the least concept with this property, i.e., if C' is an $\mathcal{FLN}$-concept description satisfying $a \in_{\mathcal{A}} C'$, then $C \sqsubseteq C'$.*

The following example demonstrates that the msc need not exist if the ABox contains cyclic role assertions: in the ABox $\mathcal{A} := \{R(a, a), (\leq 1\ R)(a)\}$, the individual a does not have a most specific concept. In fact, it is easy to see that a is an instance of $\forall R. \cdots \forall R.((\leq 1\ R) \sqcap (\geq 1\ R))$ for chains of value restrictions of arbitrary length. Consequently, the msc cannot be expressed by a finite $\mathcal{FLN}$-concept description. However, the msc of a can be described by a concept A defined in a cyclic $\mathcal{FLN}$-TBox: $A \doteq (\leq 1\ R) \sqcap (\geq 1\ R) \sqcap \forall R.A$, provided that this cyclic TBox is interpreted with greatest fixed-point semantics.

For a cyclic $\mathcal{FLN}$-TBox $\mathcal{T}$, an interpretation I of the primitive concepts and roles (*primitive interpretation*) can be extended in several ways to an interpretation of the defined concepts. The *gfp-semantics* chooses the greatest of these possible extensions as the gfp-extension of the defined concepts induced by I. Because this extension to a *gfp-model* of $\mathcal{T}$ is uniquely determined by the primitive interpretation I and the terminology $\mathcal{T}$, the following definition of cyclic $\mathcal{FLN}$-concept descriptions and their semantics makes sense.

Definition 4 (cyclic $\mathcal{FLN}$-concept description). *Assume that sets of* primitive *concept names N_P and of role names N_R are fixed. A* cyclic *$\mathcal{FLN}$-*concept description *$C = (A, \mathcal{T})$ is given by a defined concept A in a (possibly cyclic) $\mathcal{FLN}$-TBox $\mathcal{T}$ such that all the primitive concepts in $\mathcal{T}$ are elements of N_P and none of the defined concepts in $\mathcal{T}$ belongs to N_P.*

In this context, an *interpretation* I assigns subsets of $dom(I)$ to elements of N_P and binary relations on $dom(I)$ to elements of N_R. For a given cyclic concept description $C = (A, \mathcal{T})$, the interpretation C^I of C in I is the set assigned to A by the unique extension of I to a gfp-model of T. This shows that, from a semantic point of view, cyclic concept descriptions C behave just like ordinary concept descriptions, i.e., a given interpretation I assigns a unique set $C^I \subseteq dom(I)$ to C. For this reason, the definition of subsumption and of the least common subsumer can be generalized to cyclic concept descriptions in the obvious way: just replace "$\mathcal{FLN}$-concept description" by "cyclic $\mathcal{FLN}$-concept description" in Definition 1. The same is true for the definitions of ABoxes, the instance relationship, and the most specific concept.

3 Computing the lcs of cyclic $\mathcal{FLN}$-concept descriptions

Both subsumption and the lcs of cyclic $\mathcal{FLN}$-concept descriptions can be computed using automata-theoretic characterizations of so-called value-restriction sets. For convenience, we abbreviate the concept description $\forall R_1.\forall R_2 \cdots \forall R_n.C$ $(n \geq 0)$ by $\forall R_1 \cdots R_n.C$, where $R_1 \ldots R_n$ is a word over the alphabet N_R of all role names (i.e., $R_1 \ldots R_n \in N_R^*$). For an interpretation I and a word $W = R_1 \ldots R_n$, we define $W^I := R_1^I \circ \cdots \circ R_n^I$, where $\circ$ denotes the composition of binary relations.

Definition 5. *Let C be a cyclic $\mathcal{FLN}$-concept description and P a primitive concept name or a number restric-*

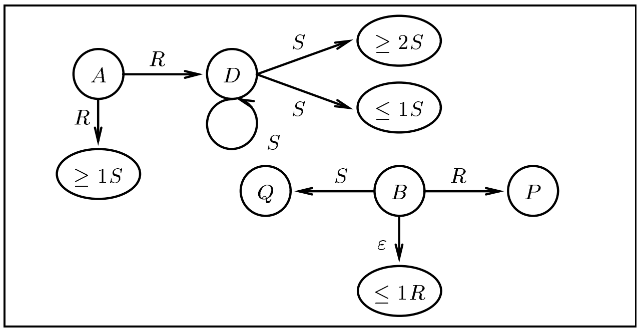

Figure 1: The automata for $\mathcal{T}_A$, $\mathcal{T}_B$.

