Characterizing the semantics of terminological cycles with the help of finite automata

Ralf Küsters
LuFG Theoretische Informatik, RWTH Aachen, Ahornstr. 55, 52074 Aachen, Germany
email: kuesters@cantor.informatik.rwth-aachen.de

Abstract

The representation of terminological knowledge may naturally lead to terminological cycles. In addition to descriptive semantics, the meaning of cyclic terminologies can also be captured by fixed-point semantics. To gain a more profound understanding of these semantics and to obtain inference algorithms for inconsistency, subsumption, and related inference tasks, this paper provides automata theoretic characterizations of these semantics. The already existing results for the language $\mathcal{FL}_0$ are extended to $\mathcal{ALN}$, which additionally allows for primitive negation and number-restrictions. Moreover, this work considers the relationship between certain schemas and $\mathcal{ALN}$-terminologies.

1 Introduction

Fixed-point semantics were first introduced by B. Nebel [Neb91] to capture the meaning of cyclic terminologies. In [Baa96], these semantics have been analyzed with the help of finite automata for the very small language $\mathcal{FL}_0$, which allows for concept conjunction and (universal) value-restriction. This automata theoretic characterization helps to decide which semantic to prefer in a specific representation task. In addition, it yields decision procedures and complexity results for subsumption. Since $\mathcal{FL}_0$ is not expressive enough for most practical representation problems, this paper[1] extends $\mathcal{FL}_0$ to $\mathcal{ALN}$ by adding primitive negation and number-restrictions. Terminological cycles in much more expressive extensions of $\mathcal{FL}_0$ have already been investigated in [Sch94] and [GL94]. K. Schild has extended the language $\mathcal{ALC}$ by the fixed-point operators of the μ-calculus to $\mu\mathcal{ALC}$, and has shown—among other results—that $\mu\mathcal{ALC}$ is more expressive than cyclic $\mathcal{ALC}$-terminologies[2]. Moreover, the language $\mu\mathcal{ALC}$ has been extended in [GL94] by (qualified) number-restrictions to $\mathcal{ALCN}\mu$. Thus, the language $\mathcal{ALCN}\mu$ contains $\mathcal{ALN}$. Consistency as well as subsumption for $\mathcal{ALCN}\mu$-concepts is EXPTIME-complete, whereas these problems are merely in PSPACE for $\mathcal{ALN}$, which justifies considering this restricted case separately. For both $\mathcal{ALCN}\mu$ and $\mathcal{ALN}$, the important inference problems can be decided with the help of finite automata. However, the automata for $\mathcal{ALCN}\mu$ are of exponential size and they are tree automata that reflect certain semantic structures, whereas the automata for $\mathcal{ALN}$ are finite automata that are merely syntactic variants of $\mathcal{ALN}$-terminologies.

In addition to the characterizations of cyclic $\mathcal{ALN}$-terminologies, this paper analyzes the relationship between $\mathcal{ALN}$-terminologies and $\mathcal{SL}_{dis}$-schemas, which have been introduced in [BDNS97]. It turns out that $\mathcal{SL}_{dis}$-schemas can be seen—w.r.t. inconsistency, validity, and subsumption—as special $\mathcal{ALN}$-terminologies. Consequently, inference problems involving these schemas can be reduced to inference problems of the corresponding terminologies.

2 Preliminaries

A terminology is called *cyclic* if there exists at least one concept which (directly or indirectly) occurs in its own definition. The following is an example (taken from [Neb91]) of a cyclic $\mathcal{ALN}$-terminology consisting only of one concept definition, where Human denotes a defined concept, Mammal a primitive concept, and parents a role:

$$T: \mathsf{Human} \;=\; \mathsf{Mammal} \sqcap \exists^{\geq 2}\mathsf{parents} \sqcap \exists^{\leq 2}\mathsf{parents} \sqcap \forall\mathsf{parents}.\mathsf{Human}$$

This terminology defines human beings as those mammals having exactly two parents all of whom are human beings. In the presence of cyclic terminologies, the interpretation of primitive concepts and roles (Mammal resp. parents in the example) cannot always be extended to a

[1] and in more detail [Küs97], which also provides detailed proofs of all the results

