Published in John F. Sowa (ed.), *Principles of Semantic Networks*, p. 331–362. Morgan Kaufmann, San Mateo, CA, 1991.

Terminological Cycles: Semantics and Computational Properties

Bernhard Nebel*
German Research Center for Artificial Intelligence
Saarbrücken, West Germany

Abstract

Terminological knowledge representation formalisms are intended to capture the analytic relationships between terms of a vocabulary intended to describe a domain. A term whose definition refers, either directly or indirectly, to the term itself presents a problem for most terminological representation systems because it is obvious neither whether such a term is meaningful, nor how it could be handled by a knowledge representation system in a satisfying manner. After some examples of intuitively sound terminological cycles are given, different formal semantics are investigated and evaluated with respect to the examples. As it turns out, none of the different styles of semantics seems to be completely satisfying for all purposes. Finally, consequences in terms of computational complexity and decidability are discussed.

1 Introduction

When trying to represent an expert's knowledge about a sufficiently complex domain we have to account for the vocabulary used in this domain [Brachman and Levesque, 1982; Swartout and Neches, 1986]. This is exactly the purpose of *terminological knowledge representation formalisms*, which have their roots in *structural inheritance networks* [Brachman, 1979]. The main building blocks of such representation formalisms are *concepts* and *roles* [Brachman and Schmolze, 1985], similar to generic frames and slots in frame systems and to type nodes and links in semantic networks. In contrast to frame systems

*Address for correspondence: DFKI GmbH, Stuhlsatzenhausweg 3, D-6600 Saarbrücken 11, West Germany; e-mail: nebel@dfki.uni-sb.de.

and semantic networks, however, it is possible to *define* concepts by specifying *necessary and sufficient* conditions, while in semantic networks and frame systems only *necessary* conditions can be specified.

The most important reasoning task in such a context is the determination of *subsumption* between concepts, i.e., whether all *instances* of a concept are necessarily instances of the other concept. This kind of reasoning can be employed to support such diverse applications as information retrieval [Patel-Schneider *et al.*, 1984], explainable expert systems [Neches *et al.*, 1985], natural language processing [Webber and Bobrow, 1980; Sondheimer and Nebel, 1986], and computer configuration [Owsnicki-Klewe, 1988].

Based on these ideas, a number of system were built, e.g. KANDOR [Patel-Schneider, 1984], KL-TWO [Vilain, 1985; Schmolze, 1989], KRYPTON [Brachman *et al.*, 1985], MESON [Edelmann and Owsnicki, 1986], BACK [von Luck *et al.*, 1987; Nebel and von Luck, 1988], LOOM [MacGregor, 1988], CLASSIC [Brachman *et al.*, 1989; Borgida *et al.*, 1989], and SB-ONE [Kobsa, 1989], and the formal properties of these systems were investigated [Schmolze and Israel, 1983; Brachman and Levesque, 1984; Patel-Schneider, 1986; Levesque and Brachman, 1987; Nebel, 1988; Schild, 1988; Patel-Schneider, 1989a; Patel-Schneider, 1989b; Schmidt-Schauß, 1989; Schmidt-Schauß and Smolka, 1990; Nebel and Smolka, 1990; Nebel, 1990; Donini *et al.*, 1990; Hollunder *et al.*, 1990].

When studying the above mentioned papers, one notes that *terminological cycles* are usually ignored or explicitly excluded. Terminological cycles arise when a concept is defined by referring directly or indirectly to itself (which amounts to a loop in the network depicting the terminological knowledge base) as in the (informal) definition of the concept Human below:

> a Human is defined as
> a Mammal with
> exactly 2 parents and
> all parents are Humans

Such a definition obviously violates the plausible idea that the meaning of a concept "can be completely understood in terms of the meaning of its parts and the way these are composed" [Schmolze and Brachman, 1982, p. 11]. In trying to understand the meaning of Human, we inevitably end up trying to figure out what the meaning of Human could be. Additionally, the *subsumption algorithms* usually employed (see e.g. [Schmolze and Israel, 1983]) would end up in an infinite loop on such definitions. For these reasons terminological cycles have been excluded in theoretical investigations and practical terminological knowledge representation systems.

This exclusion would be justified if terminological cycles were not useful in this style of knowledge representation. However, experience with terminological knowledge representation systems in applications show that terminological cycles are used regularly [Kaczmarek *et al.*, 1986, p. 982]. Also, envisioning a system that views a terminological knowledge base as an abstract entity that can be changed incrementally (as described in [Nebel, 1989]), terminological cycles can be easily created and either have to be detected and rejected by the system—which makes the system specification overly complex and hard to understand by a user—or the system has to accept them as legal constructions. In addition, a decision to prohibit the use of terminological cycles should not be based on the fact that we do not understand the meaning or do not know the inference algorithms, but it should be based on an understanding of terminological cycles and justified by arguments concerning semantics and/or computational properties. For these reasons it seems worthwhile to analyze the semantic and algorithmic nature of terminological cycles.

The rest of the paper is organized as follows. A small and simple terminological formalism is formally introduced in Section 2. In Section 3, a brief description of possible kinds of terminological cycles is given, and the intuitive semantics of them are discussed. Based on that, Section 4 presents three different styles of semantics, namely, *descriptive semantics*, *least fixpoint semantics*, and *greatest fixpoint semantics* that are evaluated with respect to the examples. As it turns out, there is no obvious "winner." There are good arguments for the descriptive semantics and equally good arguments for the greatest fixpoint semantics. In fact, which one to choose seems to be a matter of the intended purpose. In Section 5, algorithmic consequences are discussed using results presented in [Nebel, 1990] and [Baader, 1990]. Finally, we will show that depending on the expressiveness of the underlying terminological formalism, terminological cycles can lead to severe computational problems, namely, to undecidability of subsumption.

2 A Framework for Representing Terminological Knowledge

In order to have something to build on, we need a concrete terminological knowledge representation formalism. In this paper, a small terminological formalism—called $\mathcal{TLN}$[1]—which is a subformalism of almost all the formalisms used in the systems quoted above,[2] will be used to investigate the

[1] It is the formalism $\mathcal{TL}$ introduced in [Nebel, 1990] extended by number restrictions.

[2] The only exception is KRYPTON.

nature of terminological cycles.

The basic building blocks of our formalism are a set $\mathbf{R}$ of *atomic roles* (denoted by R) and a set $\mathbf{A}$ of *atomic concepts* (denoted by A and B). We will assume that there are always two predefined concepts (which are also elements of $\mathbf{A}$), namely, $\top$ intended to denote everything, and $\bot$ which denotes nothing. Using these *atomic terms*, the set $\mathbf{D}$ of *concept descriptions* (denoted by C and D) is defined by the following abstract syntax rule:

$$\begin{array}{rcll} C, D & \rightarrow & A & \textit{atomic concept} \\ & | & C \sqcap D & \textit{concept conjunction} \\ & | & \forall R{:}\, C & \textit{value restriction} \\ & | & \exists^{\geq n} R & \textit{minimum restriction} \\ & | & \exists^{\leq n} R & \textit{maximum restriction.} \end{array}$$

Intuitively, a concept description is intended to denote all *objects* that fulfill the description. For instance, the concept description

$$\mathsf{Human} \sqcap \mathsf{Female} \sqcap \exists^{\geq 1}\mathsf{child} \sqcap \forall \mathsf{child}{:}\,(\mathsf{Human} \sqcap \mathsf{Female})$$

denotes the set of all Female Humans that have at least one child and whose children are all Female Humans, i.e., this expression denotes all mothers that have only daughters.

The formal meaning is given by a model-theoretic interpretation $\mathcal{I} = \langle \mathcal{D}, [\![\cdot]\!]^{\mathcal{I}} \rangle$, where $\mathcal{D}$ is an arbitrary set, the *domain*, and $[\![\cdot]\!]^{\mathcal{I}}$ is a function, the *interpretation function*, that maps atomic concepts to subsets of $\mathcal{D}$ and atomic roles to total functions from $\mathcal{D}$ to $2^{\mathcal{D}}$ [Brachman and Levesque, 1984]. The predefined concepts $\top$ and $\bot$ have the fixed interpretation $\mathcal{D}$ and $\emptyset$, respectively. The set $[\![R]\!]^{\mathcal{I}}(d)$ will be called *role-filler set* of role R for object d. The *denotation of concept descriptions* is defined inductively by[3]

$$\begin{array}{rcl} [\![C \sqcap D]\!]^{\mathcal{I}} & = & [\![C]\!]^{\mathcal{I}} \cap [\![D]\!]^{\mathcal{I}} \\ [\![\forall R{:}\, C]\!]^{\mathcal{I}} & = & \{d \in \mathcal{D} | \; [\![R]\!]^{\mathcal{I}}(d) \subseteq [\![C]\!]^{\mathcal{I}}\} \\ [\![\exists^{\geq n} R]\!]^{\mathcal{I}} & = & \{d \in \mathcal{D} | \; \|[\![R]\!]^{\mathcal{I}}(d)\| \geq n\} \\ [\![\exists^{\leq n} R]\!]^{\mathcal{I}} & = & \{d \in \mathcal{D} | \; \|[\![R]\!]^{\mathcal{I}}(d)\| \leq n\}. \end{array}$$

Since we are not only interested in forming a variety of concept descriptions, but also in defining new concepts, the notion of a *terminology* will be used, which allows us to assign the meaning of concept descriptions to atomic concepts. Formally, a *terminology* T is a total function $T{:}\, \mathbf{A} \rightarrow \mathbf{D}$, where $T(A)$ is the concept description defining the meaning of A or, if A is

[3]The expression $\|S\|$ denotes the cardinality of a set.

primitive in the terminology, $T(A) = A$.[4] In the following, we will use $\mathbf{A_p}$ for the set of atomic concepts that are primitive in a terminology (which are sometimes denoted by P and Q) and $\mathbf{A_n}$ for the set of nonprimitive atomic concepts. For $\top$ and $\bot$, we assume $\top, \bot \notin \mathbf{A_n} \cup \mathbf{A_p}$. Furthermore, we will use the expression $|C|$ when referring to the *size of the concept description* C, which is defined as the number of operators and atomic terms appearing in C, and we will use $|T|$ to denote the *size of a terminology*, which is defined as $\sum_{A \in \mathbf{A_n}} |T(A)|$.

The intended meaning of a terminology is the restriction of all possible interpretations to those that have identical denotations for atomic concepts and their defining concept descriptions. Formally, an interpretation $\mathcal{I}$ is a *model of T* iff

$$[\![A]\!]^{\mathcal{I}} = [\![T(A)]\!]^{\mathcal{I}} \ \ \textit{for all } A \in \mathbf{A}.$$

Now we can formalize the notion of *subsumption* between concepts, which has been informally introduced in Section 1. *C is subsumed by D in the terminology T*, written $C \preceq_T D$, under the following condition:

$$C \preceq_T D \quad \textit{iff} \quad [\![C]\!]^{\mathcal{I}} \subseteq [\![D]\!]^{\mathcal{I}} \ \ \textit{for all models } \mathcal{I} \textit{ of } T.$$

Giving an example, in the terminology

$$\begin{array}{rcll} T(\mathsf{Woman}) & = & \mathsf{Human} \sqcap \mathsf{Female} & (1) \\ T(\mathsf{Mother\text{-}of\text{-}daughters}) & = & \mathsf{Woman} \sqcap \exists^{\geq 1}\mathsf{child} \sqcap \forall \mathsf{child}\colon \mathsf{Woman} & (2) \\ T(\mathsf{Parent}) & = & \mathsf{Human} \sqcap \exists^{\geq 1}\mathsf{child} \sqcap \forall \mathsf{child}\colon \mathsf{Human}, & (3) \end{array}$$

Mother-of-daughters is subsumed by Parent because every Mother-of-daughters is a Parent in any model of the terminology.