tion. Then the set $V_C(P) := \{W \in N_R^* \mid C \sqsubseteq \forall W.P\}$ *is called the* value-restriction set *of C for P.*

Even for acyclic descriptions, these value-restriction sets may be infinite. For example, for the (acyclic) description $\bot := (\geq 2\ R) \sqcap (\leq 1\ R)$ and an arbitrary primitive concept name P we have $V_\bot(P) = N_R^*$. The value-restriction sets can, however, be represented by regular languages over the alphabet N_R. To obtain these languages, the TBox of a given cyclic $\mathcal{FLN}$-concept description C is translated into a finite automaton: the concept names and the number restrictions occurring in the TBox are the states of the automaton, and the transitions of the automaton are induced by the value restrictions in the TBox (see [1, 8] for details). For example, the TBoxes $\mathcal{T}_A$ and $\mathcal{T}_B$ defining the descriptions $C_A := (A, \mathcal{T}_A)$ and $C_B := (B, \mathcal{T}_B)$

$$\begin{array}{llcl} \mathcal{T}_A: & A & \doteq & \forall R.D \sqcap \forall R.(\geq 1S) \\ & D & \doteq & \forall S.D \sqcap \forall S.(\geq 2S) \sqcap \forall S.(\leq 1S) \\ \mathcal{T}_B: & B & \doteq & (\leq 1R) \sqcap \forall R.P \sqcap \forall S.Q \end{array}$$

give rise to the automata of Fig. 1. For a cyclic $\mathcal{FLN}$-concept description $C = (A, \mathcal{T})$ and a primitive concept or number restriction P, the language $L_C(P)$ is the set of all words labeling paths in the corresponding automaton from A to P. By definition, these languages are regular. In the example, we have, e.g., $L_{C_A}(\geq 2\ S) = RS^*S$ and $L_{C_B}(P) = \{R\}$.

It is easy to see that the inclusion $L_C(P) \subseteq V_C(P)$ always holds. However, since conflicting number restrictions can create inconsistencies (i.e., unsatisfiable subconcepts), the inclusion in the other direction need not hold. Additionally, the set $V_C(P)$ may contain so-called C-excluding words:

Definition 6. *Let C be a cyclic $\mathcal{FLN}$-concept description. Then the set $E_C := \{W \in N_R^* \mid C \sqsubseteq \forall W.\bot\}$ is called the* set of C-excluding words.

Obviously, if $W \in L_C(\leq m\ R) \cap L_C(\geq n\ R)$ for $m < n$, then W must belong to E_C. Also, since $(\leq 0\ R)$ is equivalent to $\forall R.\bot$, we know that $W \in L_C(\leq 0\ R)$ implies $WR \in E_C$. In addition, if W belongs to E_C, then $WU \in E_C$ for all words U. Finally, for $W \in E_C$, at-least restrictions can also force prefixes of W to belong to E_C. In our example (see Fig. 1), the word R belongs to E_{C_A} since $RS \in E_{C_A}$ and $R \in L_{C_A}(\geq 1\ S)$. Consequently, $E_{C_A} = R\{R, S\}^*$ and it is easy to see that $E_{C_B} = \emptyset$

A more formal characterization of E_C, which also shows that E_C is a regular language, can be found in [8]. To be more precise, a finite automaton that accepts E_C and is exponential in the size of the automaton corresponding to C can be constructed. The following characterization of value-restriction sets is an easy consequence of the results in [8]:

Theorem 7. *Let C be a cyclic $\mathcal{FLN}$-concept description. Then*

1. *$V_C(P) = L_C(P) \cup E_C$ for all primitive concepts P;*
2. *$V_C(\geq m\ R) = \bigcup_{\ell \geq m} L_C(\geq \ell\ R) \cup E_C$ for all at-least restrictions $(\geq m\ R)$;*
3. *$V_C(\leq n\ R) = \bigcup_{\ell \leq n} L_C(\leq \ell\ R) \cup E_C R^{-1}$ for all at-most restrictions $(\leq n\ R)$.*[1]

Consequently, these sets are regular, and finite automata accepting them can be constructed in time exponential in the size of the automaton corresponding to C.