[2] To ensure the existence of least and greatest fixed-point models, recursively defined concepts must occur in their definition positively.

unique model of the considered terminology, i.e., there may be several possibilities of interpreting the defined concepts (Human in the example) to obtain a model of the whole terminology. In addition to descriptive semantics, which allows all models of the terminology as admissible models, B. Nebel introduced greatest fixed-point semantics (gfp-semantics) and least fixed-point semantics (lfp-semantics), which allows (w.r.t. set inclusion) only the greatest (least) extensions of the defined concepts as admissible models. To characterize these semantics, we associate a (non-deterministic) semi-automaton[3] $\mathcal{A}_T$ to a terminology T as follows:

First, T must be normalized such that the right-hand side of every concept definition is a conjunction of concept terms of the form $\forall R_1.\forall R_2 \cdots \forall R_n.C$ (short: $\forall W.C$ for $W = R_1 \cdots R_n$) for role names $R_1, \ldots, R_n$ and a concept, a primitive negation, or a number-restriction C. For a normalized terminology T, the concepts, primitive negation, and number-restrictions of T are the states of $\mathcal{A}_T$; the alphabet of $\mathcal{A}_T$ consists of the role names occurring in T; a concept definition $A = \forall W_1.A_1 \sqcap \cdots \sqcap \forall W_k.A_k$ gives rise to k transitions, where the transition from A to A_i is labeled by the word W_i (for details see [Baa96, Küs97]).

For the terminology of the example, we obtain the automaton $\mathcal{A}_T$ shown in figure 1. In the following, the regular language $L_{\mathcal{A}_T}(A, C)$ for a concept A and a concept, a primitive negation, or a number restriction C denotes the set of words (over the role names in T) that are accepted by the finite automaton ($\mathcal{A}_T$, A, C), where A denotes the initial state and C the final state of $\mathcal{A}_T$. Furthermore, $dom(I)$ denotes the domain of the interpretation I, and for a word $W = R_1 \cdots R_n$ we use W^I to denote the composition $R_1^I \circ \cdots \circ R_n^I$ of the relations R_i^I, $1 \leq i \leq n$.

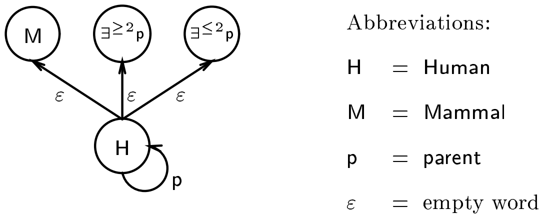

Figure 1: The semi-automaton $\mathcal{A}_T$

Since $\mathcal{ALN}$ allows for number-restrictions, primitive negation can be dispensed with: The terms $\neg P$ and P in a terminology can respectively be replaced by $\exists^{\leq 0} R_P$ and $\exists^{\geq 1} R_P$ for a new role name R_P. The language $\mathcal{FLN}$ denotes the language $\mathcal{ALN}$ without primitive negation. Without loss of generality, we consider $\mathcal{FLN}$ instead of $\mathcal{ALN}$ in the following.

In order to characterize inconsistency and subsumption, the notion of "*requiring*" is useful. Due to number-restrictions, concepts can "require" chains of role successors which have to start from every instance of such a concept.

Definition 1 (require).
Let T be an $\mathcal{FLN}$-terminology, let $\mathcal{A}_T$ be the corresponding semi-automaton and let A be a concept in T. Furthermore, $W = R_1 \cdots R_n$ denotes a finite word and $V = R_1 \cdots R_m$ a prefix of W, i.e. $m \leq n$. The word W is *required by A starting from V* iff for all i, $m \leq i < n$, there are numbers $m_{i+1} \geq 1$ such that $V R_{m+1} \cdots R_i \in L(A, \exists^{\geq m_{i+1}} R_{i+1})$. In the case of $V = \varepsilon$ we say "W is required by A" instead of "W is required by A starting from ε". $\diamond$

Because of $\mathsf{parents}^j \in L_{\mathcal{A}_T}(\mathsf{Human}, \exists^{\geq 2}\mathsf{parents})$ for all $j \geq 0$ in the example, every word $\mathsf{parents}^j$ is required by Human. Consequently, there has to exist an infinite chain of ancestors for every instance of Human.