Similarly to subsumption, *equivalence of concepts in a terminology*, written $C \approx_T D$, is defined by

$$C \approx_T D \quad \textit{iff} \quad [\![C]\!]^{\mathcal{I}} = [\![D]\!]^{\mathcal{I}} \ \ \textit{for all models } \mathcal{I} \textit{ of } T.$$

Finally, a concept C is called *incoherent in a terminology* T iff $C \approx_T \bot$.

Because of the set-theoretic semantics, incoherence and equivalence can be reduced to subsumption in $O(n)$ time and subsumption can be reduced to equivalence in $O(n)$ time, where n is the size of the concept descriptions.

[4] Note that this definition of *primitiveness* is the same as in KRYPTON [Brachman *et al.*, 1985] and different from the notion of a *primitive concept* in KL-ONE [Brachman and Schmolze, 1985]. However, this does not affect the formal expressiveness of the representation language [Nebel, 1989, Ch. 3].

Proposition 1 *Let T be a $\mathcal{TLN}$-terminology, and let C and D be concept descriptions. Then*

$$\begin{array}{lll} C \approx_T D & \textit{iff} & (C \preceq_T D \textit{ and } D \preceq_T C), \\ C \approx_T \perp & \textit{iff} & C \preceq_T \perp, \\ C \preceq_T D & \textit{iff} & C \approx_T C \sqcap D. \end{array}$$

For the sake of simplicity, we will sometimes consider terminologies such that the second argument in each value restriction is an atomic concept. Such terminologies will be called *normalized terminologies.*[5] Since any terminology can be transformed into such a form (by introducing at most $|T|$ auxiliary nonprimitive atomic concepts) without changing the denotations of the original terms, this assumption does not affect generality.

Although everything defined so far makes perfect sense, there is usually an additional syntactic restriction enforced on the form of terminologies. As pointed out in Section 1, the common intuition about terminologies is that concept definitions are "well-founded," i.e., that there are no terminological cycles such as the one presented in the beginning. Formally, a *terminological cycle* can be defined as follows. An atomic concept A *directly uses* another atomic concept B iff the expression $T(A)$ contains B. An atomic concept A_0 *uses* A_n iff there is a chain $A_0, A_1, \ldots A_n$ such that A_i directly uses A_{i+1}, $0 \leq i \leq n-1$. Finally, it will be said that a terminology T contains a *terminological cycle* iff some atomic concept uses itself.

The advantage of the acyclicity restriction is that the meaning of a concept can be understood in terms of the meaning of the atomic terms used in the defining description and the way these are composed. This is mirrored on a model-theoretic level by the fact that models of a terminology can be constructed inductively from *initial partial interpretations* that assign denotations to primitive concepts and roles only. Such initial partial interpretations will be denoted by $\check{\mathcal{I}}$.

Proposition 2 *Given a terminology T without terminological cycles, any initial partial interpretation $\check{\mathcal{I}}$ can be uniquely extended to a model of T.*

Proof Sketch: By induction on the depth of a terminology (see [Nebel, 1990, Lemma 1]). ■

From an algorithmic point of view, this means that subsumption determination in an acyclic terminology can be reduced to subsumption determination over concept descriptions, i.e., assuming that all atomic concepts are primitive (see [Nebel, 1990, Theorem 1]). This is done by expanding all nonprimitive concepts in an expression until it contains only primitive atomic concepts—which cannot be done if the terminology contains cycles.

[5] Note that all the example terminologies in this paper are normalized.

3 In Defense of Terminological Cycles

Basically, there are two kinds of terminological cycles—one which is obviously meaningful and another one which does not seem to make sense. An example of the latter kind of terminological cycles is the following terminology introducing the concepts Male-Human and Man:

$$T(\mathsf{Man}) = \mathsf{Human} \sqcap \mathsf{Male\text{-}Human} \quad (4)$$
$$T(\mathsf{Male\text{-}Human}) = \mathsf{Human} \sqcap \mathsf{Man}. \quad (5)$$

These definitions suggest that Man is a specialization of Male-Human and *vice versa*, which seems to be rather weird and violates the idea that all concepts in a terminology can be ordered hierarchically.

In general, such cycles will be called *component cycles* and the concepts involved are called *component-circular concepts.* Formally, an atomic concept A_0 *uses* A_n *as a component* iff there is a chain of atomic concepts $A_0, A_1, \ldots A_n$, $n > 0$, such that each A_i directly uses A_{i+1}, and A_{i+1} appears outside of the scope of any $\forall$ expression in $T(A_i)$, for $0 \leq i \leq n-1$. A concept A is *component-circular* iff A uses itself as a component.

We might simply prohibit the use of such cycles. However, if we view a terminological knowledge base as an abstract object on which some modification operations can be carried out as sketched in Section 1, we have to take special care to detect and reject operations intended to introduce cycles. This makes the specification of such a system complicated and clumsy. Therefore, if the semantics of the representation language could give us a sensible answer as to what such "definitions" could possibly mean, this would be much more elegant.

Besides the meaningless kind of cycles, there are cycles which are obviously meaningful and which often appear when modeling a domain. For instance, the description of recursive structures, e.g. binary trees, requires that we can use terminological cycles:

$$T(\mathsf{Binary\text{-}tree}) = \mathsf{Tree} \sqcap \exists^{\leq 2}\mathsf{branch} \sqcap \forall\mathsf{branch}\colon \mathsf{Binary\text{-}tree}. \quad (6)$$

The intuition behind this terminology is obviously that the concept Binary-Trees should describe tree-structured objects of degree two. For instance, consider the following initial partial interpretation, which is depicted in Figure 1:

$$\mathcal{D} = \{a, b, c, d, e, f, g, h\}$$
$$[\![\mathsf{Tree}]\!]^{\tilde{\mathcal{I}}} = \mathcal{D}$$

$$[\![\mathsf{branch}]\!]^{\breve{\mathcal{I}}} = \begin{cases} a & \mapsto & \{b, c\} \\ b & \mapsto & \emptyset \\ c & \mapsto & \{d\} \\ d & \mapsto & \emptyset \\ e & \mapsto & \{f, g\} \\ f & \mapsto & \{h\} \\ g & \mapsto & \{h\}. \end{cases}$$

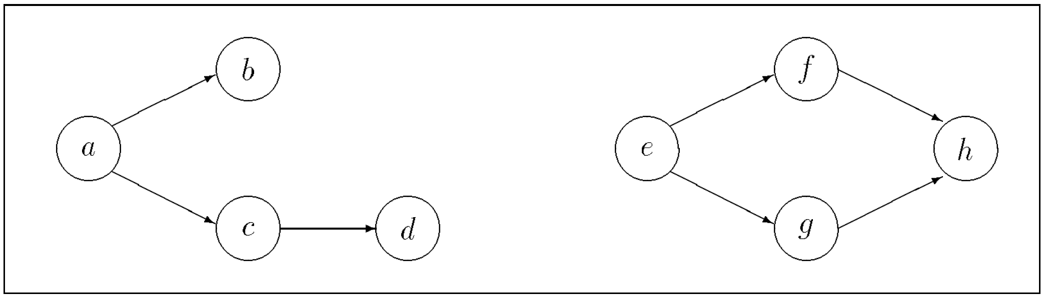

Figure 1: Some Object Structures Intended by the Definition of Binary-tree (Arrows denote branch relationships, circles denote elements of the denotation of Tree)

Extending $\breve{\mathcal{I}}$ to a model $\mathcal{I}$ of the terminology given by (6), one notes that $\{a, b, c, d\}$ is a subset of the denotation of Binary-tree, as expected. Furthermore, also $\{e, f, g, h\}$ must be a subset of $[\![\mathsf{Binary\text{-}tree}]\!]^{\mathcal{I}}$, i.e., the terminology permits also object structures which are directed acyclic graphs.

Concepts such as Binary-tree will be called *restriction-circular concepts.* Formally, an atomic concept is *restriction-circular* iff it uses itself and it is not *component circular*. Furthermore, assuming a normalized terminology T, it will be said that an atomic concept B can be *directly reached by a role* R from A iff B appears in a value restriction of $T(A)$. The atomic concept B *can be reached by a role-chain* $W = R_1R_2 \ldots R_m$ *from* A iff

1. $m = 1$ and B can be directly reached by R_1 from A, or

2. A directly uses a concept A' outside of any value restriction and B can be reached by W from A', or

3. a concept A' can be reached by R_1 and B can be reached by $R_2 \ldots R_m$ from A'.

Finally, A is said to be *restriction-circular over the role-chain* W iff A can be reached by W from A. For instance, Binary-tree is restriction-circular over the chain consisting of the one role "branch". Note that if there is one such role-chain W, then there there are infinitely many such role-chains, e.g. W^n, $n > 0$.

Another example of a restriction-circular concept, which at first sight seems to be similar, is Human as informally defined in Section 1:

$$T(\mathsf{Human}) \quad = \quad \mathsf{Mammal} \sqcap \exists^{\geq 2}\mathsf{parent} \sqcap \exists^{\leq 2}\mathsf{parent} \sqcap \forall \mathsf{parent}\colon \mathsf{Human}. \quad (7)$$

However, in this case, we are not aiming at describing finite structures but infinite ones as in Figure 2, which depicts a finite subset of an infinite interpretation where the circles denote elements of the denotation of Human and Mammal, solid arrows denote parent relationships, and the dashed arrows indicate that the tree extends infinitely to the right.

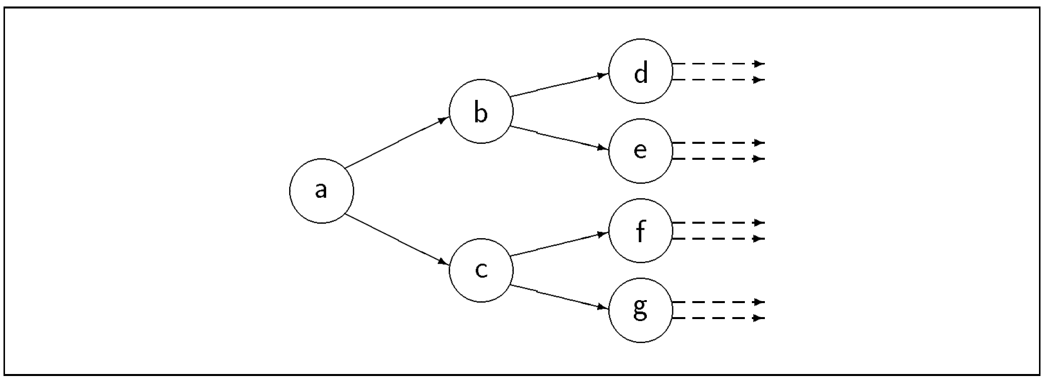

Figure 2: Object Structures Intended by the Definition of Human
(Arrows indicate parent relationships and circles denote elements of the denotation of Mammal and Human)

This kind of concept definition might raise the question of the origin of human beings. Because of space limitations, however, we will not discuss this subject further. From a formal of view, one notes that if all domain elements are in the denotation of Mammal and Human, then this interpretation is a model of the terminology (7). However, there is another model with the same initial partial interpretation such that $[\![\mathsf{Human}]\!]^{\mathcal{I}}$ is empty.

For a common sense view of the world, at least, the definition seems reasonable. I even believe that the conditions on Human are necessary and sufficient! This sounds a little strange at first but can be defended by the argument that an entity can be recognized as a Human when the entity has two $\mathsf{parents}$ which are known to be Humans.