Using the notion of value-restriction sets, the automata-theoretic characterization of subsumption of cyclic $\mathcal{FLN}$-concept descriptions provided in [8] can be formulated as follows: $C \sqsubseteq D$ iff $L_D(P) \subseteq V_C(P)$ for all primitive concept names or number restrictions P. As an easy consequence, we obtain the following characterization of the lcs of such descriptions:

Corollary 8. *Let C, D be cyclic $\mathcal{FLN}$-concept descriptions. Then the cyclic $\mathcal{FLN}$-concept description E is the lcs of C and D if $L_E(P) = V_C(P) \cap V_D(P)$ for all primitive concept names or number restrictions P.*

Given automata for the (non-empty) value-restriction sets $V_C(P)$ and $V_D(P)$, it is easy to construct a cyclic $\mathcal{FLN}$-concept description E that satisfies this property (by simply translating the automata back into TBoxes). This shows that the lcs of two cyclic $\mathcal{FLN}$-concept descriptions can be computed in exponential time, and its size is at most exponential in the size of the input descriptions.

If we apply the characterization of Corollary 8 to our example, we see that a cyclic description E is an lcs of the cyclic $\mathcal{FLN}$-concept descriptions C_A, C_B if it satisfies $L_E(P) = R\{R, S\}^* \cap \{R\} = \{R\}$, $L_E(\leq 1\ R) = (R\{R, S\}^*)R^{-1} \cap \{\varepsilon\} = \{\varepsilon\}$, and all other languages of

[1] For a language L and a letter R, we define $LR^{-1} := \{W \mid WR \in L\}$.

E are empty. Hence, the lcs of C_A and C_B in our example is the (acyclic) description $E := \forall R.P \sqcap (\leq 1\ R)$.

4 Computing the msc in $\mathcal{FLN}$-ABoxes with cyclic descriptions

In the following, we let $\mathcal{A}$ be an arbitrary but fixed $\mathcal{FLN}$-ABox with cyclic concept descriptions. In addition, we assume that $\mathcal{A}$ is consistent,[2] since for inconsistent ABoxes the msc is always the bottom concept $\perp$. In order to decide the instance problem and to compute the msc of an individual in $\mathcal{A}$, we again try to characterize value-restriction sets with the help of regular languages.

Definition 9. *Let a be an individual name in $\mathcal{A}$ and P a primitive concept name or a number restriction. Then the set $V_a(P) := \{W \in N_R^* \mid a \in_{\mathcal{A}} \forall W.P\}$ is called the* value-restriction set *of a for P.*

In addition to the automata corresponding to the cyclic concept descriptions in $\mathcal{A}$, we need an *automaton corresponding to $\mathcal{A}$*: the states of this automaton are the individual names occurring in $\mathcal{A}$, and the transitions are just the role assertions of $\mathcal{A}$, i.e., there is a transition with label R from a to b iff $R(a,b) \in \mathcal{A}$. For individual names a, b occurring in $\mathcal{A}$, the (regular) language $L_a(b)$ is the set of all words labeling paths from a to b in this automaton.

In the previous section, the value-restriction sets $V_C(P)$ for cyclic concept descriptions C could be characterized using the languages $L_C(P)$ and E_C. In order to characterize value-restriction sets for individuals, we first define a regular language whose rôle is similar to the one played by $L_C(P)$:

Definition 10. *Let a be an individual name in $\mathcal{A}$ and P a primitive concept name or a number restriction. Then the set*

$$L_a(P) := \{W \in N_R^* \mid \exists \text{ a concept assertion } C(b) \in \mathcal{A} \text{ and } U \in L_b(a) \text{ such that } UW \in L_C(P)\}$$

is called the predecessor restriction set *of a for P.*

It is easy to see that $L_a(P) \subseteq V_a(P)$. Similar to the corresponding inclusion stated in the previous section, this inclusion relationship may be strict, however. In a first attempt to overcome this problem, we introduce sets E_a corresponding to the sets E_C from the previous section. For this purpose, we adapt the syntactic definition of E_C (see the paragraph below Definition 6) by simply replacing the languages $L_C(\cdot)$ by $L_a(\cdot)$. Thus, if $W \in L_a(\leq m\ R) \cap L_a(\geq n\ R)$ for $m < n$, then W must belong to E_a; if $W \in L_a(\leq 0\ R)$, then $WR \in E_a$; etc. Unfortunately, this syntactic definition of E_a does not completely capture the semantic definition of the set of a-excluding words, i.e., the set of words E_a obtained this way may be smaller than $\{W \in N_R^* \mid a \in_{\mathcal{A}} \forall W.\perp\}$.