In the following, we need two more definitions. For a semi-automaton $\mathcal{A}_T$, a set F of $\mathcal{A}_T$-states, and a role R we define the sets:

$$\begin{aligned}\varepsilon\text{-}closure(F) &:= \{q';\ \text{there is a state } q \text{ in } F \text{ and an } \varepsilon\text{-path from } q \text{ to } q' \text{ in } \mathcal{A}_T\},\\ next_\varepsilon(F, R) &:= \varepsilon\text{-}closure(\{q';\ \text{there is a state } q \text{ in } \varepsilon\text{-}closure(F) \text{ and a transition from } q \text{ to } q' \text{ with label } R \text{ in } \mathcal{A}_T\}).\end{aligned}$$

Before characterizing inconsistency and subsumption, a closer look at the semantics themselves is needed.

3 Characterizing the semantics

The automata theoretic characterizations of the three semantics for $\mathcal{FLN}$ are easy extensions of the results for $\mathcal{FL}_0$. The characterization for gfp-semantics is as follows (for lfp- and descriptive semantics see [Baa96, Küs97]):

Theorem 2 (gfp-semantics).
Let T be an $\mathcal{FLN}$-terminology, and let $\mathcal{A}_T$ be the corresponding semi-automaton. Let I be a gfp-model of T, and let A be a concept name occurring in T. For every $d \in dom(I)$ we have $d \in A^I$ iff the following property holds:

> For all C that are either a primitive concept or a number-restriction, all words $W \in L_{\mathcal{A}_T}(A, C)$, and all individuals $e \in dom(I)$, $(d, e) \in W^I$ implies $e \in C^I$.

$\square$

This theorem leads to a more profound understanding of (gfp-)semantics and enables us to characterize and decide inconsistency as well as subsumption.

[3] A semi-automaton consists of a finite set of states, a finite alphabet and a finite set of transitions between the states labeled with words over the given alphabet.

4 Characterizing inconsistency

In $\mathcal{FL}_0$, inconsistent concepts only occur for lfp-semantics. Due to conflicting number restrictions, inconsistent $\mathcal{FLN}$-concepts may occur also for gfp- and descriptive semantics. The characterization of inconsistency for the gfp-semantics is—as the characterization of the gfp-semantics itself—easier than for the other two semantics. In this paper, we consider only gfp-semantics; for lfp- and descriptive semantics see [Küs97].

Before formulating the characterization of inconsistency w.r.t. gfp-semantics, we introduce the notion of "*exclusion sets*", which allows to construct decision algorithms for both inconsistency and subsumption. Intuitively, an exclusion set describes a set of $\mathcal{A}_T$-states which require words leading to conflicting number-restrictions.

Definition 3 (exclusion set).
Let T be an $\mathcal{FLN}$-terminology and $\mathcal{A}_T = (\Sigma, Q, E)$ be the corresponding semi-automaton without word-transitions[4]. The set $F_0 \subseteq Q$ is called *exclusion set* w.r.t. $\mathcal{A}_T$ iff there is a word $R_1 \cdots R_n \in \Sigma^*$, and for all i, $1 \leq i \leq n$, numbers $m_i \geq 1$ as well as conflicting number-restrictions $\exists^{\geq l} R$ and $\exists^{\leq r} R$, $l > r$, such that for $F_i := next_\varepsilon(F_{i-1}, R_i)$, $1 \leq i \leq n$, we have: $\exists^{\geq m_i} R_i \in F_{i-1}$ for all $1 \leq i \leq n$ and $\exists^{\geq l} R, \exists^{\leq r} R \in F_n$. ◇

Now we are able to formulate

Theorem 4 (inconsistency w.r.t. gfp-semantics).
Let T be an $\mathcal{FLN}$-terminology, let $\mathcal{A}_T$ be the corresponding semi-automaton (without word-transitions), and let A be a concept in T. Then the following statements are equivalent:

1. A is T-inconsistent w.r.t. gfp-semantics.
2. There is a word $W \in \Sigma^*$ required by A and there are conflicting number-restrictions $\exists^{\geq l} R$ and $\exists^{\leq r} R$, $l > r$, with $W \in L(A, \exists^{\geq l} R) \cap L(A, \exists^{\leq r} R)$.
3. $\varepsilon\text{-}closure(\{A\})$ is an exclusion set.