A third kind of terminological cycles stresses the idea that it may be impossible to define a concept by referring to already defined terms, but possible to define two concepts by referring each to the other. In other words, we are aiming at describing circular object structures:

$$T(\mathsf{Car}) = \mathsf{Vehicle} \sqcap \forall \mathsf{engine\text{-}part}: \mathsf{Car\text{-}engine} \sqcap \exists^{\geq 1}\mathsf{engine\text{-}part} \quad (8)$$
$$T(\mathsf{Car\text{-}engine}) = \mathsf{Engine} \sqcap \forall \mathsf{is\text{-}engine\text{-}of}: \mathsf{Car} \sqcap \exists^{\geq 1}\mathsf{is\text{-}engine\text{-}of}. \quad (9)$$

Some conceivable object structures are depicted in Figure 3, where simple circles denote Car objects, double circles denote Car-engine objects, simple arrows denote engine-part relationships, and double arrows denote is-car-engine-of relationships. Note, however, that object structures are possible which do not follow this pattern, e.g. infinite chains of Car objects and Car-engine objects connected by the appropriate relationships.[6]

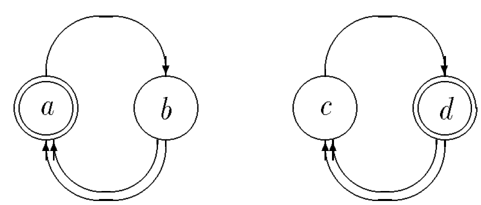

Figure 3: Object structures intended by the definition of Car-engine and Car (Simple circles denote Car objects, double circles denote Car-engine objects, simple arrows denote engine-part relationships, and double arrows denote is-car-engine-of relationships.)

Such concept definitions are obviously not "well-founded," and I know that if I would really insist that this example is reasonable and has to be part of any knowledge engineer's basic skills in modeling terminological knowledge, then I probably would loose some credibility. Therefore, I will defend this kind of terminological modeling only with a pragmatic argument.

Terminological cycles such as the one in (8)–(9) can be exploited in *hybrid representation systems*, which consists of an *assertional* and a *terminological component* [Brachman and Levesque, 1982]. For instance, if it is known that the object a is a Car, and a role filler of the engine-part role for object a is the object b, then we can conclude that object b must be a Car-engine. Obviously, this game also works the other way around. Thus, this kind of cycle permits a special and interesting mode of hybrid reasoning.

Depending on the expressiveness of the terminological formalism, there can be similar kinds of terminological cycles involving roles. Furthermore,

[6]Inverse roles are necessary to avoid such structures.

we may have cycles such that the meaning of a role depends on a concept which in turn uses the role in its definition. We will ignore these cases here, however.

4 What's in a Terminological Cycle?

We have seen in Section 2 that the meaning of defined concepts can be completely derived from the meaning of primitive concepts, roles, and the various concept-forming operators (see Prop. 2) provided the terminology is acyclic. If we continue to use the semantics specified in Section 2 (which will be called *descriptive semantics* in the following) in the presence of terminological cycles, we loose this nice property.

Using the Binary-tree example (see (6)), for instance, we note that for some initial partial interpretations, there is more than one extension of the initial interpretation to a model of T. Consider the following $\breve{\mathcal{I}}$:

$$
\begin{array}{rcl}
\mathcal{D} & = & \{a, b, c\} \\
[\![\mathsf{Tree}]\!]^{\breve{\mathcal{I}}} & = & \{a, b, c\} \\
[\![\mathsf{branch}]\!]^{\breve{\mathcal{I}}} & = & \left\{ \begin{array}{lcl} a & \mapsto & \emptyset \\ b & \mapsto & \{c\} \\ c & \mapsto & \{b\}. \end{array} \right.
\end{array}
$$

Now, there are two possible models, $\mathcal{I}$ and $\mathcal{J}$, extending $\breve{\mathcal{I}}$:

$$
\begin{array}{rcl}
[\![\mathsf{Binary\text{-}tree}]\!]^{\mathcal{I}} & = & \{a\} \\
[\![\mathsf{Binary\text{-}tree}]\!]^{\mathcal{J}} & = & \{a, b, c\}.
\end{array}
$$

Since this seems to violate the ideas spelled out in Section 1, one idea could be to use only one particular model extending a given initial partial interpretation, which is in some sense "canonical"—provided such a model is always identifiable. Whether one uses this approach or the descriptive semantics is a matter of the models one considers as plausible and the subsumption relationships one wants to have.

4.1 Fixpoint Models

In order to explore these ideas, we will characterize models as fixpoints of a certain operator on interpretations. The set of interpretations (over a given terminology T with fixed $\mathbf{A}$ and $\mathbf{R}$) that have identical interpretations of roles and primitive concepts, i.e. all interpretations with the same initial partial part $\breve{\mathcal{I}}$, will be denoted by $\Psi_{\breve{\mathcal{I}}}$. Furthermore, Γ shall be a function mapping interpretations to interpretations (for a given terminology T) as follows:

$$\begin{aligned} \Gamma\colon \Psi_{\check{\mathcal{I}}} &\rightarrow \Psi_{\check{\mathcal{I}}} \\ \Gamma\colon \mathcal{I} &\mapsto \mathcal{J} \end{aligned}$$

such that

$$\begin{aligned} [\![A]\!]^{\mathcal{J}} &= [\![T(A)]\!]^{\mathcal{I}} \\ [\![R]\!]^{\mathcal{J}} &= [\![R]\!]^{\mathcal{I}}. \end{aligned}$$

A fixed point of Γ, i.e., an interpretation $\mathcal{I}$ with the property $\Gamma(\mathcal{I}) = \mathcal{I}$, is clearly a model of the terminology T according to the definition in Section 2, which will be called *admissible model*.

Least or greatest fixpoint models (*lfp*- and *gfp*-models for short) would fulfil the requirement of being "canonical" in the sense mentioned above. In order to define *lfp*- and *gfp*-models, however, we need an ordering on $\Psi_{\check{\mathcal{I}}}$. A straightforward and intuitively plausible ordering is the component-wise set-inclusion relation over the denotation of nonprimitive atomic concepts, written $\sqsubseteq$:

$$\mathcal{I} \sqsubseteq \mathcal{J} \quad \textit{iff} \quad [\![A]\!]^{\mathcal{I}} \subseteq [\![A]\!]^{\mathcal{J}} \textit{ for all } A \in \mathbf{A_n}, \mathcal{I}, \mathcal{J} \in \Psi_{\check{\mathcal{I}}}.$$

Obviously, $(\Psi_{\check{\mathcal{I}}}, \sqsubseteq)$ is a partial ordering which forms together with component-wise union as the least upper bound ($\bigsqcup$) a complete lattice. Thus, it seems reasonable that it is possible to apply Tarski's [1955] fixpoint theorem, which says that for a complete lattice L (with $\perp_L$ as the least element) and for any *monotone* function $f\colon L \rightarrow L$ (i.e., for all $x, y \in L : f(x) \sqsubseteq f(y)$ if $x \sqsubseteq y$)

1. the set of fixed points of f is nonempty and forms a complete lattice,
2. if f is *continuous* (i.e., for any totally ordered set $X \subseteq L : f(\bigsqcup X) = \bigsqcup f(X)$), then the least fixed point of f is equal to $\bigsqcup_{n=0}^{\infty} f^n(\perp_L)$.

As a matter of fact, the basic condition of the theorem can be easily verified.

Proposition 3 *Given a terminology T and an initial partial interpretation $\check{\mathcal{I}}$, Γ is monotone on $\Psi_{\check{\mathcal{I}}}$.*

Proof: By structural induction on the definition of the denotation of concept descriptions. ■

Furthermore, we get the intuitive result that the set of *lfp*- and *gfp*-models is identical with the set of admissible models when the terminology is acyclic.

Proposition 4 *Let T be a terminology without terminological cycles. Then $\mathcal{I}$ is an admissible model iff it is a lfp-model and a gfp-model.*

Proof: Immediate by Proposition 2 and the fact that all admissible models are fixed points of Γ. ■

Similarly to subsumption w.r.t. descriptive semantics as defined in Section 2, we define *lfp*-subsumption, written $\preceq_T^{lfp}$, and *gfp*-subsumption, written $\preceq_T^{gfp}$, by

$$\begin{array}{lll} C \preceq_T^{lfp} D & \textit{iff} & [\![C]\!]^{\mathcal{I}} \subseteq [\![D]\!]^{\mathcal{I}} \;\; \textit{for all lfp-models } \mathcal{I} \textit{ of } T, \\ C \preceq_T^{gfp} D & \textit{iff} & [\![C]\!]^{\mathcal{I}} \subseteq [\![D]\!]^{\mathcal{I}} \;\; \textit{for all gfp-models } \mathcal{I} \textit{ of } T. \end{array}$$

For concept equivalence w.r.t to *lfp*- and *gfp*-semantics we will use a similar notation.

Although we know by monotonicity of Γ on $\Psi_{\check{\mathcal{I}}}$ and Tarski's fixpoint theorem that the least and greatest fixpoint exist, it is not possible to generate the least fixed point by an ordinary fixpoint iteration. The reason is that Γ is not a continuous function because the denotation of $(\forall R\colon C)$ can depend on infinitely many elements in $\mathcal{D}$.[7] Thus, we could either use transfinite fixpoint iteration (see [Lloyd, 1984, p. 29]) or we could restrict our attention to models with finite role-filler sets, which will be called *roleset finite*. Indeed, this restriction does not change subsumption relationships, and, furthermore, such models seem to be more plausible.

Lemma 1 $C \preceq_T^{lfp} D$ *iff* $[\![C]\!]^{\mathcal{I}} \subseteq [\![D]\!]^{\mathcal{I}}$ *for all roleset-finite lfp-models $\mathcal{I}$.*

Proof: The "only if" direction is obvious. For the "if" direction let $\mathcal{I}$ be any admissible model of a normalized terminology T. Assume that for some $d \in \mathcal{D}$ the set $[\![R]\!]^{\mathcal{I}}(d)$ is infinite. Then there exists another model $\mathcal{J}$ of T which is identical with $\mathcal{I}$ except that $[\![R]\!]^{\mathcal{J}}(d)$ is a finite subset of $[\![R]\!]^{\mathcal{I}}(d)$ (which satisfies or violates all value, minimum and maximum restrictions for all defined concepts and all subexpressions of C and D which were originally satisfied or violated, for the element d).

Assume that $\mathcal{I}$ is a *lfp*-model, i.e., the smallest model extending the initial, partial interpretation $\check{\mathcal{I}}$. Now, we will show that $\mathcal{J}$ is the smallest model extending the initial partial interpretation $\check{\mathcal{J}}$. Assuming otherwise, i.e., that there is a model $\mathcal{J}'$ that extends $\check{\mathcal{J}}$ and that is smaller than $\mathcal{J}$, leads to the conclusion that for some $e \in \mathcal{D}$ and some concept C: $e \notin [\![C]\!]^{\mathcal{J}'}$ but $e \in [\![C]\!]^{\mathcal{J}}$. Since $\check{\mathcal{J}}$ and $\check{\mathcal{I}}$ differ only in role-filler sets of d, d must be such an element. However, since we preserved all value, minimum, and maximum restrictions,

[7] An example of noncontinuity can be found in [Baader, 1990].

it follows that there is a model $\mathcal{I}'$ which extends $\check{\mathcal{I}}$ such that $d \notin [\![C]\!]^{\mathcal{I}'}$, i.e., $\mathcal{I}'$ is smaller than $\mathcal{I}$—a contradiction of the assumption. Thus, if $\mathcal{I}$ was a *lfp*-model, then $\mathcal{J}$ is a *lfp*-model, as well.

