Characterizing the Semantics of Terminological Cycles in $\mathcal{ALN}$ Using Finite Automata

Ralf Küsters
LuFg Theoretische Informatik, RWTH Aachen, Ahornstr. 55, 52074 Aachen, Germany
e-mail: 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, namely, greatest and least fixed-point semantics. To gain a more profound understanding of these semantics and to obtain inference algorithms as well as complexity results for inconsistency, subsumption, and related inference tasks, this paper provides automata theoretic characterizations of these semantics. More precisely, the already existing results for $\mathcal{FL}_0$ are extended to the language $\mathcal{ALN}$, which additionally allows for primitive negation and number-restrictions. Unlike $\mathcal{FL}_0$, the language $\mathcal{ALN}$ allows to express inconsistent concepts, which makes non-trivial extensions of the characterizations and algorithms necessary. Nevertheless, the complexity of reasoning does not increase when going from $\mathcal{FL}_0$ to $\mathcal{ALN}$. This distinguishes $\mathcal{ALN}$ from the very expressive languages with fixed-point operators proposed in the literature. It will be shown, however, that cyclic $\mathcal{ALN}$-terminologies are expressive enough to capture schemas in certain semantic data models.

1 INTRODUCTION

Cyclic terminologies were first investigated by B. Nebel (Nebel, 1987; Nebel, 1990a; Nebel, 1990b), who has introduced fixed-point semantics (namely, greatest and least fixed-point semantics) in addition to descriptive semantics to capture the meaning of such terminologies. Which of the semantics should be preferred depends on the particular representation problem at hand. Furthermore, Nebel has shown—by proving the finite model property—that for the language $\mathcal{ALN}$, which allows for concept conjunction, (universal) value-restriction, primitive negation, and unqualified number-restrictions, subsumption w.r.t. cyclic terminologies and the descriptive semantics is decidable.

In order to gain a more profound understanding of all three semantics as well as more feasible decision algorithm and complexity results for subsumption, F. Baader (Baader, 1996b) has proposed automata theoretic characterizations of the three semantics for the small representation language $\mathcal{FL}_0$, which allows for concept conjunction and (universal) value-restrictions. Following this approach, B. Nebel (Nebel, 1991) has given an automata theoretic characterization of equivalence of concepts w.r.t. cyclic $\mathcal{FL}_0$-terminologies. Although (Nebel, 1991) introduces the semantics for $\mathcal{ALN}$, the technical results are restricted to the sublanguage $\mathcal{FL}_0$.

Since $\mathcal{FL}_0$ is not expressive enough for most practical representation problems, in this paper the results for $\mathcal{FL}_0$ are extended to the language $\mathcal{ALN}$. Generalizing the results from $\mathcal{FL}_0$ to $\mathcal{ALN}$ is not trivial due to inconsistent concepts, which are expressible in $\mathcal{ALN}$, but not in $\mathcal{FL}_0$. Technically, this fact is dealt with by introducing the notion of so-called *excluding words*. The new complexity results presented in this paper are summarized in Table 1 for all three semantics both for general (i.e., possibly cyclic) and weak-acyclic terminologies (allowing only for ε-cycles) where 'PSp' denotes PSPACE-completeness and 'NP' NP-completeness. The complexity upper bounds are proved by deriving decision algorithms from the automata theoretic characterizations of inconsistency and subsumption presented in this work. The complexity lower bounds are (more or less straightforward) consequences of known complexity results for $\mathcal{FL}_0$-terminologies and schemas.

Table 1: Complexity Results for $\mathcal{ALN}$-Terminologies

	descriptive	gfp	lfp
Subsumption general	PSp	PSp	PSp
Inconsistency general	PSp	PSp	PSp
Inconsistency (weak-)acyclic	NP	NP	NP

Terminological cycles in much more expressive extensions of $\mathcal{FL}_0$ have been investigated in (Schild, 1994) and (Giacomo and Lenzerini, 1994). 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 general $\mathcal{ALC}$-terminologies.[1] Moreover, the language $\mu\mathcal{ALC}$ has been extended in (Giacomo and Lenzerini, 1994) by (qualified) number-restrictions to $\mathcal{ALCQ}\mu$. Thus, the language $\mathcal{ALCQ}\mu$ contains $\mathcal{ALN}$. Consistency as well as subsumption for $\mathcal{ALCQ}\mu$-concepts is EXPTIME-complete, whereas these problems are merely PSPACE-complete for $\mathcal{ALN}$ (see Table 1). This justifies to consider the restricted case separately since it is likely that implemented decision algorithms for PSPACE-complete problems are easier to optimize than those for EXPTIME-complete problems (Giunchiglia and Sebastiani, 1996). For both $\mathcal{ALCQ}\mu$ and $\mathcal{ALN}$, the important inference problems can be decided with the help of finite automata. However, the automata for $\mathcal{ALCQ}\mu$ are of exponential size and they are tree automata reflecting certain semantic structures, while the automata for $\mathcal{ALN}$ are finite automata on words that are merely syntactic variants of $\mathcal{ALN}$-terminologies.

Cycles can increase the complexity of inference problems or may even lead to undecidability. For instance, subsumption for general $\mathcal{FL}_0$-terminologies which allow for feature agreements is undecidable (Nebel, 1991), whereas this problem is decidable for acyclic terminologies.[2] In this paper, we exhibit another instance of this phenomenon: It is shown that inconsistency w.r.t. acyclic $\mathcal{ALN}$-terminologies is NP-complete, whereas it is PSPACE-complete for general $\mathcal{ALN}$-terminologies (see Table 1).

Nevertheless, the literature points out the need for cyclic definitions of concepts (see, e.g., (MacGregor, 1992; Buchheit et al., 1994; Calvanese, 1996)). For instance, description logics can be used to describe schemas in many semantic and object-oriented data models (see, e.g., (Borgida, 1995; Bergamaschi and Sartori, 1992; Calvanese et al., 1994)). These schemas are often cyclic; e.g., the Boss of an Employee is a Manager who is himself an Employee. Unlike terminologies, which consist of concept definitions, schemas state only necessary (rather than necessary and sufficient) conditions for concepts. In this paper, however, it is shown that the important inference problems for so-called (cyclic) $\mathcal{SL}_{dis}$-schemas, which where introduced in (Buchheit et al., 1994), can be reduced to corresponding problems for (cyclic) $\mathcal{ALN}$-terminologies. Hence, the decision algorithms for terminologies presented below can be used to decide inference problems for schemas. This yields new proofs for the complexity upper bounds for such schemas, and moreover, for a more expressive schema language since it allows for arbitrary number-restrictions. Conversely, existing results for complexity lower bounds of schemas can be used to derive complexity results for terminologies.

In the next section we introduce some basic definitions and state simple properties. In section 3 we characterize the three semantics. The subsequent sections contain characterizations of inconsistency as well as subsumption for all three semantics. Within these sections the results for the greatest fixed-point semantics are presented in detail; the extensions required by the descriptive and least fixed-point semantics are briefly described.