□

For a given set of states, it is in general a non-trivial problem to decide whether this set is an exclusion set. The complexity of this problem results from the fact that role successors can be required. Since it can be shown that the property of being an exclusion set is decidable using polynomial space, we have a PSPACE-algorithm for inconsistency w.r.t. gfp-semantics. Inconsistency for lfp- and descriptive semantics can also be decided by a PSPACE-algorithm using exclusion sets. For this purpose, the definition of exclusion sets has to be extended appropriately. In addition to these PSPACE-results, one can prove the NP-hardness of inconsistency for all three semantics by reducing schema problems to problems of terminologies (see section 6), and using existing complexity results for schemas [BDNS97]. For so-called "*weak-acyclic*"[5] $\mathcal{ALN}$-terminologies the complexity of these problems can be shown to be NP-complete.

5 Characterizing subsumption

For the language $\mathcal{FL}_0$, subsumption has been characterized in [Baa96] by inclusion of regular languages. With respect to gfp-semantics the characterization is as follows:

$$A \sqsubseteq_{gfp,T} B \quad \text{iff} \quad L_{\mathcal{A}_T}(B, P) \subseteq L_{\mathcal{A}_T}(A, P) \text{ for all primitive concepts } P. \tag{1}$$

Due to conflicting number restrictions, the right-hand side of this equivalence is not necessary for the subsumption of A and B w.r.t. $\mathcal{FLN}$-terminologies: The concept A may be "*excluded*" by a word that is contained in $L_{\mathcal{A}_T}(B, P)$ but not in $L_{\mathcal{A}_T}(A, P)$. To simplify the formal definition of the notion "exclusion" we can (without loss of generality) assume that an $\mathcal{FLN}$-terminology contains no minimum-restrictions of the form $\exists^{\leq 0} R$ since such a term can be substituted by $\forall R.\bot$[6]. In the following we consider only terminologies without $\exists^{\leq 0} R$, which we call $\mathcal{FLN}^r$-terminologies.

Definition 5 (exclusion).
Let T be an $\mathcal{FLN}^r$-terminology, let $\mathcal{A}_T = (\Sigma, Q, E)$ be the corresponding semi-automaton, and let A be a concept in T. The word $W \in \Sigma^*$ *excludes* A iff there exists a prefix $V \in \Sigma^*$ of W, a word $V' \in \Sigma^*$ as well as conflicting number-restrictions $\exists^{\geq l} R$, $\exists^{\leq r} R$, $l > r$, such that $VV' \in L(A, \exists^{\geq l} R) \cap L(A, \exists^{\leq r} R)$ and VV' is required by A starting from V. ◇

It is easy to see that if a word W excludes a concept A, every individual that has a W-successor cannot be an instance of A. In the following example, the concept A is excluded by the word RS, and we will formally show that no instance of A has RS-successors:

Let T be an $\mathcal{FLN}^r$-terminology and $\mathcal{A}_T$ the corresponding automaton given in figure 2. Furthermore, let I be a gfp-model of T, and let d be an individual with $d \in A^I$. Assume $d(RS)^I e$. Since $RS \in L_{\mathcal{A}_T}(A, \exists^{\geq 1} Q)$, there has to be a Q-successor f of e. Further, since $RSQ \in L_{\mathcal{A}_T}(A, \exists^{\geq 3} S) \cap L_{\mathcal{A}_T}(A, \exists^{\leq 2} S)$, the characterization of the semantics yields the contradiction: $f \in$

[4] A semi-automaton without word-transitions allows only for transitions labeled with letters or the empty word. Word-transitions can be eliminated by replacing each of them with new introduced transitions (labeled with letters or the empty word) using new states.

[5] A terminology is weak-acyclic if $W \notin L_{\mathcal{A}_T}(A, A)$ for all *non-empty* finite words W and all concepts A of T.