Since d was chosen arbitrarily, the arguments above apply to the entire domain. Hence, if there is an arbitrary *lfp*-model $\mathcal{I}$, then there exists a roleset-finite *lfp*-model $\mathcal{J}$ with identical denotations for all $A \in \mathbf{A}$ and for given C and D. Thus, subsumption is identical. ■

For this set of roleset-finite interpretations, denoted by $\Psi^f_{\check{\mathcal{I}}}$, the desired property of Γ can be easily proven.

Proposition 5 *Given a terminology T and an initial partial interpretation $\check{\mathcal{I}}$, Γ is continuous on $\Psi^f_{\check{\mathcal{I}}}$.*

Proof: Let Φ a totally ordered subset of $\Psi^f_{\check{\mathcal{I}}}$. By monotonicity we know $\bigsqcup \Gamma(\Phi) \sqsubseteq \Gamma(\bigsqcup \Phi)$. For the other direction let $\mathcal{I} = \Gamma(\bigsqcup \Phi)$, $\mathcal{J} = \bigsqcup \Phi$, and assume $d \in [\![A]\!]^{\mathcal{I}}$. The reason for $d \in [\![A]\!]^{\mathcal{I}}$ is that there is a finite subset F of $\mathcal{D}$ such that for all $e \in F : e \in [\![B]\!]^{\mathcal{J}}$ for some atomic concepts $B \in \mathbf{A}$ and $e \in [\![R]\!]^{\mathcal{J}}(d)$ for some atomic roles $R \in \mathbf{R}$. Since Φ is totally ordered, there must be $\mathcal{J}' \in \Phi$ with the same property. Hence, $d \in [\![A]\!]^{\sqcup\Gamma(\Phi)}$, i.e., $\bigsqcup \Gamma(\Phi) \sqsupseteq \Gamma(\bigsqcup \Phi)$. ■

4.2 Least Fixpoint Semantics

When there is choice between different fixed points, the least one is usually the most "attractive" one—it is the fixed point which makes the least "commitments". For instance, in semantics of programming languages, we are usually interested in the least fixed point because in the space of functions the least fixed point corresponds to the partial function giving results for terminating computations and being undefined for nonterminating computations. In our case, an *lfp*-model amounts to something similar. It is the least model contained in all admissible models. Furthermore, for roleset-finite *lfp*-models, the semantics is highly constructive. For a given initial partial interpretation, we have $d \in [\![C]\!]^{\mathcal{I}}$ for the *lfp*-model $\mathcal{I} \in \Psi^f_{\check{\mathcal{I}}}$ iff there exists some finite approximation $\mathcal{J} = \bigsqcup_{n=0}^{i} \Gamma(\perp_{\Psi^f_{\check{\mathcal{I}}}})$ such that $d \in [\![C]\!]^{\mathcal{J}}$.

For these reasons, *lfp*-semantics might seem to be the most plausible one to choose. Before we do so, however, the *lfp*-semantics should be evaluated against the intuitions spelled out in Section 3.

The first kind of cycles, the component cycles, are treated in way which seems to be reasonable.

Proposition 6 *If A is a component-circular concept in a terminology T, then $A \approx^{lfp}_T \perp$.*

Proof: Let $\mathbf{C_c}$ be the maximal set of nonprimitive atomic concepts that use A as a component and are used by A as a component at the same time. All concepts in $\mathbf{C_c}$ are obviously component-circular (and it is the greatest such set containing A).

Now note that if all $B \in \mathbf{C_c}$ have an empty denotation in an interpretation $\mathcal{I}$, this holds for $\Gamma(\mathcal{I})$ as well. Thus, there exists no natural number i such that $\mathcal{J} = \bigsqcup \Gamma^i_{n=0}(\perp_{\Psi^f_{\mathcal{I}}})$ and $[\![B]\!]^{\mathcal{J}} \neq \emptyset$ for any $B \in \mathbf{C_c}$. Thus, for any *lfp*-model $\mathcal{I}$ of T: $[\![B]\!]^{\mathcal{I}} = [\![\perp]\!]^{\mathcal{I}}$ for all $B \in \mathbf{C_c}$. ■

This means we could eliminate these cycles by defining all component-circular concepts as $\perp$ without changing the meaning of the terminology.

Corollary 1 *If the terminology T contains component cycles, then there exists a terminology T' that does not contain component cycles and*

$$\mathcal{I} \text{ is a lfp-model of } T \quad \text{iff} \quad \mathcal{I} \text{ is a lfp-model of } T'.$$

It should be obvious that the other type of terminological cycles, the restriction cycle, is more complicated. In particular, we can describe object structures which cannot be described by acyclic terminologies, and, hence, such cycles cannot be eliminated from a terminology.

In order to describe the effect of these cycles, the denotation of role-chains (as introduced in Section 3) has to be defined. Let $W = R_1 R_2 \ldots R_m$. Then the denotation of W is defined as the functional composition $[\![W]\!]^{\mathcal{I}} = [\![R_m]\!]^{\mathcal{I}} \circ \ldots \circ [\![R_2]\!]^{\mathcal{I}} \circ [\![R_1]\!]^{\mathcal{I}}$. The expression $|W|$ will be used to denote the length of a role-chain.

Proposition 7 *Let A be a restriction-circular concept that is circular over the role-chains W_j in a terminology T. Then for all roleset-finite lfp-models $\mathcal{I}$ of T: If $d \in [\![A]\!]^{\mathcal{I}}$, then there exists a natural number $n > 0$ such that $[\![W_j]\!]^{\mathcal{I}}(d) = \emptyset$ for all role-chains with $|W_j| \geq n$.*

Proof: Assume that $[\![W_j]\!]^{\mathcal{I}}(d) \neq \emptyset$ for all role-chains regardless of their length. By induction over the construction of the least fixed point it follows that for all natural numbers m: for $\mathcal{J} = \bigsqcup_{i=0}^{m} \Gamma^i(\perp_{\Psi^f_{\mathcal{I}}})$, we have $d \notin [\![A]\!]^{\mathcal{J}}$, and, hence, $d \notin [\![A]\!]^{\mathcal{I}}$ for any *lfp*-model $\mathcal{I}$ of T. ■

Applying Prop. 7 to the Binary-tree example (see (6)), we see that we neither get circular binary trees nor trees with infinite depth, which matches nicely with the intuition. However, this also means that we do not get the structure we would have expected in the cases of Human (see (7)) and Car (see (8)–(9)). Even worse, the denotations of Human and Car are empty in all *lfp*-models, as can be easily deduced from the next corollary.

Corollary 2 *Let $A_0 A_1 \ldots A_n$, where $A_0 = A_n$, be a restriction cycle such that A_0 is circular over $P = R_1 \ldots R_m$, and let $A_{i_1} \ldots A_{i_m}$ be the concepts such that A_{i_j+1} is directly reached by R_j from A_{i_j}. If all expressions $T(A_{i_j})$ contain a minimum restriction on R_j, then $A_i \approx_T^{lfp} \perp$, for $0 \leq i \leq n$.*

We could take the observation that humans and cars do not exist as a deep truth (of which nobody was aware), or, taking a more pragmatic view, as an indication that *lfp*-semantics might be not the right choice.

4.3 Greatest Fixpoint Semantics

Using *gfp*-semantics obviously avoids the cruel consequences concerning the existence of the reader, the author, and the cars they possess. However, there are other shortcomings. First of all, it is not as constructive as *lfp*-semantics.[8] Second, it violates one intuition spelled out in Section 3. Elaborating on the Human example, we could define the concept Horse in the same way:

$$T(\mathsf{Human}) = \mathsf{Mammal} \sqcap \exists^{\geq 2}\mathsf{parent} \sqcap \exists^{\leq 2}\mathsf{parent} \sqcap \forall\mathsf{parent}{:}\, \mathsf{Human} \quad (10)$$
$$T(\mathsf{Horse}) = \mathsf{Mammal} \sqcap \exists^{\geq 2}\mathsf{parent} \sqcap \exists^{\leq 2}\mathsf{parent} \sqcap \forall\mathsf{parent}{:}\, \mathsf{Horse}. \quad (11)$$

As is easy to see, $[\![\mathsf{Human}]\!]^{\mathcal{I}} = [\![\mathsf{Horse}]\!]^{\mathcal{I}}$ for all *gfp*-models because assuming that the denotations are different leads to the conclusion that there is another fixpoint which is greater and has identical denotations for Human and Horse. That means we have to give up the intuition that (10) defines the concept Human. One should add a primitive atomic concept, say Humanness, to the definition of Human in order to distinguish Human from other Mammals with two parents. As a consequence it follows that under *gfp*-semantics the condition that ones parents are human beings is not sufficient for proving that one is a human being as well—a way of reasoning that could be nicely exploited when assertional knowledge is represented as well.

After having now an idea what *gfp*-semantics does to restriction-circular concepts, it seems worthwhile to analyze component-circular concepts.

Proposition 8 *Let A be a component-circular concept and let $\mathbf{C_c}$ be the largest set of concepts that use A as a component and are used by A as a component. Let $D(A)$ be a concept description identical to $T(A)$ except that all occurrences of concepts from $\mathbf{C_c}$ that do not appear in value restrictions are replaced by $\top$. Then for all $B \in \mathbf{C_c}$:*

$$B \approx_T^{gfp} \textstyle\bigsqcap_{A \in \mathbf{C_c}} D(A).$$

[8] In [Baader, 1990], however, it is shown that Γ is "downward continuous" on $\Psi_{\mathcal{I}}^{\mathcal{T}}$, which means that any element in the complement of a concept denotation is not contained in some finite approximation of the greatest fixed point denotation of this concept.

Proof: First of all, note that all concepts in $\mathbf{C_c}$ must have the same denotation. Second, since enlarging the denotation of one concept, enlarges the entire interpretation, all concepts in $\mathbf{C_c}$ must have the greatest possible denotation, which is just $\bigcap_{A \in \mathbf{C_c}} D(A)$. ∎

Although this result is less satisfying than Prop. 6, it is, of course, tolerable because we are not much interested in component-circular concepts. Furthermore, it shows that under *gfp*-semantics, such cycles can be easily eliminated, as well.

Corollary 3 *If the terminology T contains component cycles, then there exists a terminology T' that does not contain component cycles and*

$$\mathcal{I} \text{ is a gfp-model of } T \quad \text{iff} \quad \mathcal{I} \text{ is a gfp-model of } T'.$$

4.4 Descriptive Semantics

Finally, descriptive semantics should be briefly characterized. This style of semantics—which is similar to the ordinary semantics of first-order logic—does not lead to equivalence of Humans and Horses because we can think of infinite or circular object structures which satisfy the terminology (10)–(11) without making the denotations of Human and Horse identical. This means on one hand that we can indeed somebody recognize as a Human if and only if her two parents are Humans without being committed to conclude that she is a Horse, as well. On the other hand, the definition does not determine a unique interpretation for given initial, partial interpretations of primitive concepts and roles, which leads to the fact that for datatype-like concepts such as Binary-tree and Ternary-tree expected subsumption-relations are missed.

$$T(\mathsf{Binary\text{-}tree}) = \mathsf{Tree} \sqcap \exists^{\leq 2}\mathsf{branch} \sqcap \forall \mathsf{branch}\colon \mathsf{Binary\text{-}tree} \tag{12}$$
$$T(\mathsf{Ternary\text{-}tree}) = \mathsf{Tree} \sqcap \exists^{\leq 3}\mathsf{branch} \sqcap \forall \mathsf{branch}\colon \mathsf{Ternary\text{-}tree} \tag{13}$$

Although $\mathsf{Binary\text{-}tree} \preceq_T \mathsf{Ternary\text{-}tree}$ is something everybody would expect, descriptive semantics does not support this subsumption relationships because models may contain infinite and circular object structures. However, it should be noted that a hybrid reasoner would classify any (finite) tree-structured object that can be classified as Binary-tree as a Ternary-tree as well.