2 BASIC DEFINITIONS

The basic syntactic elements of a concept language are concepts and roles. Let A denote an atomic concept, i.e., a concept name, as well as C, D arbitrary concepts, R a role name, and n a non-negative integer. Then *$\mathcal{ALN}$-concepts* are formed by means of the following syntax rules:

$$C, D \longrightarrow A \mid \neg A \mid C \sqcap D \mid \forall R.C \mid (\geq n\ R) \mid (\leq n\ R)$$

A *(general) $\mathcal{ALN}$-terminology* T consists of a finite set of concept definitions $A = C$ where A denotes an atomic concept and C an $\mathcal{ALN}$-concept. In addition, it is required that for every atomic concept A in T there is at most one concept definition with left-hand side A. We call atomic concepts appearing on the left-hand side of some concept definition in T *defined*, otherwise *primitive*. For an $\mathcal{ALN}$-terminology T only primitive negation is allowed, i.e., if $\neg A$ is a sub-concept in T, then A must be primitive.

A terminology is *cyclic* if there exists at least one

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

[2] One can use the technique described in (Aït-Kaci, 1984) to obtain this result (see also (Hollunder and Nutt, 1990)).

atomic concept which (directly or indirectly) occurs in its own definition. Formally, cycles are defined as follows: Let A denote a defined concept and B an atomic concept. We say that A *directly uses* B if B occurs on the right-hand side of the concept definition $A = C$ in T. Now let *uses* denote the transitive closure of *directly uses*. Then a terminology T is cyclic if there is a defined concept in T that *uses* itself, otherwise T is called *acyclic*.

The following is an example (taken from (Nebel, 1991)) of a cyclic $\mathcal{ALN}$-terminology consisting of only one concept definition:

$$T: \mathsf{Human} = \mathsf{Mammal} \sqcap (\geq 2\ \mathsf{parents}) \sqcap (\leq 2\ \mathsf{parents}) \sqcap \forall \mathsf{parents}.\mathsf{Human}$$

Here, Human is a defined concept, Mammal a primitive concept, and parents a role. Intuitively, this terminology defines human beings as those mammals having exactly two parents all of whom are human beings.

In order to define the semantics of $\mathcal{ALN}$-terminologies formally, we—similarly to the first-order predicate logic—have to introduce the notion "interpretation".

An *interpretation* I consists of a domain $dom(I)$ and a mapping assigning a subset C^I of $dom(I)$ (the *extension of* C) to every atomic concept C as well as a binary relation R^I over $dom(I)$ (the *extension of* R) to every atomic role R. This interpretation is extended to $\mathcal{ALN}$-concepts as defined in Table 2 where $R^I(d) := \{e \in dom(I) \mid (d,e) \in R^I\}$ denotes the set of *R-successors* of d in I. For convenience we write

Table 2: Semantics of $\mathcal{ALN}$-Concepts

Syntax	Semantics
$\neg A$	$dom(I) \setminus A^I$
$C \sqcap D$	$C^I \cap D^I$
$\forall R.C$	$\{d \in dom(I); R^I(d) \subseteq C^I\}$
$(\leq n\ R)$	$\{d \in dom(I); \lvert R^I(d)\rvert \leq n\}$
$(\geq n\ R)$	$\{d \in dom(I); \lvert R^I(d)\rvert \geq n\}$

$\forall W.C$ instead of $\forall R_1.\forall R_2 \cdots \forall R_n.C$, $n \geq 0$, where W denotes the word $R_1 \cdots R_n$. According to this abbreviation, let W^I be the composition $R_1^I \circ \cdots \circ R_n^I$ of the relations R_i^I, $1 \leq i \leq n$. For the empty word, ε^I denotes the identity relation. As for an atomic role, $W^I(d)$ denotes the set of W-successors of d in I.

An interpretation I is a *model* of a terminology T iff $A^I = C^I$ for all concept definitions $A = C$ in T. The *descriptive semantics* of T is defined by the set of all models I of T.

In presence of cyclic dependencies the descriptive semantics does not capture the intuitive meaning of a terminology in any case. Therefore, the meaning of cyclic terminologies is also specified by so-called fixed-point semantics, which have been introduced by B. Nebel (Nebel, 1991). In order to define these semantics, we divide the interpretation I of the terminology T into the *primitive interpretation* J and the interpretation of the defined concepts of T (Human in the example). The primitive interpretation determines the extensions of the primitive concepts and roles in T (Mammal and role in the example). For acyclic terminologies the primitive interpretation J of T can uniquely be extended to a model I of T, i.e., the extensions of the defined concepts in T are uniquely determined by J and T. However, in the presence of cyclic dependencies there may exist several possible extensions of the primitive interpretation J to a model I of T. In such a model the extensions of the defined concepts are fixed-points of the following mapping:

Definition 1.
Let T be a terminology consisting of the concept definitions $A_1 = D_1, \ldots, A_n = D_n$, and let J be a primitive interpretation. The mapping $T_J : (2^{dom(J)})^n \longrightarrow (2^{dom(J)})^n$ is defined as follows: Let $\underline{A} \in (2^{dom(J)})^n$ be a tuple, namely, a tuple of extensions of the defined concepts, and I the interpretation given by J and $\underline{A}$. Then $T_J(\underline{A}) := (D_1^I, \ldots, D_n^I)$. ◇

Note, that for $\mathcal{ALN}$ the mapping T_J is monotonic where the tuples are ordered componentwise by set inclusion. Thus, for every $\mathcal{ALN}$-terminology T and every primitive interpretation J there is always a unique greatest and least fixed-point of T_J. Furthermore, it is not hard to see that if I is a model of T with primitive interpretation J, then the tuple of the extensions of the defined concepts in T is a fixed-point of T_J, i.e., I is specified by J and a fixed-point of T_J.

Now, we are equipped to define the fixed-point semantics mentioned above. The *lfp-semantics* allows only those models of T as admissible models which come from a primitive interpretation J and the least fixed-point of the mapping T_J (*lfp-models*); analogously, the *gfp-semantics* allows only those models of T as admissible models which come from a primitive interpretation J and the greatest fixed-point of the mapping T_J (*gfp-models*). For more information on these semantics see, e.g., (Nebel, 1991; Baader, 1996b; Baader, 1990; Küsters, 1997).

Let A, B be atomic concepts in the terminology T. The concept A is *inconsistent* w.r.t. T (T-inconsistent) and w.r.t. the descriptive semantics (lfp-, gfp-

semantics) iff in every model (lfp-, gfp-model) of T the extension of A is empty. The concept A is *subsumed* by B w.r.t. T and the descriptive semantics (lfp-, gfp-semantics) iff for all models (lfp-, gfp-models) I of T we have: $A^I \subseteq B^I$. In this case, we write $A \sqsubseteq_T B$ ($A \sqsubseteq_{lfp,T} B$, $A \sqsubseteq_{gfp,T} B$).

Since $\mathcal{ALN}$ allows for number-restrictions, primitive negation can be dispensed with using the technique proposed in (Baader, 1996a): The concepts $\neg P$ and P in a terminology can be replaced by $(\leq 0\ R_P)$ and $(\geq 1\ R_P)$, respectively, for a new role name R_P. By this substitution one can reduce inconsistency and subsumption of concepts for $\mathcal{ALN}$-terminologies to $\mathcal{FLN}$-terminologies, that do not allow for primitive negation. Without loss of generality, we therefore restrict our attention to $\mathcal{FLN}$ in the sequel.

In order to characterize the three semantics we first have to normalize the terminology T and then associate a semi-automaton to the normalized terminology: T is called *normalized* if every right-hand side of a concept definition is a finite conjunction of concepts of the form $\forall W.C$ where W is a finite word over the role names in T and C is an atomic concept or a number-restriction. A *semi-automaton* is defined by a finite alphabet Σ, a finite set of states Q, and a finite set $E \subseteq Q \times \Sigma^* \times Q$ of *word-transitions*.