[6] The symbol $\bot$ denotes the empty concept, which can be described by $\exists^{\geq 2} R \sqcap \exists^{\leq 1} R$.

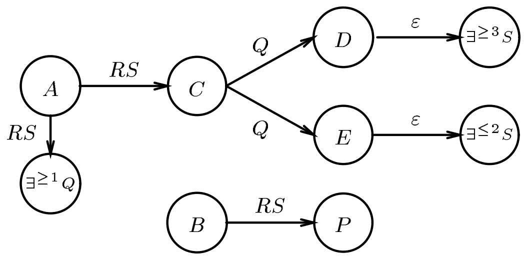

Figure 2: The terminology T given by $\mathcal{A}_T$

$(\exists^{\geq 3} S)^I$ and $f \in (\exists^{\leq 2} S)^I$. Thus, d cannot have an RS-successor.

This proves that no instance of A has RS-successors. Thus, for every $d \in A^I$ (I gfp-model of T) we have: $d \in (\forall RS.P)^I$, and hence $d \in B^I$. So the subsumption relation $A \sqsubseteq_{gfp,T} B$ holds. Together with $L_{\mathcal{A}_T}(B,P) \not\subseteq L_{\mathcal{A}_T}(A,P)$ (since $\{RS\} = L_{\mathcal{A}_T}(B,P)$ and $RS \notin L_{\mathcal{A}_T}(A,P)$) the example shows that because of excluding words (RS in the example) the right-hand side of (1) is not necessary for subsumption w.r.t. $\mathcal{ALN}$.

With the help of E_A, which denotes the set of all A-excluding words, we are able to characterize subsumption in $\mathcal{FLN}^r$ ($\mathcal{ALN}$) w.r.t. gfp-semantics as follows:

Theorem 6 (subsumption w.r.t. gfp-semantics). Let T be an $\mathcal{FLN}^r$-terminology, let $\mathcal{A}_T$ be the corresponding semi-automaton, and let A, B be concept names occurring in T. Then we have: $A \sqsubseteq_{gfp,T} B$ iff

1. $L(B,P) \subseteq L(A,P) \cup E_A$ for all primitive concepts P in T; and
2. $L(B,\exists^{\geq l} R) \subseteq (\bigcup_{r \geq l} L(A, \exists^{\geq r} R) \cup E_A)$ for all maximum-restrictions of the form $\exists^{\geq l} R$ in T with $l > 0$; and
3. $L(B,\exists^{\leq l} R) \cdot R \subseteq (((\bigcup_{r \leq l} L(A, \exists^{\leq r} R)) \cdot R) \cup E_A)$ for all minimum-restrictions of the form $\exists^{\leq l} R$ in T.[7]

□

Similar to the characterization of subsumption w.r.t. lfp- and descriptive semantics for $\mathcal{FL}_0$, this characterization of gfp-semantics for $\mathcal{FLN}^r$ can be extended to lfp- and descriptive semantics as well. Again, the set E_A plays an important role in these characterizations whereby the definition of E_A has to be modified according to the considered semantics. Characterizing E_A by exclusion sets, we are able to formulate a PSPACE-algorithm for subsumption. Furthermore, together with the results in [Baa96], this yields PSPACE-completeness for subsumption w.r.t. gfp- and lfp-semantics in $\mathcal{FLN}^r$ ($\mathcal{ALN}$).

[7] For $L \subseteq \Sigma^*$ and $R \in \Sigma^*$ the set $L \cdot R$ is defined as $\{w \cdot R; w \in L\}$ where "$\cdot$" denotes concatenation of words.

6 $\mathcal{ALN}$-schemas as special $\mathcal{ALN}$-terminologies

In [BDNS97], a terminology has been divided into a schema and a view part—following (object-oriented) databases. The schema merely restricts the number of admissible models of the terminology so that the meaning of schemas is captured by descriptive semantics. In the view part of the terminology, concepts are *defined* with the help of schema concepts. For this reason, fixedpoint semantics is used for the view part.