In some sense, descriptive semantics seems to assign more importance to *concept names* of circular concepts. Restriction-circular concepts are very similar to primitive concepts in that they can "choose" their denotation.

However, since the denotation of these concepts is not completely unconstrained, there are some very subtle relations between such concepts, which will be analyzed in the next section. For component-circular concepts the picture is clearer. We get again the result that such cycles are superfluous and that they can be eliminated without changing the meaning of the terminology.

Proposition 9 *Assume as in Prop. 8 a component-circular concept A, the largest set of component-circular concepts $\mathbf{C_c}$ that contains A, and $D(A)$ as identical to $T(A)$ except that all concepts occurring in $\mathbf{C_c}$ which appear outside of a value restrictions are replaced by $\top$. Then for all $B, B' \in \mathbf{C_c}$:*

1. *$B \approx_T B'$,*
2. *$B \preceq_T \bigsqcap_{A \in \mathbf{C_c}} D(A)$,*
3. *$C \preceq_T B$ iff $C \approx_T \bot$ or C uses B as a component.*

Proof: The first property follows immediately from the definition of a model in Section 2, and the second property is a direct consequence of Prop. 8. Furthermore, the "if" direction of the third property is obvious. For the "only if" direction assume that $C \preceq_T B$, but C does not use B and $C \not\approx_T \bot$. Let $\mathcal{I}$ be a model such that $[\![C]\!]^{\mathcal{I}} \neq \emptyset$ and $[\![C]\!]^{\mathcal{I}} \subseteq [\![B]\!]^{\mathcal{I}}$. Let $d \in [\![C]\!]^{\mathcal{I}}$. Now extend the model $\mathcal{I}$ to an interpretation $\mathcal{I}'$ with domain $\mathcal{D}' = \mathcal{D} \cup \{d'\}$ such that for all $e \in \mathcal{D}$: $e \in [\![R]\!]^{\mathcal{I}'}(d')$ iff $e \in [\![R]\!]^{\mathcal{I}}(d)$ and $d' \in [\![A]\!]^{\mathcal{I}'}$ iff $d \in [\![A]\!]^{\mathcal{I}}$, for all $R \in \mathbf{R}$ and all $A \in \mathbf{A}$. As can be easily verified, $\mathcal{I}'$ is a model again. Removing d' from $[\![B]\!]^{\mathcal{I}'}$ and from all denotations of concepts that use B as a component leads to another interpretation $\mathcal{J}$, which is again a model. $\mathcal{J}$ does not satisfy our assumption, however. ■

This means, a set of component circular concepts behaves as if a unique, fresh primitive concept is used in the definition of all of them. As a matter of fact, we can transform the terminology into such a form without changing "relevant parts" of the models. Let $\mathcal{I}|_X$ denote the restriction of an interpretation to a certain set X of concepts and roles.

Corollary 4 *If the terminology T over $\mathbf{A}$ and $\mathbf{R}$ contains component cycles, then there exists a terminology T' over $\mathbf{A}'$ and $\mathbf{R}$, where $\mathbf{A} \subseteq \mathbf{A}'$, such that for every admissible model $\mathcal{I}$ of T there exists an admissible model $\mathcal{I}'$ of T', and vice versa, with $\mathcal{I}|_{(\mathbf{A}\cup\mathbf{R})} = \mathcal{I}'|_{(\mathbf{A}\cup\mathbf{R})}$, and T' does not contain a component cycle.*

Proof: Let $\mathbf{C_c}$ and $D(A)$ be defined as in Prop. 9. Let $\mathsf{P} \notin \mathbf{A}$ and $\mathbf{A}' = \mathbf{A} \cup \{\mathsf{P}\}$. Let T' be a terminology defined as follows:

$$T'(A) \quad = \quad \begin{cases} \mathsf{P} & \textit{if } A = \mathsf{P}, \\ \mathsf{P} \sqcap \bigsqcap_{B \in \mathbf{C_c}} D(B) & \textit{if } A \in \mathbf{C_c}, \\ T(A) & \textit{otherwise.} \end{cases}$$

If $\mathcal{I}$ is a model of T, then we obtain a model $\mathcal{I}'$ of T' by setting $[\![X]\!]^{\mathcal{I}'} = [\![X]\!]^{\mathcal{I}}$ for all $X \in \mathbf{A} \cup \mathbf{R}$ and $[\![\mathsf{P}]\!]^{\mathcal{I}'} = [\![B]\!]^{\mathcal{I}}$, for some $B \in \mathbf{C_c}$.

Let $\mathcal{I}'$ be a model of T'. Restricting $\mathcal{I}'$ to $\mathbf{A} \cup \mathbf{R}$, we get an interpretation $\mathcal{I}$. To show that $\mathcal{I}$ is a model of T, we have to verify that all equations $[\![A]\!]^{\mathcal{I}} = [\![T(A)]\!]^{\mathcal{I}}$ are satisfied. This is trivially true for all concepts $A \notin \mathbf{C_c}$. For the concepts in $\mathbf{C_c}$, the definition can be written as $T(A) = D(A) \sqcap B$ for some $B \in \mathbf{C_c}$. Since $[\![A]\!]^{\mathcal{I}'} = [\![B]\!]^{\mathcal{I}'}$ and $[\![A]\!]^{\mathcal{I}'} \subseteq [\![D(A)]\!]^{\mathcal{I}'}$, we know $[\![A]\!]^{\mathcal{I}} = [\![T(A)]\!]^{\mathcal{I}}$ for all $A \in \mathbf{C_c}$.

Thus, removing component cycles iteratively, we can obtain a terminology T' with the desired property. ∎

4.5 Comparing the Different Styles of Semantics

First of all, the different styles of semantics shall be characterized in terms of the induced subsumption relation. It is obvious that subsumption w.r.t. descriptive semantics implies *lfp*- and *gfp*-subsumption because in the former case, more models, i.e., all fixpoints, have to be considered. Furthermore, since *lfp*-semantics tends to force denotations of circular concepts to the empty set, which leads to the fact that Human and Car are identical, we know that *lfp*-subsumption does not imply *gfp*-subsumption. The converse, however, seems to be plausible—but does not hold. Because of Prop. 7, we know that a restriction-circular concept cannot be *lfp*-equivalent to $\top$, but there are restriction-circular concepts that are *gfp*-equivalent to $\top$, for instance, $T(A) = \forall R\colon A$.

When evaluating the three styles of semantics against the intuitions spelled out in Section 3, it is obvious that *lfp*-semantics is a looser since it forces us to conclude that a number of examples which are intuitively plausible are in fact incoherent. There is no such clear judgement for the remaining two styles, however. Although, at first sight, *gfp*-semantics seems to be the more plausible one, there are a number of good arguments for the descriptive semantics, as well. Greatest fixpoint semantics has on the positive side that

- it supports subsumption relationships one would expect between "structurally similar concepts,"[9] such as the one between Binary-tree

[9]This somewhat vague notion will become more precise in the next section.

and Ternary-tree,

- extends an initial partial interpretation to a unique model similar to the acyclic case, i.e., it generalizes the idea of determining the meaning of a concept in terms of the meaning of its parts and the way these are composed.

On the negative side, we find that *gfp*-semantics does not permit a special mode of hybrid reasoning where we conclude the humanness of an object from the humanness of the parents—without some unacceptable consequences. A more serious argument against *gfp*-semantics is that it cannot be generalized to more powerful terminological languages. For instance, if roles can be defined in terms of concepts,[10] Γ is not monotone any longer. The reason is that increasing role denotations leads to smaller concept denotations in the general case.

When considering descriptive semantics, we can conclude that

- by not forcing structurally similar concepts to be equivalent, hybrid reasoning might be better supported,
- it is the conceptually most straightforward generalization of the standard semantics, and
- it can be applied to arbitrary terminological languages.

On the other hand, conditions for subsumption w.r.t. descriptive semantics are conceptually more complicated than *gfp*-semantics, as we will see below.

All in all, I believe there are no conclusive arguments yet. However, by having explored the space of reasonable alternatives, we know now what the implications are—to a certain extent.

5 Reasoning with Terminological Cycles

As mentioned in Section 1, there are two main reasons why terminological cycles are usually omitted. One is the unclear semantics, and the other one is the problem cycles create for subsumption algorithms—a problem we will tackle in this section. Based on results presented in [Nebel, 1990] and [Baader, 1990], it will be shown that subsumption in *general terminologies* that may contain cycles is more difficult than subsumption in acyclic ones. For this purpose, we will concentrate on a even smaller terminological language, called $\mathcal{TL}$, that does not contain minimum and maximum restrictions.

[10]The `restrict` operator of the language $\mathcal{FL}$ described in [Levesque and Brachman, 1987], for example, can be used for this purpose.

Although it seems to be straightforward to generalize the results obtained for $\mathcal{TLN}$, they cannot be generalized to arbitrarily powerful terminological languages, which is shown by giving an example of a terminological language for which subsumption is decidable in the acyclic case but undecidable when terminological cycles are permitted.

Usually subsumption algorithms are specified over concept descriptions only, assuming that all atomic concepts are primitive (see, for instance, [Levesque and Brachman, 1987; Patel-Schneider, 1989a; Schmidt-Schauß and Smolka, 1990; Hollunder *et al.*, 1990]). This is sufficient as long as the terminology is acyclic because in this case we can expand all nonprimitive concepts by their definitions until the concept descriptions contain only primitive atomic concepts.[11]

In [Nebel, 1990] it was shown that another perspective on subsumption determination in terminologies is possible when considering terminological languages containing only value restrictions and concept conjunctions, namely, to view acyclic terminologies as acyclic *nondeterministic finite state automata* (NDFA). Under this view it turns out that concept equivalence is reducible to automaton equivalence. Similarly, concept subsumption is reducible to inclusion of the languages accepted by the automata.

5.1 Viewing Terminologies as Automata

Restricting our attention to a terminological language containing only concept conjunction and value restriction, a normalized terminology T can be viewed as a set of NDFAs

$$\mathcal{A}_{\langle T,A,S\rangle} = (\mathbf{R}, \mathbf{A}, \delta, A, S),$$

where $A \in \mathbf{A}$, $S \subseteq \mathbf{A_p} \cup \{\bot\}$, and

1. $\mathbf{R}$ is the set of *input symbols*,
2. $\mathbf{A}$ is the set of *states*
3. A is the *initial state*,
4. S is the set of *final states*, and
5. the *transition function* $\delta\colon \mathbf{A} \times (\mathbf{R} \cup \{\epsilon\}) \to 2^{\mathbf{A}}$, where ϵ is the empty word, is defined as follows. If $P \in \mathbf{A_p} \cup \{\top\}$ then $\delta(P, R) = \emptyset$ for all $R \in \mathbf{R}$. Similarly, for the empty word ϵ we set $\delta(P, \epsilon) = \emptyset$. Furthermore, $\delta(\bot, R) = \mathbf{A_p} \cup \{\bot\}$, for all $R \in \mathbf{R} \cup \{\epsilon\}$. For all nonprimitive

[11] Note that these expanded concept descriptions may have a size exponential in the size of the original terminology, however.

concepts $A \in \mathbf{A_n}$, we set $B \in \delta(A, R)$ if and only if $T(A)$ contains a subexpression of the form $\forall R\colon B$ and $B \in \delta(A, \epsilon)$ if and only if $T(A)$ contains the atomic concept B outside of a value restriction.