Now, the (non-deterministic) semi-automaton $\mathcal{A}_T = (\Sigma, Q, E)$ of the *normalized* $\mathcal{FLN}$-terminology T is defined as follows: The alphabet Σ of $\mathcal{A}_T$ consists of the role names occurring in T; the atomic concepts and number-restrictions of T are the states $q \in Q$ of $\mathcal{A}_T$; a concept definition $A = \forall W_1.A_1 \sqcap \cdots \sqcap \forall W_k.A_k$ gives rise to k word-transitions, where the transition from A to A_i is labeled with the word W_i (for more details see (Baader, 1996b; Küsters, 1997; Nebel, 1990b)). Note that in a semi-automaton, word-transitions can easily be eliminated by replacing each of these transitions by a sequence of new introduced transitions (labeled with letters) using new states. Therefore, if needed, $\mathcal{A}_T$ is (w.l.o.g.) supposed to be a *semi-automaton without word-transitions*, i.e., $E \subseteq Q \times (\Sigma \cup \{\varepsilon\}) \times Q$. For the terminology of the example we obtain the semi-automaton $\mathcal{A}_T$ shown in Figure 1 where $\mathsf{M} = \mathsf{Mammal}$, $\mathsf{H} = \mathsf{Human}$, and $\mathsf{p} = \mathsf{parents}$.

Let A be an atomic concept and C an atomic concept or a number-restriction. Then $L_{\mathcal{A}_T}(A, C)$ (resp. $L(A, C)$ if the relationship to $\mathcal{A}_T$ is clear from the context) denotes the set of words (over the role names in T) that are labels of paths from A to C in $\mathcal{A}_T$. We also have to consider infinite words, i.e., elements of Σ^ω. The set $U_{\mathcal{A}_T}(A)$ (resp. $U(A)$) denotes the set of (finite and infinite) words which are labels of infinite

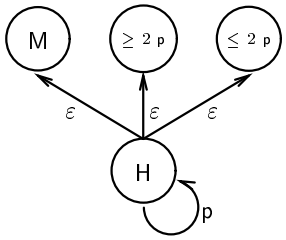

Figure 1: The Semi-Automaton $\mathcal{A}_T$

paths in $\mathcal{A}_T$ starting from A. Note that since $\mathcal{A}_T$ allows for ε-transitions, $U(A)$ may contain finite words. For $L \subseteq \Sigma^*$ and $R \in \Sigma$ the set $L \cdot R$ is defined as $\{W \cdot R \mid W \in L\}$ where "$\cdot$" denotes concatenation of words; furthermore $L \cdot R^{-1} := \{W \in \Sigma^* \mid W \cdot R \in L\}$.

In order to characterize inconsistency and subsumption, the notion of "*requiring*" is very useful. This notion is quite similar to the notion "active chain of roles" introduced in (Calvanese, 1996). Due to number-restrictions, atomic concepts can "require" chains of role successors which have to start from every instance of such a concept.

Definition 2 (require).
Let T be an $\mathcal{FLN}$-terminology, $\mathcal{A}_T$ the corresponding semi-automaton, and A an atomic concept in T. Furthermore, let $W = R_1 \cdots R_n$ denote 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 $k_{i+1} \geq 1$ such that $V R_{m+1} \cdots R_i \in L(A, (\geq k_{i+1}\ R_{i+1}))$. For infinite words this notion is defined analogously. If $V = \varepsilon$ we say "W is required by A" instead of "W is required by A starting from ε". $\diamond$

It is easy to see that if an instance of A in definition 2 has a V-successor w.r.t. a model I of T, then it must also have a W-successor. Using this observation it follows that every instance of Human in the example has an infinite chain of ancestors: Because of $\mathsf{parents}^j \in L(\mathsf{Human}, (\geq 2\ \mathsf{parents}))$ for all $j \geq 0$, every word $\mathsf{parents}^j$ is required by 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:

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

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 $\mathcal{FL}_0$. We just have to add conditions for number-restrictions which are analogous to the conditions for primitive concepts. The characterization for the gfp-semantics is as follows:

Theorem 3 (gfp-semantics).
Let T be an $\mathcal{FLN}$-terminology, $\mathcal{A}_T$ the corresponding semi-automaton, I a gfp-model of T, and A an atomic concept 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, and for all words $W \in L_{\mathcal{A}_T}(A, C)$ it holds that $d \in (\forall W.C)^I$. □

For lfp- and descriptive semantics an additional condition must be satisfied, which, however, coincides with the one for $\mathcal{FL}_0$ (Baader, 1996b).

Descriptive semantics: For the descriptive semantics, relationships between defined concepts also have to be taken into account. In order to characterize the descriptive semantics formally, we must introduce some more notions: As already mentioned, for a model I of T (with primitive interpretation J) the tuple of extensions of the defined concepts is a fixed-point of T_J. We can describe this fixed-point as greatest fixed-point $\underline{A}\text{-}gfp(T_J)$ of T_J that is less or equal a tuple $\underline{A}$. Since T_J is monotonic, it can be shown that such a fixed-point exists if $T_J(\underline{A}) \subseteq \underline{A}$ (see (Baader, 1996b; Küsters, 1997) for details). Using this observation, it is easy to see that for every model I with primitive interpretation J there is a tuple $\underline{A}$ such that $\underline{A}\text{-}gfp(T_J)$ exists and I consists of J and $\underline{A}\text{-}gfp(T_J)$. We refer to the extension of a defined concept B in $\underline{A}$ by $(\underline{A})_{index(B)}$. Now let I be a model of T defined by J and $\underline{A}\text{-}gfp(T_J)$. With this notation, Theorem 3 is extended by the following condition, which additionally must be satisfied (Küsters, 1997):

> For all defined concepts B and all $W \in L(A, B)$ it holds $W^I(d) \subseteq (\underline{A})_{index(B)}$.

Lfp-semantics: For the lfp-semantics there is an additional condition to be satisfied as well. This condition intuitively prohibits infinite role chains starting from $d \in A^I$. Formally, Theorem 3 is extended by the following condition:

> For all infinite paths of the form $A, W_1, C_1, W_2, C_2, \ldots$ and all individuals $d_1, d_2, d_3, \ldots \in dom(I)$ there is an $n \geq 1$ such that $(d_{n-1}, d_n) \notin W_n^I$ where $d_0 = d$.

The characterizations provide a more intuitive way to understand and to compare the different semantics. This can make the decision easier which of the semantics to prefer in a given representation task. Furthermore, we can employ these descriptions to characterize and decide inconsistency and subsumption.

4 CHARACTERIZING INCONSISTENCY

In $\mathcal{FL}_0$, inconsistent concepts only occur for lfp-semantics. Due to conflicting number restrictions, inconsistent $\mathcal{FLN}$-concepts may also occur for the gfp- and descriptive semantics.

Before formulating the characterization of inconsistency, we introduce the notion "*exclusion set*", which will become important in the construction of decision procedures both for inconsistency and for subsumption. Intuitively, an exclusion set describes a set of $\mathcal{A}_T$-states that require words leading to conflicting number-restrictions.