Knowledge engineers are interested in validity of schemas as well as subsumption w.r.t. schemas. In [BDNS97] $\mathcal{SL}_{dis}$-schemas have been introduced and a special PSPACE-decision algorithm has been developed for deciding (local) validity of these schemas. In [Küs97], it is shown that $\mathcal{ALN}$-schemas (and therefore $\mathcal{SL}_{dis}$-schemas) are—w.r.t. inconsistency, validity, and subsumption—special $\mathcal{ALN}$-terminologies. Consequently, inference problems for schemas can be reduced to those of terminologies. In addition, it is possible—as already mentioned—to prove the co-NP-hardness of the consistency problem for $\mathcal{ALN}$-terminologies using this reduction and complexity results from [BDNS97].

In the following we have a closer look at this reduction by constructing a terminology T_S from a schema S and considering the relationship between T_S and S w.r.t. inconsistency and subsumption. But first of all, we have to introduce schemas formally.

An $\mathcal{ALN}$-schema S consists of a finite set of concept inclusions, which define necessary conditions for concepts, and role inclusions, which define (simple) necessary conditions for roles. Concept inclusions are of the form $A \sqsubseteq D$ where A denotes a concept name and D an $\mathcal{ALN}$-concept term. Role inclusions are of the form $R \sqsubseteq A \times B$ for the role name R and the (primitive or defined) concept names A and B. An interpretation I is a *model* of S if all concept inclusions $A \sqsubseteq B$ and all role inclusions $R \sqsubseteq A \times B$ of S are satisfied, i.e., $A^I \subseteq B^I$ and $R^I \subseteq A^I \times B^I$. A concept A is *consistent* w.r.t. S if there is a model I of S with $A^I \neq \emptyset$. A schema S is *locally valid* if every concept in S is consistent. We call a schema S *valid* if there is a model I with $A^I \neq \emptyset$ for every concept A in S. The following fact will allow us to reduce validity of schemas to consistency of terminologies:

An ($\mathcal{SL}_{dis}$-)$\mathcal{ALN}$-schema is valid iff it is locally valid (see [BDNS97, Küs97]). (2)

In [Küs97], consistency, (local) validity and subsumption of $\mathcal{ALN}$-schemas are reduced to the corresponding problems of $\mathcal{SLN}$-schemas. These schemas allow for role inclusions but restrict the right-hand side of concept inclusions to concept names, primitive negation, number restrictions and (universal) value restriction of the form $\forall R.B$ for a role name R and a concept name B. The $\mathcal{SL}_{dis}$-schemas introduced in [BDNS97] coincide with $\mathcal{SLN}$-schemas apart from the fact that they only allow for number restrictions of the form $\exists^{\leq 1}R$ and $\exists^{\geq 1}R$. On the other hand, $\mathcal{SL}_{dis}$-schemas allow for arbitrary negation. Since it can be shown that negation can also be expressed by primitive negation, $\mathcal{SLN}$-schemas extend $\mathcal{SL}_{dis}$-schemas with arbitrary number-restrictions.

Now, we construct an $\mathcal{ALN}$-terminology T_S from an $\mathcal{SLN}$-schema S such that T_S behaves like S w.r.t. consistency, (local) validity and subsumption. Before defining T_S, we transform S into a schema S' that does not contain role inclusions. Role inclusions only have to be taken into account if role successors are required, otherwise they can be neglected. The definition of S' is as follows:

The schema S' contains all concept inclusions from S of the form $A \sqsubseteq B$, $A \sqsubseteq \neg B$, $A \sqsubseteq \forall R.B$ and $A \sqsubseteq \exists^{\leq n}R$. In addition, S' contains a concept inclusion $A \sqsubseteq \exists^{\geq n}R$, $n \geq 1$, from S if there is no role inclusion in S belonging to R, i.e., a role inclusion of the form $R \sqsubseteq C_1 \times C_2$ in S. For every pair of inclusions $A \sqsubseteq \exists^{\geq n}R$, $n \geq 1$, and $R \sqsubseteq C_1 \times C_2$ in S, the schema S' contains the concept inclusion $A \sqsubseteq \exists^{\geq n}R \sqcap \forall R.C_2 \sqcap C_1$. The schema S' contains no other inclusions than these, especially no role inclusions.