A word W is called the *label* of a path from A_0 to A_n iff there is a sequence of states $A_0, A_1, \ldots, A_n$, and there is an associated sequence of symbols $R_0, \ldots, R_{n-1}$, where $R_i \in \mathbf{R} \cup \{\epsilon\}$, such that $R_0 R_1 \ldots R_{n-1} = W$ and $A_{i+1} \in \delta(A_i, R_i)$. Note that by construction of the NDFAs it follows that a label W of a path from A_0 to A_n is a role-chain such that A_n can be reached by W from A_0 in T, with the addition that if $\perp$ can be reached by a role chain W from a concept A, then A can reach $\perp$ and all primitive concepts by all role chains of the form $W\mathbf{R}^*$.

The word W is *accepted by* $\mathcal{A}$ iff W is a label of a path from the initial state to one of the final states. The set of all words accepted by $\mathcal{A}$ is called the *language accepted by* $\mathcal{A}$, written $\mathcal{L}(\mathcal{A})$. For $\mathcal{L}(\mathcal{A}_{\langle T,A,S\rangle})$ we will also write $\mathcal{L}(T, A, S)$. Based on this view, subsumption of concepts reduces to inclusion of languages accepted by the associated NDFAs.

Theorem 1 *Let T be an acyclic $\mathcal{TL}$-terminology with $A, B \in \mathbf{A}$. Then*

$$A \preceq_T B \quad \textit{iff} \quad \mathcal{L}(T, A, \{P\}) \supseteq \mathcal{L}(T, B, \{P\}) \ \textit{for all} \ P \in \mathbf{A_p} \cup \{\perp\}.$$

Proof Sketch: The proof follows by generalizing the proof of Theorem 2 in [Nebel, 1990]. Note that in order to decide inclusion of languages for the automata $\mathcal{A}_{\langle T,A,S\rangle}$ generated from acyclic terminologies it suffices to consider only words up to a length of $\|\mathbf{A_n}\|$. ∎

Intuitively, this theorem says that the set of constraints of the form $\forall W\colon P$ ($P \in \mathbf{A_p} \cup \{\perp\}$) that an instance of a concept has to obey is the same as the set of words the corresponding automata with final state P recognize. This reduction has a number of important consequences. For instance, it can be used to show that concept subsumption in acyclic $\mathcal{TL}$-terminologies is more difficult than perceived, namely, of the same complexity as the equivalence problem for NDFAs that accept finite languages, which is a co-NP-complete problem [Garey and Johnson, 1979, p. 265].

Corollary 5 *Concept subsumption in acyclic $\mathcal{TL}$-terminologies is co-NP-complete.*

Proof: Note that $C \preceq_T D$ can be reduced to subsumption of atomic concepts in polynomial time by adding appropriate definitions to the terminology, and by Theorem 1 this problem can be reduced to a language inclusion problem, i.e., subsumption is in co-NP. Since by employing Theorem 1, automaton equivalence for acyclic automata can be polynomially reduced to concept equivalence in acyclic terminologies, which can be reduced in polynomial time to subsumption (Prop. 1), subsumption is co-NP-hard. ■

This means subsumption in acyclic terminologies is co-NP-hard for all terminological formalisms considered so far—even when subsumption determination over concept descriptions is polynomial. However, subsumption determination in acyclic terminologies seems to be fairly efficient in almost all cases occurring in practice [Nebel, 1990].

Additionally, Theorem 1 shows that instead of expanding definitions and determining subsumption over concept descriptions, it is also possible to transform the terminology into a form corresponding to a deterministic automaton for which equivalence and subsumption can be decided in polynomial time—which is often more efficient than the former strategy.[12] Finally, it provides us with a tool that can be used to characterize subsumption in cyclic terminologies.

5.2 Subsumption in General Terminologies

Since component cycles can be removed from a terminology without changing the meaning of any concept (see Corollaries 1, 3, and 4), and since this can be obviously done in polynomial time, let us assume in the following that there are no such cycles.[13] This means we have to consider only restriction cycles.

In trying to generalize the view spelled out above, one notes that in general terminologies it is not enough to consider only finite role-chains, but infinite role-chains are also important (see Prop. 7). In order to capture this formally, let $\mathcal{U}(T, A, \emptyset)$ be the set of all (infinite) labels of infinite paths starting at the initial state A. Furthermore, sometimes even the atomic concepts may play a role, as is highlighted by the "Tree" terminology (12)–(13)). Formalizing this aspect, let $AW_0BW_1BW_2\ldots$ denote the infinite path starting at A which reaches B infinitely often where W_j are nonempty labels from A to B for $j = 0$ and from B to B for $j > 0$.

Based on the view spelled out above, Baader [1990] analyzed subsumption in general $\mathcal{TL}$-terminologies and characterized subsumption as follows.[14]

Theorem 2 *Let T be a general $\mathcal{TL}$-terminology, and let A, B be two atomic concepts. Then*

[12] As a matter of fact, this technique is used in most implemented terminological representation systems. See, for instance, the informal description of the first implemented terminological reasoning component in KL-ONE [Lipkis, 1982].

[13] Note that component cycles correspond to ϵ-cycles in the associated automata.

[14] In [Baader, 1990], also ϵ-cycles are covered, which we eliminated beforehand.

$A \preceq_T^{gfp} B$ *iff*

$$\mathcal{L}(T, A, \{P\}) \supseteq \mathcal{L}(T, B, \{P\}) \text{ for all } P \in \mathbf{A_p} \cup \{\perp\}.$$

$A \preceq_T^{lfp} B$ *iff*

1. $\mathcal{L}(T, A, \{P\}) \supseteq \mathcal{L}(T, B, \{P\})$ *for all* $P \in \mathbf{A_p} \cup \{\perp\}$ *and*
2. $\mathcal{U}(T, A, \emptyset) \supseteq \mathcal{U}(T, B, \emptyset)$.

$A \preceq_T B$ *iff*

1. $\mathcal{L}(T, A, \{P\}) \supseteq \mathcal{L}(T, B, \{P\})$ *for all* $P \in \mathbf{A_p} \cup \{\perp\}$ *and*
2. *for all infinite paths* $BW_0B'W_1B'W_2\ldots$ *there is a natural number* $k \geq 0$ *such that* $W_0W_1\ldots W_k$ *is a label of a path from* A *to* B' *or to* $\perp$.

Proof: Generalize the proofs in [Baader, 1990] to cover $\perp$. ∎

A consequence of this result is that *gfp*-subsumption and *lfp*-subsumption in general terminologies is more difficult than subsumption in acyclic terminologies—from a theoretical point of view.

Corollary 6 *Concept subsumption w.r.t. lfp- and gfp-semantics in general* $\mathcal{TL}$*-terminologies is PSPACE-complete.*

Proof: Since the PSPACE-complete problem of deciding language inclusion for general NDFAs and the problem of deciding concept subsumption for general $\mathcal{TL}$-terminologies are interreducible for *gfp*-semantics, *gfp*-subsumption is PSPACE-complete. For a proof of PSPACE-completeness of *lfp*-subsumption see [Baader, 1990]. ∎

Additionally, it shows that *gfp*-semantics has indeed the conceptually easiest characterization. Furthermore, it leads directly to deterministic algorithms for *gfp*-subsumption determination in general $\mathcal{TL}$-terminologies, namely, a transformation of the NDFA corresponding to the terminology to a deterministic automaton, on which language inclusion can be decided in polynomial time. Although, in general, the set of states which can be reached by the initial state increases exponentially when transforming a nondeterministic into an equivalent deterministic automaton, I expect that this behavior occurs rather seldomly in the context of terminologies because terminologies are usually formulated in a way such that the corresponding NDFA is "almost" deterministic.

Unfortunately, descriptive semantics does not lead to such a straightforward result. In [Baader, 1990] it is shown that subsumption w.r.t. descriptive semantics can be reduced to an inclusion problem for a class of languages containing infinite words (languages accepted by Büchi automata).

An alternative characterization in terms of the *structure* of automata can be given when the corresponding *deterministic* automaton (DFA) is considered.[15] Let $\mathcal{A}_{\langle T,A,S\rangle} = (\mathbf{R}, \mathbf{A}, \delta, A, S)$ be a NDFA as defined above. Then $\widehat{\mathcal{A}}_{\langle T,A,\widehat{S}\rangle} = (\mathbf{R}, \widehat{\mathbf{A}}, \widehat{\delta}, A, \widehat{S})$ shall denote the corresponding DFA, which can be created using the *subset-construction* (see, e.g., [Lewis and Papadimitriou, 1981, p. 59ff]). Each state $X \in \widehat{\mathbf{A}}$ is a subset of the states in the NDFA, where singletons are identified with elements.

For notational convenience, $\widehat{\delta}^*$ will be used to denote the canonical extension of the transition function $\widehat{\delta}$ to words, i.e., $\widehat{\delta}^*(X, \epsilon) = X$ and $\widehat{\delta}^*(X, RW) = \widehat{\delta}^*(\widehat{\delta}(X, R), W)$.

Using these assumptions, concept equivalence for descriptive semantics can be characterized in terms of language equivalence and the structure of the DFA.[16] Informally, two concepts A and B are equivalent if and only if the corresponding automatons accept the same language and there are not two different cycles with identical labels in the DFAs such that one is reachable from A by a word W and the other one is reachable by the same word W from B.

Proposition 10 *Let T be a general $\mathcal{TL}$-terminology. Then $A \approx_T B$ iff*

1. $\mathcal{L}(T, A, \{P\}) = \mathcal{L}(T, B, \{P\})$ *for all* $P \in \mathbf{A_P} \cup \{\bot\}$ *and*

2. *for all words* $W \in \mathbf{R}^*$*, if* $X = \widehat{\delta}^*(A, W)$*,* $Y = \widehat{\delta}^*(B, W)$*, and* $X \neq Y$*, then there is no word* $V \in \mathbf{R}^+$ *such that* $X = \widehat{\delta}^*(X, V)$ *and* $Y = \widehat{\delta}^*(Y, V)$.

Proof: For the "if" direction assume that the concepts are not equivalent. By Theorem 2, either the languages of the automata are not identical—which violates the first condition in the proposition—or there exists w.l.g. an infinite path in in the NDFA of the form $AW_1B'W_2B'\ldots$ such that for no $k \geq 0$ there is a label $W_1W_2\ldots W_k$ of a path from B to B'.

Note that for a path of the above form in the NDFA starting at A there is a corresponding path in the DFA with the same label and there is at least one state $Z \in \widehat{\mathbf{A}}$ that appears infinitely often and $B' \in Z$, i.e. there is an infinite path $AV_1ZV_2Z\ldots$ in the DFA. Starting at B in the DFA, we have a similar path and a sequence of states $Z_1, Z_2, \ldots$ such that $BV_1Z_1V_2Z_2\ldots$.

[15] A DFA has no ϵ-transitions and the transition function does not map states and symbols to sets of states but to single states.

[16] Note that it is not possible to describe concept subsumption in terms of the structure of the DFA.

Since there are only finitely many states in the DFA, we know that there are i, j such that $Z_i = Z_j$. Assuming that $Z = Z_i = Z_j$ would result in the conclusion that there is a number $k \geq 0$ s.t. there is a path $BW_1W_2 \ldots W_kB'$ in the NDFA, which contradicts our assumption. Assuming that $Z \neq Z_i$ violates the second condition in the proposition. Hence, if the concepts are not equivalent, then one of the conditions will be violated.