Definition 4 (exclusion set).
Let T be an $\mathcal{FLN}$-terminology and $\mathcal{A}_T = (\Sigma, Q, E)$ the corresponding semi-automaton without word-transitions. 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 $m_i \geq 1$, and $l > r$ such that for $F_i := next_\varepsilon(F_{i-1}, R_i)$, $1 \leq i \leq n$, the following holds: $(\geq m_i\ R_i) \in F_{i-1}$ for all $1 \leq i \leq n$ and $(\geq l\ R), (\leq r\ R) \in F_n$. ◇

Now we are ready to formulate

Theorem 5 (inconsistency).
Let T be an $\mathcal{FLN}$-terminology, $\mathcal{A}_T$ the corresponding semi-automaton (without word-transitions), and A an atomic concept in T. Then the following statements are equivalent:

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

Proof sketch of Theorem 5 (for details see (Küsters, 1997)): It is easy to see that statement 2. implies inconsistency of A. Furthermore, proving equivalence of 2. and 3. is straightforward. The non-trivial part of the proof is to show that statement 2. is necessary for inconsistency. We show that, if statement 2. does not hold, then it is possible to define a canonical gfp-model I of T in which the extension of A is non-empty. More precise, I is a finite branching tree-model of T and the root d_0 of this tree is contained in A^I. In order to construct the primitive interpretation J of I, one first satisfies the $\geq$-restrictions required by A. Then, if 2. does not hold, it can be shown that the $\leq$-restrictions stated for A are not violated. To satisfy the $\geq$-restrictions required by A, we enumerate the (finite) words required by A, where the enumeration is ordered by the length of the words (ε, parents, parents parents, ... in the example). Starting from d_0 the tree is inductively extended according to the required words of the enumeration (in the example this leads to a binary tree since for every individual two parents are required by Human). Formally, the canonical model is defined as follows:

Definition 6 (canonical gfp-model).
Let T be an $\mathcal{FLN}$-terminology, $\mathcal{A}_T = (\Sigma, Q, E)$ the corresponding semi-automaton, and A an atomic concept in T. The primitive canonical interpretation $J = J(A, d_0)$ for A and the individual d_0 is defined as follows:

Let $W_0, W_1, W_2, W_3, \ldots$ be the infinite (w.o.l.g.; the finite case can be dealt with analogously) enumeration of the words required by A such that $W_i \neq W_j$ for all $i \neq j$, and $i < j$, implies $|W_i| \leq |W_j|$. Note that if one word is required by A, then all prefixes of this word are also required. Consequently, $W_0 = \varepsilon$ and $|W_i| < |W_{i+1}|$ implies $|W_{i+1}| = |W_i| + 1$. We define J inductively:

J_0: $dom(J_0) := \{d_0\}$; $R^{J_0} := \emptyset$ for all roles R in T; the extensions of the primitive concepts are not defined here, they will be defined for J.

J_{i+1}: For W_{i+1} there is exactly one $j < i+1$ and one role R in T such that $W_{i+1} = W_j R$. Let $m \geq 1$ be maximal such that $W_j \in L(A, (\geq m\ R))$, i.e., there is no $m' > m$ with $W_j \in L(A, (\geq m'\ R))$ (m exists since W_{i+1} is required by A, and T contains only a finite number of number-restrictions). Let $K_{i+1} := \{d_1, \ldots, d_r\} := W_j^{J_i}(d_0)$ (where $r \geq 0$) be the set of W_j-successors of d_0 in J_i. Furthermore, let $e_1^1, \ldots, e_m^1, \ldots, e_1^r, \ldots, e_m^r$ be $(r \cdot m)$ new individuals. We extend J_i to J_{i+1} by

$dom(J_{i+1}) := dom(J_i) \dot\cup \{e_1^1, \ldots, e_m^1, \ldots, e_1^r, \ldots, e_m^r\}$; $S^{J_{i+1}} := S^{J_i}$ for all roles $S \neq R$ and $R^{J_{i+1}} := R^{J_i} \dot\cup \{(d_k, e_l^k) \mid 1 \leq k \leq r, 1 \leq l \leq m\}$.

Now, J is defined as follows:

$dom(J) := \bigcup_{i \geq 0} dom(J_i)$; $S^J := \bigcup_{i \geq 0} S^{J_i}$ for all roles S in T; for all primitive concepts P and individuals $d \in dom(J)$ let: $d \in P^J$ iff there exists a word $W \in \Sigma^*$ such that $W \in L(A, P)$ and $d \in W^J(d_0)$.

The *canonical gfp-model* $I = I(A, d_0)$ for A and d_0 is defined as the gfp-model of $J(A, d_0)$ and T, i.e., I is defined by the primitive interpretation $J(A, d_0)$ and the greatest fixed-point of $T_{J(A,d_0)}$. ◇

Now, the following Propositions (Küsters, 1997) completes the proof of Theorem 5.

Proposition 7.
If statement 2. in Theorem 5 does not hold, then $d_0 \in A^I$ where $I = I(A, d_0)$. □

In general, for a given set of states it is a non-trivial problem to decide whether a given set of states is an exclusion set. The complexity of this problem is due to the fact that role successors can be required. It is not hard to specify a PSPACE-algorithm deciding the property of being an exclusion set. Thus, by Theorem 5 inconsistency is decidable using polynomial space (see (Küsters, 1997) for details). In (Calvanese, 1996), it has been shown that consistency of concepts w.r.t. $\mathcal{AL}$-schemas (see section 6) and descriptive semantics is PSPACE-complete. Using Theorem 17 this shows

Proposition 8.
Inconsistency in general $\mathcal{ALN}$-terminologies with respect to gfp-semantics is PSPACE-complete. □

Descriptive semantics: Every gfp-model of a terminology is also a model with respect to the descriptive semantics. Furthermore, the extensions of defined concepts w.r.t. gfp-semantics are maximal. Thus, an atomic concept A is inconsistent w.r.t. the gfp-semantics iff it is inconsistent w.r.t. the descriptive semantics. Consequently, Theorem 5 also holds for the descriptive semantics and inconsistency for the descriptive semantics is PSPACE-complete.

Lfp-semantics: Theorem 5 does not hold for the lfp-semantics due to the effect of infinite role chains. For lfp-semantics, inconsistency of A is equivalent to the disjunction of statement 2. in Theorem 5 and the following condition:

> There is a (finite or infinite) word W required by A in $U(A)$.

Definition 4 has to be adjusted accordingly, i.e., additionally to what is stated in this definition, F_0 is also an exclusion set if there is an *infinite* chain of sets $F_i := next_\varepsilon(F_{i-1}, R_i)$, $R_i \in \Sigma$, and numbers $m_i \geq 1$, $i \geq 1$, such that $(\geq m_i\ R_i) \in F_{i-1}$ for all $i \geq 1$, or if there is a *finite* chain $F_0, \ldots, F_n$ of sets satisfying analogous conditions where, in addition, F_n contains an atomic concept C in an ε-cycle, i.e., there is a non-empty path in $\mathcal{A}_T$ from C to C labeled with ε.

As for the other two semantics, one can construct a PSPACE-algorithm that tests whether a set of states is an exclusion set. Thus, inconsistency w.r.t. lfp-semantics is decidable using polynomial space (see (Küsters, 1997) for details). Unlike gfp-semantics, PSPACE-completeness of inconsistency for lfp-semantics is not an immediate consequence of PSPACE-completeness of inconsistency of concepts in $\mathcal{AL}$-schemas w.r.t. descriptive semantics (Calvanese, 1996). The proof of this complexity result for $\mathcal{AL}$-schemas is by reducing the validity of an (arbitrary) quantified Boolean formulae q to consistency of an atomic concept C_q w.r.t. a cyclic $\mathcal{AL}$-schema S_q. The problem is that S_q may require infinite chains starting from C_q although q is valid, i.e., C_q may be inconsistent w.r.t. lfp-semantics although q is valid. Fortunately, one can avoid such infinite chains in S_q such that C_q is consistent in S_q w.r.t. descriptive semantics (or even gfp-semantics) iff C_q is consistent in S_q w.r.t. lfp-semantics. This provides us with PSPACE-completeness of inconsistency in general $\mathcal{ALN}$-terminologies w.r.t. lfp-semantics (see (Küsters, 1997) for details).