Definition 7 (the terminology T_S).
Let S be an $\mathcal{SLN}$-schema and S' defined as above. For every defined concept A in S an axiom for A in T_S is constructed as follows:
Let $A \sqsubseteq C_1, \ldots, A \sqsubseteq C_n$ be all concept inclusions belonging to the defined concept A in S'. Let $\overline{A}$ be a new (primitive) concept. The axiom for A in T_S is of the form $A = \overline{A} \sqcap C_1 \sqcap \cdots \sqcap C_n$. ◇

Obviously, we can construct T_S from S in time linear in the size of S. Since T_S takes role inclusions into account if role successors are required ($A \sqsubseteq \exists^{\geq n}R$, $n \geq 1$), the following theorem holds:

Theorem 8.
Let S be an $\mathcal{SLN}$-schema, and let T_S be the corresponding $\mathcal{ALN}$-terminology as defined above. For all concepts A and B in S we have:

- A is S-consistent iff A is T_S-consistent w.r.t. descriptive semantics (iff A is T_S-consistent w.r.t. gfp-semantics).
- $A \sqsubseteq_S B$ iff $A \sqsubseteq_{T_S} B$ (iff $A \sqsubseteq_{gfp,T_S} B$).

□

This shows that consistency—and with (2) also (local) validity—of ($\mathcal{ALN}$-)$\mathcal{SLN}$-schemas is decidable using polynomial space by deciding the problem "ε-*closure*($\{A\}$) is an exclusion set". The algorithm presented in [BDNS97] for deciding consistency of $\mathcal{SL}_{dis}$-schemas is quite similar to this algorithm. Both traverse a graph searching for a conflict node.

Furthermore, as an immediate consequence of the theorem, subsumption for $\mathcal{SLN}$-schemas (and therefore $\mathcal{SL}_{dis}$-schemas) is in PSPACE. In [BDNS97], this PSPACE-result has been shown for $\mathcal{SL}_{dis}$-schemas. The complexity of subsumption for schemas is only due to testing consistency of concepts. After testing the consistency of the left-hand side of the subsumption problem, testing subsumption for $\mathcal{SL}_{dis}$- ($\mathcal{SLN}$- and $\mathcal{ALN}$-) schemas is a simple (i.e. polynomial) syntactical test. Hence, disallowing concept forming operators that enable definition of inconsistent concepts (number restrictions and primitive negation) makes reasoning tractable: In fact, for $\mathcal{FL}_0$-schemas, subsumption is in P. However, subsumption for $\mathcal{FL}_0$-terminologies w.r.t. gfp-semantics is still PSPACE-hard.

References

[Baa96] F. Baader. Using automata theory for characterizing the semantics of terminological cycles. In *Annals of Mathematics and Artificial Intelligence*, volume 18, pages 175–219. J.C. Baltzer AG, Science Publishers, 1996.

[BDNS97] M. Buchheit, F.M. Donini, W. Nutt, and A. Schaerf. Refining the structure of terminological systems: Terminology = schema + view. *Proceedings of the 12th National Conference of the American Association for Artificial Intelligence*, AAAI-94, Seattle, 1994. An extended version will appear in: *Artificial Intelligence*, 1997.

[GL94] G. De Giacomo and M. Lenzerini. Concept languages with number restrictions and fixpoints, and its relationship with mu-calculus. In *Proceedings of the Eleventh European Conference on Artificial Intelligence, ECAI-94*, pages 411–415. John Wiley and Sons, 1994.

[Küs97] R. Küsters. Characterizing the semantics of terminological cycles with the help of finite automata. Diploma thesis (in German), RWTH Aachen, May 1997. ftp://www-lti.informatik.rwth-aachen.de/pub/papers/1997/Kuesters-DA-97.ps.gz

[Neb91] B. Nebel. Terminological cycles: Semantics and computational properties. In J. Sowa, editor, *Formal Aspects of Semantic Networks*, pages 331–361. Morgan Kaufmann, San Mateo, 1991.

[Sch94] K. Schild. Terminological cycles and the propositional μ-calculus. In *Proceedings of the Fourth International Conference on Principles of Knowledge Representation and Reasoning, KR'94*, pages 509–520, Bonn, Germany, 1994. Morgan Kaufmann.