For the "only if" direction assume that the concepts are equivalent, but one of the conditions is violated. If the first condition is violated, then by Theorem 2, the concepts cannot be equivalent. If the second condition is violated, then there are words W, V and states $X, Y \in \widehat{\mathbf{A}}$ such that the condition is violated. Without loss of generality, let us assume $X - Y \neq \emptyset$. Choose one element B' in $X - Y$ such that there is path $B'V^nB'$ in the NDFA, for some $n > 0$. Such an element exists because of the following reasons. Since $X = \widehat{\delta}^*(X, V)$, each element in X must be reachable in the NDFA by V from some element in X. Assuming that there is no state $B' \in X - Y$ s.t. $B'V^nB'$ is a path in the NDFA leads to the conclusion that some elements in $X - Y$ must be reachable in the NDFA by V from some elements in $Y \cap X$. This, however, means $Y \neq \widehat{\delta}^*(Y, V)$.

Finally, using the chosen state B', it is possible to find an infinite path of the form $AWB'V^nB'V^n \ldots$ in the NDFA such that there is no natural number k with $BWV^{n^k}B'$. Thus, the concepts cannot be equivalent by Theorem 2. ■

This observation leads to a PSPACE decision procedure for equivalence (and, thus, subsumption) of concepts w.r.t. to descriptive semantics.[17]

Corollary 7 *Concept subsumption w.r.t. to descriptive semantics in general $\mathcal{TL}$-terminologies is in PSPACE.*

Proof: Guessing two words W, V and two sets of states $S, S' \subseteq \mathbf{A}$, we can verify in polynomial space that the the second condition in Prop. 10 is violated. Since the first condition can be checked in polynomial space, as well, concept equivalence is in PSPACE. Since concept equivalence and subsumption are interreducible in linear time, concept subsumption is also in PSPACE. ■

It is by no means obvious, however, whether subsumption w.r.t. descriptive semantics is PSPACE-complete or easier.

[17] The same result follows from the reduction to inclusion of languages accepted Büchi automatons [Baader, 1990].

5.3 Terminological Cycles in more Powerful Languages

After having now an idea what subsumption algorithms for general terminologies look like and how difficult subsumption determination can be, there is the natural question of how to extend this result to more powerful terminological languages.

In [Nebel, 1989, Ch. 5], a slightly more powerful language was analyzed with respect to terminological cycles. This language contains $\mathcal{TLN}$ plus subroles and negation of primitive concepts. It was shown that subsumption w.r.t. descriptive semantics is still decidable for this formalism by using an argument to the effect that it is always possible to consider only models up to a certain finite size in order to decide subsumption. Generalizing this argument, it seems possible to prove decidability for other languages. However, there are, of course, limits. In order to demonstrate where these limits are, $\mathcal{TL}$ will be extended in a way such that it captures an essential subset of the terminological language used in the CLASSIC system [Brachman *et al.*, 1989; Borgida *et al.*, 1989].

Let us assume a set $\mathbf{F}$ of *single-valued roles*, also called *features*[18] in the following, (denoted by f) that is a subset of $\mathbf{R}$. The interpretation of these features is constrained by

$$\|[\![f]\!]^{\mathcal{I}}(d)\| \leq 1 \quad \textit{for all } d \in \mathcal{D} \textit{ and all } f \in \mathbf{F},$$

Chains of features are denoted by v and w. These are interpreted in the same way as role chains (see Section 4.2). Finally, we define a new description-forming operator $v \downarrow w$, called *coreference constraint*, intended to denote all elements such that the role-filler of v is identical to the role-filler of w, formally:

$$[\![v \downarrow w]\!]^{\mathcal{I}} \quad = \quad \{d \in \mathcal{D} |\, [\![v]\!]^{\mathcal{I}}(d) = [\![w]\!]^{\mathcal{I}}(d)\}$$

Adding this operator to $\mathcal{TL}$ results in a terminological language—we will call $\mathcal{TLC}$—with a very interesting property. Subsumption over concept descriptions is polynomial [Donini *et al.*, 1990], i.e., subsumption in acyclic $\mathcal{TLC}$-terminologies is decidable,[19] but if terminological cycles come into play, subsumption becomes undecidable.

The claim above will be shown by reducing the word problem in Thue systems to subsumption in general $\mathcal{TLC}$-terminologies using the same proof

[18] I use the term *features* because single-valued roles are essentially the same as features in *feature logic* (see, e.g., [Nebel and Smolka, 1990]).

[19] Note that coreference constraints lead to undecidability of subsumption in acyclic terminologies if the role-chains in the constraint are not features but ordinary roles [Schmidt-Schauß, 1989].

technique as in [Schmidt-Schauß, 1989] and [Smolka, 1989, Sect. 8]. A *Thue system* is a finite set of unordered pairs $\mathcal{S} = \left\{\{v_i, w_i\}\right\}$, where v_i and w_i are words over an alphabet $\mathbf{F}$. Two words $v, w \in \mathbf{F}^*$ are *interderivable in one step*, written $v \overset{\mathcal{S}}{\leftrightarrow} w$, iff

$$v = xv_jy \textit{ and } w = xw_jy \textit{ for some } \{v_j, w_j\} \in \mathcal{S}$$

Two words v, w are *interderivable* iff they are related by the transitive, reflexive closure of $\overset{\mathcal{S}}{\leftrightarrow}$, which will be denoted by $\overset{\mathcal{S}}{\Leftrightarrow}$. Note that $\overset{\mathcal{S}}{\Leftrightarrow}$ is an equivalence relation on $\mathbf{F}^*$. The *word problem in Thue systems* is the problem to decide whether $v \overset{\mathcal{S}}{\Leftrightarrow} w$. It is well known that this problem is undecidable (see e.g. [Lewis and Papadimitriou, 1981, Sect. 6.4]).

Lemma 2 *Let* $\mathcal{S} = \left\{\{v_i, w_i\}\right\}$ *be a Thue system over an alphabet* $\mathbf{F} = \{f_j\}$, *and let* T *be the following* $\mathcal{TLC}$*-terminology:*

$$T(A) \quad = \quad \textstyle\bigsqcap \forall f_j \colon A \sqcap \bigsqcap v_i \downarrow w_i.$$

Then

$$v \overset{\mathcal{S}}{\Leftrightarrow} w \quad \textit{iff} \quad [\![A]\!]^{\mathcal{I}} \subseteq [\![v \downarrow w]\!]^{\mathcal{I}} \textit{ for all models } \mathcal{I} \textit{ of } T.$$

Proof: Let $\mathcal{I}$ be a model of T and assume $v \overset{\mathcal{S}}{\leftrightarrow} w$, where $v = xv_iy$ and $w = xw_iy$. Now we know for all $d \in [\![A]\!]^{\mathcal{I}}$: $[\![x]\!]^{\mathcal{I}}(d) \subseteq [\![A]\!]^{\mathcal{I}}$ (because of $\bigsqcap \forall f_j \colon A$). From that it follows that $[\![xv_i]\!]^{\mathcal{I}}(d) = [\![xw_i]\!]^{\mathcal{I}}(d)$, hence $[\![v]\!]^{\mathcal{I}}(d) = [\![w]\!]^{\mathcal{I}}(d)$ for all $d \in [\![A]\!]^{\mathcal{I}}$. By induction, we can conclude that $[\![A]\!]^{\mathcal{I}} \subseteq [\![v \downarrow w]\!]^{\mathcal{I}}$ if $v \overset{\mathcal{S}}{\Leftrightarrow} w$.

For the other direction assume that $[\![A]\!]^{\mathcal{I}} \subseteq [\![v \downarrow w]\!]^{\mathcal{I}}$ for all models $\mathcal{I}$ of T. Let $[x]_{\mathcal{S}}$ denote the equivalence class of x w.r.t. $\overset{\mathcal{S}}{\Leftrightarrow}$. Now we construct a particular model of T as follows:

$$\begin{aligned} \mathcal{D} &= \{[x]_{\mathcal{S}} \mid x \in \mathbf{F}^*\} \\ [\![f_j]\!]^{\mathcal{I}} &= \left\{[x]_{\mathcal{S}} \mapsto \{[xf_j]_{\mathcal{S}}\}\right\} \\ [\![A]\!]^{\mathcal{I}} &= \mathcal{D}. \end{aligned}$$

Obviously, $\mathcal{I}$ is a model of T since

1. for all $d \in [\![A]\!]^{\mathcal{I}}$ it holds that $[\![f_j]\!]^{\mathcal{I}}(d) \in [\![A]\!]^{\mathcal{I}}$ for all $f_j \in \mathbf{F}$, and

2. $[\![v_i]\!]^{\mathcal{I}}(d) = [\![w_i]\!]^{\mathcal{I}}(d)$ for all pairs $\{v_i, w_i\}$ of the Thue system $\mathcal{S}$ because $[\![v_i]\!]^{\mathcal{I}}([x]_{\mathcal{S}}) = \{[xv_i]_{\mathcal{S}}\} = \{[xw_i]_{\mathcal{S}}\} = [\![w_i]\!]^{\mathcal{I}}([x]_{\mathcal{S}})$.

Because of our assumption, we know that $[\![v]\!]^{\mathcal{I}}(d) = [\![w]\!]^{\mathcal{I}}(d)$ for all elements $d \in [\![A]\!]^{\mathcal{I}} = \mathcal{D}$. Thus, in particular, we have $[\![v]\!]^{\mathcal{I}}([\epsilon]_{\mathcal{S}}) = [\![w]\!]^{\mathcal{I}}([\epsilon]_{\mathcal{S}})$, hence $[\epsilon v]_{\mathcal{S}} = [\epsilon w]_{\mathcal{S}}$, hence $[v]_{\mathcal{S}} = [w]_{\mathcal{S}}$, which means $v \overset{\mathcal{S}}{\Leftrightarrow} w$. ■

From that the undecidability of subsumption w.r.t. descriptive semantics follows immediately.

Theorem 3 *Subsumption w.r.t. descriptive semantics in general $\mathcal{TLC}$-terminologies is undecidable.*

Proof: Since the word problem in Thue systems is undecidable and it can be reduced to subsumption w.r.t. to descriptive semantics, subsumption w.r.t. descriptive semantics is undecidable. ■

As should be obvious, adding coreference constraints to our language does not change the monotonicity of Γ, i.e., it makes sense to ask about the behavior of subsumption under *lfp*- and *gfp*-semantics. It is easy to see that the above result applies to *gfp*-semantics, as well.

Corollary 8 *Subsumption w.r.t. gfp-semantics in general $\mathcal{TLC}$-terminologies is undecidable.*

Proof: Since the first part of Lemma 2 applies to all models, it applies to *gfp*-models, as well. The model constructed in the second part of the proof is a *gfp*-model, as can be easily verified. ■

Unfortunately, the proof technique used above does not seem to be usable for showing *lfp*-subsumption to be undecidable. However, since we ruled out this semantics in Section 4.5 because of other reasons, we will not dig deeper at this point.

In general, these undecidability results mean that terminological cycles are not always tolerable. In particular, when coreference constraints are part of the language, the unrestricted use of terminological cycles should be prohibited.