Finally, we consider weak-acyclic[3] $\mathcal{ALN}$-terminologies and sketch the proof of

Proposition 9.
Consistency for concepts in *weak-acyclic* $\mathcal{ALN}$-terminologies is NP-complete with respect to the gfp-, lfp-, and descriptive semantics. □

We only consider gfp- and descriptive semantics. If T is a weak-acyclic $\mathcal{ALN}$-terminology, then the property of being an exclusion set is decidable by an NP-algorithm since for the sets F_i as defined in Definition 4 one can show that $F_{n+1} = \emptyset$, where n denotes the number of states in $\mathcal{A}_T$. From the proof in (Buchheit et al., 1994), which verifies NP-completeness of *validity* of $\mathcal{SL}_{dis}$-schemas, it follows that *consistency* of concepts with respect to $\mathcal{SL}_{dis}$-schemas is NP-complete as well. Now, since consistency w.r.t. $\mathcal{SL}_{dis}$-schemas can be reduced to consistency w.r.t. $\mathcal{ALN}$-terminologies (Theorem 17), this provides us with the complexity lower bounds stated in the above Proposition (see (Küsters, 1997) for details).

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

5 CHARACTERIZING SUBSUMPTION

For the language $\mathcal{FL}_0$, subsumption has been characterized in (Baader, 1996b) via inclusion of regular languages. With respect to gfp-semantics this characterization is as follows:

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

Due to conflicting number restrictions, the right-hand side of this equivalence is not necessary for A to be subsumed by B w.r.t. $\mathcal{FLN}$-terminologies: The atomic concept A may be "excluded" by a word that is contained in $L_{\mathcal{A}_T}(B, P)$, but not in $L_{\mathcal{A}_T}(A, P)$. Intuitively, the words W excluding A are exactly those words such that $A \sqsubseteq_{gfp,T} (\forall W.\bot)$.[4] To simplify the formal definition of the notion "exclusion" we may (without loss of generality) assume that an $\mathcal{FLN}$-terminology contains no $\leq$-restrictions of the form $(\leq 0\ R)$ since such a term can be substituted by $\forall R.\bot$. In the following we consider only terminologies without $(\leq 0\ R)$ and call them $\mathcal{FLN}^r$-terminologies.

Definition 10 (exclusion).
Let T be an $\mathcal{FLN}^r$-terminology, $\mathcal{A}_T = (\Sigma, Q, E)$ the corresponding semi-automaton, and A an atomic concept in T. The word $W \in \Sigma^*$ *excludes* A iff there exist a prefix $V \in \Sigma^*$ of W, a word $V' \in \Sigma^*$ as well as conflicting number-restrictions $(\geq l\ R)$, $(\leq r\ R)$, $l > r$, such that $VV' \in L(A, (\geq l\ R)) \cap L(A, (\leq r\ R))$ and VV' is required by A starting from V. Let E_A denote the set of words excluding A (A-*excluding words*). ◇

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, i.e., $A \sqsubseteq_{gfp,T} (\forall W.\bot)$. 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 the $\mathcal{FLN}^r$-terminology with corresponding automaton $\mathcal{A}_T$ given in Figure 2. Furthermore, let I be a gfp-model of T and d an individual with $d \in A^I$. Assume $e \in (RS)^I(d)$. Since $RS \in L_{\mathcal{A}_T}(A, (\geq 1\ Q))$, there must be a Q-successor f of e. Further, since

[4] The symbol $\bot$ denotes the empty concept, which is equivalent to, e.g., $(\geq 2\ R) \sqcap (\leq 1\ R)$.

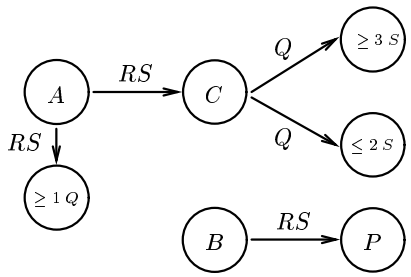

Figure 2: A Terminology Defined by Its Automaton

$RSQ \in L_{\mathcal{A}_T}(A, (\geq 3\ S)) \cap L_{\mathcal{A}_T}(A, (\leq 2\ S))$, the characterization of the gfp-semantics yields the contradiction: $f \in (\geq 3\ S)^I$ and $f \in (\leq 2\ S)^I$. Thus, d cannot have an RS-successor.

This proves that $A \sqsubseteq_{gfp,T} (\forall RS.\bot)$, and thus, $A \sqsubseteq_{gfp,T} (\forall RS.P)$. Consequently, $A \sqsubseteq_{gfp,T} B$. Because of $L_{\mathcal{A}_T}(B, P) \not\subseteq L_{\mathcal{A}_T}(A, P)$, the example shows that due to excluding words (RS in the example) the right-hand side of (1) is not necessary for subsumption w.r.t. $\mathcal{FLN}^r$.

Using E_A, we are able to characterize subsumption w.r.t. $\mathcal{FLN}^r$-terminologies and the gfp-semantics as follows:

Theorem 11 (subsumption).
Let T be an $\mathcal{FLN}^r$-terminology, $\mathcal{A}_T$ the corresponding semi-automaton, and A, B atomic concepts 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;
2. $L(B, (\geq l\ R)) \subseteq \bigcup_{r \geq l} L(A, (\geq r\ R)) \cup E_A$ for all $\geq$-restrictions of the form $(\geq l\ R)$ in T where $l > 0$; and
3. $L(B, (\leq l\ R)) \subseteq \bigcup_{r \leq l} L(A, (\leq r\ R)) \cup (E_A \cdot R^{-1})$ for all $\leq$-restrictions of the form $(\leq l\ R)$ in T. □

Proof sketch of Theorem 11 (for details see (Küsters, 1997)): Using that $W \in E_A$ implies $A \sqsubseteq_{gfp,T} \forall W.\bot$ and that $W \in (E_A \cdot R^{-1})$ implies $A \sqsubseteq_{gfp,T} \forall W.(\leq 0\ R)$ (thus, $A \sqsubseteq_{gfp,T} \forall W.(\leq l\ R)$ for all $l \geq 0$), the if direction of the claim is easy to show. The non-trivial part of the proof is the only-if direction. The idea is as follows: If one of the inclusions does not hold, then we define an extended canonical gfp-model I such that $d_0 \in A^I \setminus B^I$. For this purpose, the canonical model for A and d_0 is extended: If an inclusion is invalid because of the word W, i.e., this word is contained in the left-hand side but not in the right-hand side of the inclusion relation, the tree of the canonical model with root d_0 is extended such that there is a path in the tree starting at d_0 labeled with W. If W violates the inclusion relation of a $\leq$-restriction $(\leq l\ R)$, then the path with label W additionally has to be extended by $(l+1)$ R-successors; more precisely by $(l+1)$ minus the already existing R-successors. Finally, this tree must be completed such that the condition formulated in Theorem 3 is satisfied w.r.t. A and d_0. Note that, because of the extension of the canonical model, these conditions initially need not be satisfied. Formally, the extended model is defined as follows:

Definition 12 (extended canonical gfp-model).
Let T be an $\mathcal{FLN}^r$-terminology, $\mathcal{A}_T$ the corresponding semi-automaton, A an atomic concept in T, W a word in Σ^*, $r \in \mathbb{N}$ and $R \in \Sigma$. For $r = 0$ we denote the *extended primitive canonical interpretation* by $J' = J(A, d_0, W)$; for $r > 0$ it is denoted by $J' = J(A, d_0, W, R, r)$. Let $U_1, U_2, U_3, \ldots$ be an enumeration of all words in Σ^* where $U_i \neq U_j$ for $i \neq j$ and $|U_i| \leq |U_j|$ for all $i < j$. Let $J = J(A, d_0)$ be the primitive canonical interpretation (introduced in Definition 6) and $I = I(A, d_0)$ the corresponding gfp-model of T. We define J' inductively as follows:

J_0: If W is required by A, then there is an individual $d_1 \in dom(J)$ such that $d_1 \in W^I(d_0)$ (see above). If $(r - |R^J(d_1)|) \geq 0$, then let $k := r - |R^J(d_1)|$ else $k := 0$. Let $f_1, \ldots, f_k$ be new, pairwise distinct individuals. We define:

$dom(J'_0) := dom(J) \,\dot\cup\, \{f_1, \ldots, f_k\}$; $S^{J'_0} := S^J$ for all roles $S \neq R$; $R^{J'_0} := R^J \,\dot\cup\, \{(d_1, f_i) \mid 1 \leq i \leq k\}$.

If W is not required by A, then there is a prefix U of W of maximal length which is required by A. Consequently, there is a $d_1 \in dom(J)$ such that $d_0 U^J d_1$. Furthermore, since W is not required by A, there is a $V \in \Sigma^+$, $V = R_1 \cdots R_n$, $n \geq 1$, where $W = UV$. Let $k := r$, and $d_2, d_3, \ldots, d_{n+1}, f_1, \ldots, f_k$ be new, pairwise distinct individuals. We define:

$dom(J'_0) := dom(J) \,\dot\cup\, \{d_2, \ldots, d_{n+1}, f_1, \ldots, f_k\}$; $S^{J'_0} := S^J \,\dot\cup\, \{(d_i, d_{i+1}) \mid 1 \leq i \leq n,\ S = R_i\} \,\dot\cup\, \{(d_{n+1}, f_i) \mid 1 \leq i \leq k,\ S = R\}$ for all roles S in T.

The extensions of the primitive concepts are defined only for J'.

J'_{i+1}: We define $M \subseteq dom(J'_i) \times \Sigma \times (\mathbb{N} \setminus \{0\})$ by $(g, S, m) \in M$ iff $g \in U_{i+1}^{J'_i}(d_0)$, $U_{i+1} \in L(A, (\geq m\ S))$, $m \geq 1$, m maximal with this property (i.e., there is no $m' > m$ such that $U_{i+1} \in L(A, (\geq m'\ S))$), and $|S^{J'_i}(g)| < m$. Let $(g_1, S_1, m_1), \ldots, (g_s, S_s, m_s)$ denote the triples in M, $s \geq 0$. Intuitively, M contains a triple (g, S, m) if for g in the corresponding tree to J'_i

additionally $(m-|S^{J'_i}(g)|)$ S-successors must be added in order to satisfy the conditions for the $\geq$-restrictions in Theorem 3 w.r.t. d_0 and A.

For this purpose, let $l_j := |S_j^{J'_i}(g_j)|$ (note that $l_j < m_j$) for all $1 \leq j \leq s$ and $D := \{g_1^1, \ldots, g_{m_1-l_1}^1, \ldots, g_1^s, \ldots, g_{m_s-l_s}^s\}$ a set of new, pairwise distinct individuals. We define:

$dom(J'_{i+1}) := dom(J'_i) \dot\cup D$; $S^{J'_{i+1}} := S^{J'_i} \dot\cup \bigcup_{1\leq j\leq s, S=S_j}\{(g_j, g_1^j), \ldots, (g_j, g_{m_j-l_j}^j)\}$ for all roles S in T.

Now the extended primitive canonical interpretation J' is defined as follows:

$dom(J') := \bigcup_{i\in I\!N} dom(J'_i)$; $S^{J'} := \bigcup_{i\in I\!N} S^{J'_i}$ for all roles S in T; for all primitive concepts P and individuals d let $d \in P^{J'}$ iff there is a $V' \in L(A, P)$ such that $d \in V'^{J'}(d_0)$.

The *extended canonical gfp-models* $I(A, d_0, W)$ and $I(A, d_0, W, R, r)$ are the gfp-models of T and $J' = J(A, d_0, W)$ and $J' = J(A, d_0, W, R, r)$, respectively. $\diamond$

Now, if one of the inclusions in Theorem 11, 1. or 2. is violated by W, i.e., W is contained in the left-hand side but not in the right-hand side, then it can be shown that $d_0 \in A^{I'} \setminus B^{I'}$ where $I' := I(A, d_0, W)$. Furthermore, if in 3. the inclusion for the $\leq$-restriction $(\leq l\ R)$ is violated by W, then it can be shown that $d_0 \in A^{I'} \setminus B^{I'}$ where $I' := I(A, d_0, W, R, l+1)$. Since the proof that I' is the appropriate counter model for the subsumption of A and B is lengthy and technical, it is omitted here (see (Küsters, 1997) for details).

The inclusions in Theorem 11 can be decided by a decision procedure for inclusions of regular languages known from automata theory. Due to lack of space, we restrict our attention to the inclusions for primitive concepts.

In automata theory, the inclusion $L_1 \subseteq L_2$ of two regular languages is usually reduced to the emptiness problem of the finite automaton corresponding to the language $L_1 \cap \overline{L_2}$. It is known that this problem can be decided using space polynomial in the size of the automata $\mathcal{A}_1$, $\mathcal{A}_2$ corresponding to L_1 and L_2, respectively. Such an algorithm must cope with the complement $\overline{L_2}$ of L_2, and therefore, simulates the powerset automaton of L_2, which may be exponential in the size of $\mathcal{A}_2$.

In order to apply this technique from automata theory, we have to show that the language E_A is regular:

Definition 13 (reaching an exclusion set).
Let T be an $\mathcal{FLN}^r$-terminology and $\mathcal{A}_T$ the corresponding semi-automaton without word-transitions. There is an *exclusion set reachable by a word* $W \in \Sigma^*$ starting from the atomic concept A if there is a prefix V of W such that $next_\varepsilon(A, V)$ is an exclusion set. $\diamond$

Now, E_A can be described as follows (Küsters, 1997):

Lemma 14.
Let T be an $\mathcal{FLN}^r$-terminology, $\mathcal{A}_T$ the corresponding semi-automaton without word-transitions, and A an atomic concept in T. Then the following holds:

$$E_A = \{W \in \Sigma^* \mid \text{an exclusion set is reachable by } W \text{ starting from } A\}.$$

□

Using this lemma, one can construct a finite automaton $\mathcal{B}$ corresponding to E_A (Küsters, 1997). Roughly speaking, this automaton is an extension of the powerset automaton of $\mathcal{A}_T$ where the exclusion sets are certain states in $\mathcal{B}$. This shows that E_A is regular. Furthermore, $L(B, P) \subseteq L(A, P) \cup E_A$ can be decided in space polynomial in the size of $\mathcal{A}_T$ since the powerset automaton of $\mathcal{A}_T$, which is used to cope with $\overline{L(A, P) \cup E_A}$, can be simulated (see (Küsters, 1997) for details and for the inclusions in 2. and 3. of Theorem 11).