It turns out that this problem is a special case of the following more general problem: the definition of the languages $L_a(P)$ only takes into account value restrictions that come from predecessors of a. At-most restrictions in the ABox can, however, also require the propagation of value restrictions from successors of a back to a.

Let us first illustrate this phenomenon by a simple example. Assume that the ABox $\mathcal{A}$ consists of the following assertions: $R(a,b)$, $(\leq 1\ R)(a)$, $(\forall S.P)(b)$. It is easy to see that $RS \notin L_a(P) \cup E_a$. However, $(\leq 1\ R)(a)$ makes sure that, in any model I of $\mathcal{A}$, b^I is the only R^I-successor of a^I. Consequently, all $(RS)^I$-successors of a^I are S^I-successors of b^I, and thus $b^I \in (\forall S.P)^I$ implies $a^I \in (\forall RS.P)^I$. This shows that $RS \in V_a(P)$.

More generally, this problem occurs if concept assertions involving at-most restrictions in the ABox force role chains to use role assertions explicitly present in the ABox. In the example, we were forced to use the assertion $R(a,b)$ when going from a^I to an $(RS)^I$-successor of a^I.

Unfortunately, it is not yet clear how to give a *direct* characterization (as a regular language) of $V_a(P)$ that is based on an appropriate characterization of the set of words in $V_a(P) \setminus (L_a(P) \cup E_a)$ that come from this "backward propagation." Instead, we will describe the complement of $V_a(P)$ as a regular language. Since the class of regular languages is closed under complement, this also shows that $V_a(P)$ is regular. In the following, we restrict our attention to the case where P is a primitive concept name. Number restrictions can be treated similarly.

Before we can give the characterization of $\overline{V_a(P)}$, we need to define one more set of words. Let $R_{\mathcal{A}}(a) := \{b \mid R(a,b) \in \mathcal{A}\}$ denote the set of explicit R-successors of a in $\mathcal{A}$, and let $|R_{\mathcal{A}}(a)|$ denote the cardinality of this set. In addition, let $c_a^{\leq R} := \mathit{min}\{n \mid \varepsilon \in L_a(\leq n\ R)\}$ denote the minimal number occurring in an at-most restriction that must hold for a. Then we define

$$N_a := \{\varepsilon\} \cup \bigcup \{R \cdot N_R^* \mid R \in N_R \text{ and } |R_{\mathcal{A}}(a)| < c_a^{\leq R}\}.$$

Intuitively, a word of the form RU belongs to N_a if at-most restrictions in the ABox do not force all R-successors of a to be reached using role assertions explicitly present in the ABox. The empty word is contained in N_a for technical reasons.

Using the language N_a as well as the languages $L_a(P)$, $L_b(a)$, and E_a, the complement $\overline{V_b(P)}$ of the value-

[2] An ABox is consistent iff it has a model. Note that testing $\mathcal{FLN}$-ABoxes with cyclic descriptions for consistency is a PSPACE-complete problem [2].

restriction set of b for P can be described as follows:

$$\overline{V_b(P)} = \bigcup_{a\in I_{\mathcal{A}}} L_b(a)\cdot(N_a \cap \overline{L_a(P)\cup E_a}) \quad (1)$$

where $I_{\mathcal{A}}$ denotes the set of all individual names occurring in $\mathcal{A}$.

The syntactic description of value-restriction sets for number-restrictions requires modifications of (1) that are similar to the differences between 1., 2., and 3. in Theorem 7, i.e., $L_a(P)$ is replaced by a union of languages $L_a(\geq l\ R)$ and $L_a(\leq l\ R)$, respectively. In addition, for at-most restrictions $(\leq n\ R)$, the set E_a must be replaced by $E_a R^{-1}$. A precise description of the syntactic characterization of value-restriction sets is given in [2].