PSPACE-completeness of subsumption in general $\mathcal{FL}_0$-terminologies (Baader, 1996b) implies

Proposition 15.
Subsumption in general $\mathcal{ALN}$-terminologies with respect to gfp-semantics is PSPACE-complete. □

Descriptive semantics: For the descriptive semantics, the notion "exclusion" is specified as in Definition 10. However, Theorem 11 has to be extended by the following additional condition:

4. For all defined concepts C in T and all infinite paths of the form $B, U_0, C, U_1, C, U_2, C, \ldots$ there is a $k \geq 0$ such that $U_0 \cdots U_k \in L(A, C) \cup E_A$.

Apart from the union with E_A on the right-hand side of the element relationship, this condition coincides with the one for $\mathcal{FL}_0$ in (Baader, 1996b) (for $\mathcal{FL}_0$ the set E_A is empty).

Based on the power-set automata and the exclusion sets of $\mathcal{A}_T$ one can decide condition 4. using polynomial space (Küsters, 1997). For $\mathcal{FL}_0$-terminologies and the descriptive semantics PSPACE-completeness of subsumption is still an open problem. However, PSPACE-completeness of inconsistency for the descriptive semantics in $\mathcal{ALN}$-terminologies provides us

with PSPACE-completeness of subsumption in $\mathcal{ALN}$-terminologies w.r.t. descriptive semantics since inconsistency can be reduced to subsumption.

Lfp-semantics: Similar to the notion "exclusion set" we have to extend the notion "exclusion" for lfp-semantics due to the special treatment necessary for infinite role chains. Beside the condition for exclusion given in Definition 10, for the lfp-semantics a word $W \in \Sigma^*$ also excludes the atomic concept A if

> there is a (finite or infinite) word $X \in U(A)$ as well as a word $V \in \Sigma^*$ which is prefix of W and X such that X is required by A starting from V.

In addition, the notion of "exclusion" must be extended to infinite words $W \in \Sigma^\omega$: W *excludes* A if there is a finite prefix of W which excludes A or if $W \in U(A)$. As for the other two semantics, we refer to the set of finite A-excluding words by E_A. Moreover, the set of finite and infinite A-excluding words is denoted $E_{A,\omega}$. Now, for the lfp-semantics Theorem 11 is extended by the following condition:

4. $U(B) \subseteq U(A) \cup E_{A,\omega}$.

Again, based on the power-set automata and the exclusion sets of $\mathcal{A}_T$ one can decide this condition using polynomial space (Küsters, 1997). Furthermore, PSPACE-completeness of subsumption for $\mathcal{FL}_0$-terminologies with respect to lfp-semantics implies PSPACE-completeness of subsumption even for $\mathcal{ALN}$-terminologies.

6 $\mathcal{SLN}$-SCHEMAS AND $\mathcal{ALN}$-TERMINOLOGIES

In (Buchheit et al., 1994), terminologies are divided into a schema and a view part. The schema merely restricts the number of admissible models of a terminology. Therefore, the meaning of schemas is captured by descriptive semantics. In the view part of a terminology, concepts are *defined* with the help of schema concepts. For this reason, fixed-point semantics is used for this part.

Knowledge and database engineers are interested in validity of schemas as well as subsumption w.r.t. schemas. In (Buchheit et al., 1994), $\mathcal{SL}_{dis}$-schemas have been introduced, which can express constraints frequently occurring in the static part of object-oriented database schemas. Furthermore, a special PSPACE-decision algorithm has been developed for deciding (local) validity of these schemas. In (Küsters, 1997), it is shown that inconsistency, validity, and subsumption for $\mathcal{SLN}$-schemas (and hence, for $\mathcal{SL}_{dis}$-schemas) can be reduced to corresponding problems for $\mathcal{ALN}$-terminologies. As mentioned above, this allows us to prove some hardness results for $\mathcal{ALN}$-terminologies.

In the following, we have a closer look at this reduction by showing how to construct a terminology T_S from a given schema S and by considering the relationship between T_S and S w.r.t. inconsistency and subsumption. First, however, we must introduce schemas formally.

An *$\mathcal{SLN}$-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. To be more precise, let A, B be atomic concepts, R a role name, and n a non-negative integer. Concept inclusions are of the form $A \sqsubseteq C$ where C denotes a concept of the form $B \mid \neg B \mid \forall R.B \mid (\geq n\ R) \mid (\leq n\ R)$, and role inclusions are of the form $R \sqsubseteq A \times B$. An atomic concept A with at least one concept inclusion of the form $A \sqsubseteq C$ in S is called *defined*, and otherwise *primitive*. Analogously, we refer to *defined* and *primitive* roles. Note that here we allow for primitive negation only, i.e., for concept inclusions $A \sqsubseteq \neg B$ where B is primitive. Using so-called *$\mathcal{ALN}$-schemas*, which allow for arbitrary $\mathcal{ALN}$-concepts (rather than "flat" $\mathcal{ALN}$-concepts) on the right-hand side of a concept inclusion, does not increase the expressive power (Küsters, 1997). Furthermore, $\mathcal{SL}_{dis}$-schemas introduced in (Buchheit et al., 1994) coincide with $\mathcal{SLN}$-schemas (w.r.t. expressive power) apart from the fact that they only allow for number restrictions of the form $(\leq 1\ R)$ and $(\geq 1\ R)$. Finally, $\mathcal{AL}$-schemas correspond to $\mathcal{SL}_{dis}$-schemas in which role inclusions are not allowed.

An interpretation I is a *model* of S if all concept inclusions $A \sqsubseteq C$ and all role inclusions $R \sqsubseteq A \times B$ in S are satisfied, i.e., $A^I \subseteq C^I$ 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 atomic concept in S is consistent. We call a schema S *valid* if there is a model I with $A^I \neq \emptyset$ for *every* atomic concept A in S. The following fact (Buchheit et al., 1994; Küsters, 1997) can be used to reduce validity of schemas to consistency of terminologies:

> An $\mathcal{SLN}$-schema ($\mathcal{SL}_{dis}$-schema) is valid iff it is locally valid. (2)

Now, we derive an $\mathcal{ALN}$-terminology T_S from an $\mathcal{SLN}$-schema S such that T_S behaves like S w.r.t. consistency 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 dispensed with. The formal definition of S' is as follows:

The schema S' contains all concept inclusions of S of the form $A \sqsubseteq B$, $A \sqsubseteq \neg B$, $A \sqsubseteq \forall R.B$, and $A \sqsubseteq (\leq n\, R)$. For every $A \sqsubseteq (\geq n\, R) \in S$, $n \geq 1$, we distinguish two cases: (a) R is not defined in S; (b) R is defined in S. In case (a), $A \sqsubseteq (\geq n\, R)$ is contained in S'. In case (b), S' contains $A \sqsubseteq (\geq n\, R) \sqcap \forall R.C_2 \sqcap C_1$ for every $R \sqsubseteq C_1 \times C_2 \in S$. The schema S' contains no other inclusions than these, in particular, no role inclusions.

Definition 16 (the terminology T_S).
Let S be an $\mathcal{SLN}$-schema and S' as defined above. For every defined concept A in S a concept definition for A in T_S is constructed as follows:
Let $A \sqsubseteq C_1, \ldots, A \sqsubseteq C_n$ be all concept inclusions for the defined concept A in S'. Let $\overline{A}$ be a new (primitive) concept. Then the concept definition for A in T_S is of the form $A = \overline{A} \sqcap C_1 \sqcap \cdots \sqcap C_n$. ◇

Obviously, one can compute T_S in time linear in the size of S. The relationship between S and T_S is described by the following theorem (Küsters, 1997):

Theorem 17.
Let S be an $\mathcal{SLN}$-schema and T_S the corresponding $\mathcal{ALN}$-terminology. For all atomic 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 $A \sqsubseteq_{gfp,T_S} B$). □

Consequently using the complexity results presented in section 4 and 5, consistency—and with (2) also (local) validity—as well as subsumption w.r.t. descriptive and gfp-semantics of $\mathcal{SLN}$- and $\mathcal{ALN}$-schemas are decidable using polynomial space. This result not only extends results for complexity upper bounds of $\mathcal{SL}_{dis}$-schemas to $\mathcal{SLN}$-schemas, but also uses well-known results and techniques from automata theory, rather than defining special graphs as in (Buchheit et al., 1994). Furthermore, it clarifies the relationship between schemas and terminologies for the considered languages: $\mathcal{ALN}$-terminologies are at least as expressive as $\mathcal{SLN}$-schemas w.r.t. the important inference problems. Finally, it should be noted that the complexity of subsumption w.r.t. schemas is only due to testing consistency of concepts (Buchheit et al., 1994). Hence, disallowing concept forming operators that enable the definition of inconsistent concepts (number restrictions and primitive negation) makes reasoning tractable, whereas for $\mathcal{FL}_0$-terminologies subsumption is already PSPACE-complete (w.r.t. fixed-point semantics).

7 CONCLUSION

Cyclic terminologies are a natural way to describe the terminological knowledge of a problem domain. Unlike the case of first-order predicate logic, the transitive closure of relations is expressible using the gfp-semantics. Furthermore, as already pointed out, cyclic definitions occur frequently, e.g., in constraints for semantic and object-oriented data models. The previous section has shown that general $\mathcal{ALN}$-terminologies are expressive enough to capture important parts of such constraints.

Neither the descriptive nor the fixed-point semantics capture the intuition of a cyclic terminology in any case (see, e.g., (Nebel, 1990a; Baader, 1996b; Küsters, 1997)). Therefore, it is crucial to gain a more profound understanding of the intuition of these semantics, in order to make decision easier which semantics is to prefer in the representation task at hand. For this reason, automata theoretic characterizations have been provided for the three semantics with respect to general $\mathcal{FLN}$-terminologies. The characterizations of the three semantics themselves are easy extensions of the characterizations for $\mathcal{FL}_0$-terminologies (Baader, 1996b).

Using this automata theoretic descriptions, we have also proved characterizations of the important inference problems inconsistency and subsumption for $\mathcal{FLN}$ where non-trivial extensions in comparison with $\mathcal{FL}_0$-terminologies are necessary. Moreover, these descriptions have been used to derive decision algorithms and complexity results for inconsistency and subsumption.

Finally, we have shown that the important inference problems for $\mathcal{SLN}$- and $\mathcal{SL}_{dis}$-schemas can be reduced to corresponding problems for terminologies. This reduction provides us with automata theoretic decision algorithms for these problems, and it clarifies the expressive power of the considered schemas and terminologies; in addition, it extends the results for $\mathcal{SL}_{dis}$-schemas (Buchheit et al., 1994) to $\mathcal{SLN}$-schemas, which allow for arbitrary number-restrictions.

Future work will extend the characterizations presented here in order to derive decision algorithms for consistency and instance w.r.t. knowledge bases consisting of cyclic terminologies and *A-Boxes*. Furthermore, it has turned out that the automata-theoretic characterizations can be used to handle inference prob-

lems like computing the least common subsumer of concepts (LCS) and the most specific concept of individuals (MSC). Cyclic terminologies are necessary in this context since w.r.t. A-Boxes with role cycles the MSC of an individual may not be describable in terms of acyclic terminologies.

Acknowledgment

I am very grateful to Franz Baader for his helpful comments and suggestions as well as Ralf Molitor for valuable discussions.

References

Aït-Kaci, H. (1984). *A Lattice-Theoretic Approach to Computations Based on a Calculus of Partially Ordered Type Structures*. PhD thesis, University of Pennsylvania, PA.

Baader, F. (1990). Terminological cycles in KL-ONE-based knowledge representation languages. In *Proceedings of AAAI-90*, pages 621–626.

Baader, F. (1996a). A formal definition for the expressive power of terminological knowledge representation languages. *J. of Logic and Computation*, 6(1):33–54.

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

Bergamaschi, S. and Sartori, C. (1992). On taxonomic reasoning in conceptual design. *ACM Trans. on Database Systems*, 17(3):385–442.

Borgida, A. (1995). Description logics in data management. *IEEE Trans. on Knowledge and Data Engineering*, 7(5):671–682.

Buchheit, M., Donini, F., Nutt, W., and Schaerf, A. (1994). Refining the structure of terminological systems: Terminology = Schema + View. In *Proceedings of the 12th National Conference of the American Association for Artificial Intelligence, AAAI-94*, Seattle.

Buchheit, M., Donini, F., Nutt, W., and Schaerf, A. (1997). A refined architecture for terminological systems: Terminology = Schema + View. *Artificial Intelligence*. To appear.

Calvanese, D. (1996). Reasoning with inclusion axioms in description logics: Algorithms and Complexity. In Wahlster, W., editor, *Proceedings of the 12th European Conference on Artificial Intelligence, ECAI-96*. John Wiley & Sons.

Calvanese, D., Lenzerini, M., and Nardi, D. (1994). A unified framework for class based representation formalisms. In *Proceedings of the Fourth International Conference on the Principles of Knowledge Representation and Reasoning (KR-94)*, pages 109–120. Morgan Kaufmann, Los Altos.

Giacomo, G. D. and Lenzerini, M. (1994). 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.

Giunchiglia, F. and Sebastiani, R. (1996). A sat-based decision procedure for $\mathcal{ALC}$. In *5th International Conference on Principles of Knowledge Representation and Reasoning - KR'96*. Morgan Kaufmann, Los Altos.

Hollunder, B. and Nutt, W. (1990). Subsumption algorithms for concept languages. Technical Report RR-90-04, German Research Center for Artificial Intelligence, Germany.

Küsters, R. (1997). Characterizing the semantics of terminological cycles in $\mathcal{ALN}$ using finite automata. Technical Report LTCS-97-04, RWTH Aachen, Germany. ftp://www-lti.informatik.rwth-aachen.de/pub/papers/1997/Kuesters-LTCS-97-04.ps.gz.

MacGregor, R. (1992). What's needed to make a description logic a good KR citizen. In *Working Notes of the AAAI Fall Symposium on Issues on Description Logics: Users meet Developers*, pages 53–55.

Nebel, B. (1987). On terminological cycles. In *KIT Report 58*, Technische Universität Berlin.

Nebel, B. (1990a). Reasoning and revision in hybrid representation systems. In *Lecture Notes in Computer Science 422*. Springer Verlag, Berlin.

Nebel, B. (1990b). Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43:235–249.

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

Schild, K. (1994). 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. Morgan Kaufmann.