Since the languages N_a, $L_a(P)$, $L_b(a)$, and E_a involved in this characterization are regular and finite automata accepting them can effectively be computed, this also holds for value-restriction sets:

Theorem 11. *Value-restriction sets are regular and finite automata accepting them can effectively be computed.*

Using these sets, the instance problem can now be decided as follows:

Theorem 12. *Let $\mathcal{A}$ be a consistent $\mathcal{FLN}$-ABox with cyclic concept descriptions, C be a cyclic $\mathcal{FLN}$-concept description, and b an individual occurring in $\mathcal{A}$. Then $b \in_{\mathcal{A}} C$ iff for all primitive concept names or number restrictions P we have $L_C(P) \subseteq V_b(P)$.*

As an easy consequence of this theorem, we obtain the following characterization of the msc:

Corollary 13. *Let $\mathcal{A}$ be a consistent $\mathcal{FLN}$-ABox with cyclic concept descriptions, C be a cyclic $\mathcal{FLN}$-concept description, and b an individual occurring in $\mathcal{A}$. Then C is the msc of b in $\mathcal{A}$ if for all primitive concept names or number restrictions P we have $L_C(P) = V_b(P)$.*

Given automata for the sets $V_b(P)$ it is again easy to construct a cyclic $\mathcal{FLN}$-concept description C that satisfies this property. This shows that the msc can effectively be computed.

5 Related and future work

An important topic for future work is to determine the exact worst-case complexities for computing the lcs and the msc, and for deciding the instance problem for $\mathcal{FLN}$-ABoxes with cyclic concept descriptions. Our algorithm for computing the lcs of two cyclic $\mathcal{FLN}$-concept descriptions is exponential, and we conjecture that this complexity cannot be avoided, i.e., there is no polynomial algorithm for computing the lcs in this case.

A naive analysis of the algorithms for deciding the instance problem and for computing the msc derived from our characterization of value-restriction sets would yield a triply exponential upper bound (due to repeated application of powerset construction). We conjecture, however, that the instance problem can be decided in PSPACE, and that the msc can be computed in exponential time.

To the best of our knowledge, all the existing work on computing the lcs of description logic concepts [5, 7, 6] can only handle acyclic concept descriptions. In addition, the approach for computing the msc proposed by Cohen and Hirsh [7] yields only an approximation of the msc. In fact, since they allow for acyclic descriptions only, they cannot always derive an exact description for the msc. The pragmatic solution proposed in [7] is to restrict the length of value restriction chains occurring in the computed description by some arbitrary but fixed number. This way, one obtains an acyclic description, which may, however, be less specific than the real msc.

References

[1] F. Baader. Using automata theory for characterizing the semantics of terminological cycles. *Annals of Mathematics and Artificial Intelligence*, 18(2–4):175–219, 1996.

[2] F. Baader and R. Küsters. Computing the least common subsumer and the most specific concept in the presence of cyclic $\mathcal{ALN}$-concept descriptions. Technical Report LTCS-Report 98-06, RWTH Aachen, Germany, 1998. See http://www-lti.informatik.rwth-aachen.de/Forschung/Papers.html.

[3] F. Baader and U. Sattler. Knowledge representation in process engineering. In *Proc. of DL'96*, Cambridge (Boston), MA, U.S.A., 1996.

[4] A. Borgida, R. J. Brachman, D. L. McGuinness, and L. A. Resnick. CLASSIC: A structural data model for objects. In *Proc. of the 1989 ACM SIGMOD International Conference on Management of Data*, pages 59–67, Portland, OR, 1989.

[5] W. W. Cohen, A. Borgida, and H. Hirsh. Computing least common subsumers in description logics. In William Swartout, editor, *Proc. of the 10th Nat. Conf. on Artificial Intelligence*, pages 754–760, San Jose, CA, July 1992. MIT Press.

[6] W. W. Cohen and H. Hirsh. Learnability of description logics with equality constraints. *Machine Learning*, 17(2/3), 1994.

[7] W. W. Cohen and H. Hirsh. Learning the classic description logic: Theoretical and experimental results. In *Proc. of KR-94*, pages 121–133, San Francisco, Calif., 1994. Morgan Kaufmann.

[8] R. Küsters. Characterizing the semantics of terminological cycles in $\mathcal{ALN}$ using finite automata. In *Proc. of KR'98*, Trento, Italy, 1998. To appear.