6 Conclusions

Terminological cycles present conceptual and algorithmic problems for terminological representation systems. As shown in Section 4, it is possible to extend the standard semantics of terminological representation formalisms to cover cyclic terminologies. However, it is not completely obvious which style of semantics is the best one. Greatest fixpoint semantics has the advantage that it leads to a conceptually simple subsumption relation, which is identical to language inclusion of the nondeterministic automata corresponding to the terminology. However this style of semantics cannot be extended to cover more powerful formalisms. Descriptive semantics, on the other hand, is the most straightforward extension of the standard semantics, covers all conceivable terminological formalisms, and permits interesting inferences in hybrid representation systems, but leads to subsumption relationships which are not fully obvious—except one considers the structure of the deterministic automata corresponding to a terminology. Finally, it was shown that the unrestricted use of terminological cycles can lead to undecidability. In particular, it was shown that adding terminological cycles to an essential subset of the terminological formalism used in the CLASSIC systems results in undecidability.

Acknowledgement

This paper greatly profited from discussions with Franz Baader, Kai von Luck, Bob MacGregor, Bernd Mahr, Peter Patel-Schneider, Christof Peltason, Albrecht Schmiedel, Jim Schmolze, Gert Smolka, and Marc Vilain. In particular Franz helped me to get a deeper understanding of some of the subtle points in subsumption determination in cyclic terminologies. Also, Alfred Kobsa and Peter provided valuable comments on this paper.

References

[Baader, 1990] Franz Baader. Terminological cycles in KL-ONE-based KR-languages. In *Proceedings of the 8th National Conference of the American Association for Artificial Intelligence*, Boston, Mass., 1990. To appear. A version of this paper containg proofs is available as Research Report RR-90-01, German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, West Germany, Januar 1990.

[Borgida *et al.*, 1989] Alexander Borgida, Ronald J. Brachman, Deborah L. McGuinness, and Lori Alperin Resnick. CLASSIC: a structural data model for objects. In *Proceedings of the 1989 ACM SIGMOD International Conference on Mangement of Data*, pages 59–67, Portland, Oreg., June 1989.

[Brachman and Levesque, 1982] Ronald J. Brachman and Hector J. Levesque. Competence in knowledge representation. In *Proceedings of the 2nd National Conference of the American Association for Artificial Intelligence*, pages 189–192, Pittsburgh, Pa., August 1982.

[Brachman and Levesque, 1984] Ronald J. Brachman and Hector J. Levesque. The tractability of subsumption in frame-based description languages. In *Proceedings of the 4th National Conference of the American Association for Artificial Intelligence*, pages 34–37, Austin, Tex., August 1984.

[Brachman and Schmolze, 1985] Ronald J. Brachman and James G. Schmolze. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9(2):171–216, April 1985.

[Brachman *et al.*, 1985] Ronald J. Brachman, Victoria Pigman Gilbert, and Hector J. Levesque. An essential hybrid reasoning system: Knowledge and symbol level accounts in KRYPTON. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 532–539, Los Angeles, Cal., August 1985.

[Brachman *et al.*, 1989] Ronald J. Brachman, Alexander Borgida, Deborah L. McGuinness, and Lori Alperin Resnick. The CLASSIC knowledge representation system, or, KL-ONE: The next generation. In *Preprints of the Workshop on Formal Aspects of Semantic Networks*, Two Harbors, Cal., February 1989.

[Brachman, 1979] Ronald J. Brachman. On the epistemological status of semantic networks. In N. V. Findler, editor, *Associative Networks: Representation and Use of Knowledge by Computers*, pages 3–50. Academic Press, New York, N.Y., 1979.

[Donini *et al.*, 1990] Francesco M. Donini, Bernhard Hollunder, Maurizio Lenzerini, Alberto Marchetti Spaccamela, Daniele Nardi, and Werner Nutt. The frontier of tractability for concept description languages. DFKI Report, German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, West Germany, 1990. In preparation.

[Edelmann and Owsnicki, 1986] Jürgen Edelmann and Bernd Owsnicki. Data models in knowledge representation systems: A case study. In C.-R. Rollinger and W. Horn, editors, *GWAI-86 und 2. Österreichische Artificial-Intelligence-Tagung*, pages 69–74. Springer-Verlag, Berlin, West Germany, 1986.

[Garey and Johnson, 1979] Michael R. Garey and David S. Johnson. *Computers and Intractability—A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, Cal., 1979.

[Hollunder *et al.*, 1990] Bernhard Hollunder, Werner Nutt, and Manfred Schmidt-Schauß. Subsumption algorithms for concept description languages. DFKI Report, German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, West Germany, 1990. Forthcoming.

[Kaczmarek *et al.*, 1986] Thomas S. Kaczmarek, Raymond Bates, and Gabriel Robins. Recent developments in NIKL. In *Proceedings of the 5th National Conference of the American Association for Artificial Intelligence*, pages 978–987, Philadelphia, Pa., August 1986.

[Kobsa, 1989] Alfred Kobsa. The SB-ONE knowledge representation workbench. In *Preprints of the Workshop on Formal Aspects of Semantic Networks*, Two Harbors, Cal., February 1989.

[Levesque and Brachman, 1987] Hector J. Levesque and Ronald J. Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3:78–93, 1987.

[Lewis and Papadimitriou, 1981] Harry R. Lewis and Christos H. Papadimitriou. *Elements of the Theory of Computation*. Prentice-Hall, Englewood Cliffs, N.J., 1981.

[Lipkis, 1982] Thomas Lipkis. A KL-ONE classifier. In J. G. Schmolze and R. J. Brachman, editors, *Proceedings of the 1981 KL-ONE Workshop*, pages 128–145, Cambridge, Mass., 1982. The proceedings have been published as BBN Report No. 4842 and Fairchild Technical Report No. 618.

[Lloyd, 1984] John W. Lloyd. *Foundations of Logic Programming*. Springer-Verlag, Berlin, West Germany, 1984.

[MacGregor, 1988] Robert MacGregor. A deductive pattern matcher. In *Proceedings of the 7th National Conference of the American Association for Artificial Intelligence*, pages 403–408, Saint Paul, Minn., August 1988.

[Nebel and Smolka, 1990] Bernhard Nebel and Gert Smolka. Representation and reasoning with attributive descriptions. In K.-H. Bläsius, U. Hedtstück, and C.-R. Rollinger, editors, *Sorts and Types in Artificial Intelligence*. Springer-Verlag, Berlin, West Germany, 1990. To appear. Also available as IWBS Report 81, IBM Germany Scientific Center, IWBS, Stuttgart, West Germany, September 1989.

[Nebel and von Luck, 1988] Bernhard Nebel and Kai von Luck. Hybrid reasoning in BACK. In Z. W. Ras and L. Saitta, editors, *Methodologies for*

Intelligent Systems, volume 3, pages 260–269. North-Holland, Amsterdam, Holland, 1988.

[Nebel, 1988] Bernhard Nebel. Computational complexity of terminological reasoning in BACK. *Artificial Intelligence*, 34(3):371–383, April 1988.

[Nebel, 1989] Bernhard Nebel. *Reasoning and Revision in Hybrid Representation Systems*. PhD thesis, Universität des Saarlandes, Saarbrücken, West Germany, June 1989. To appear in Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, West Germany.

[Nebel, 1990] Bernhard Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43:235–249, 1990. To appear.A preliminary version is available as IWBS Report 82, IBM Germany Scientific Center, IWBS, Stuttgart, West Germany, October 1989.

[Neches *et al.*, 1985] Robert Neches, William R. Swartout, and Johanna D. Moore. Explainable (and maintainable) expert systems. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 382–389, Los Angeles, Cal., August 1985.

[Owsnicki-Klewe, 1988] Bernd Owsnicki-Klewe. Configuration as a consistency maintenance task. In W. Hoeppner, editor, *GWAI-88. 12th German Workshop on Artificial Intelligence*, pages 77–87. Springer-Verlag, Berlin, West Germany, 1988.

[Patel-Schneider *et al.*, 1984] Peter F. Patel-Schneider, Ronald J. Brachman, and Hector J. Levesque. ARGON: Knowledge representation meets information retrieval. In *Proceedings of the 1st Conference on Artificial Intelligence Applications*, pages 280–286, Denver, Col., 1984.

[Patel-Schneider, 1984] Peter F. Patel-Schneider. Small can be beautiful in knowledge representation. In *Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems*, pages 11–16, Denver, Colo., 1984. An extended version including a KANDOR system description is available as AI Technical Report No. 37, Palo Alto, Cal., Schlumberger Palo Alto Research, October 1984.

[Patel-Schneider, 1986] Peter F. Patel-Schneider. A four-valued semantics for frame-based description languages. In *Proceedings of the 5th National Conference of the American Association for Artificial Intelligence*, pages 344–348, Philadelphia, Pa., August 1986.

[Patel-Schneider, 1989a] Peter F. Patel-Schneider. A four-valued semantics for terminological logics. *Artificial Intelligence*, 38(3):319–351, April 1989.

[Patel-Schneider, 1989b] Peter F. Patel-Schneider. Undecidability of subsumption in NIKL. *Artificial Intelligence*, 39(2):263–272, June 1989.

[Schild, 1988] Klaus Schild. Undecidability of $\mathcal{U}$. KIT Report 67, Department of Computer Science, Technische Universität Berlin, Berlin, West Germany, October 1988.

[Schmidt-Schauß and Smolka, 1990] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with unions and complements. *Artificial Intelligence*, 1990. To appear. A preliminary version of this paper is available as IWBS Report 68, IBM Germany Scientific Center, IWBS, Stuttgart, West Germany, June 1989.

[Schmidt-Schauß, 1989] Manfred Schmidt-Schauß. Subsumption in KL-ONE is undecidable. In R. J. Brachman, H. J. Levesque, and R. Reiter, editors, *Proceedings of the 1st International Conference on Principles of Knowledge Representation and Reasoning*, pages 421–431, Toronto, Ont., May 1989.

[Schmolze and Brachman, 1982] James G. Schmolze and Ronald J. Brachman, editors. *Proceedings of the 1981 KL-ONE Workshop*, Cambridge, Mass., 1982. Bolt, Beranek, and Newman Inc. BBN Report No. 4842.

[Schmolze and Israel, 1983] James G. Schmolze and David J. Israel. KL-ONE: Semantics and classification. In *Research in Knowledge Representation and Natural Language Understanding, BBN Technical Report, No. 5421*, pages 27–39. Bolt, Beranek, and Newman Inc., Cambridge, Mass., 1983.

[Schmolze, 1989] James G. Schmolze. The language and semantics of NIKL. Technical Report 89-4, Department of Computer Science, Tufts University, Medford, Mass., September 1989.

[Smolka, 1989] Gert Smolka. Feature constraint logics for unification grammars. IWBS Report 93, IBM Germany Scientific Center, IWBS, Stuttgart, West Germany, November 1989. To appear in the Proccedings of the Workshop on Unification Formalisms—Syntax, Semantics, and Implementation, Titisee, April 1988, The MIT Press.

[Sondheimer and Nebel, 1986] Norman K. Sondheimer and Bernhard Nebel. A logical-form and knowledge-base design for natural language generation. In *Proceedings of the 5th National Conference of the American Association for Artificial Intelligence*, pages 612–618, Philadelphia, Pa., August 1986.

[Swartout and Neches, 1986] William R. Swartout and Robert Neches. The shifting terminological space: An impediment to evolvability. In *Proceedings of the 5th National Conference of the American Association for Artificial Intelligence*, pages 936–941, Philadelphia, Pa., August 1986.

[Tarski, 1955] Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.

[Vilain, 1985] Marc B. Vilain. The restricted language architecture of a hybrid representation system. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 547–551, Los Angeles, Cal., August 1985.

[von Luck *et al.*, 1987] Kai von Luck, Bernhard Nebel, Christof Peltason, and Albrecht Schmiedel. The anatomy of the BACK system. KIT Report 41, Department of Computer Science, Technische Universität Berlin, Berlin, West Germany, January 1987.

[Webber and Bobrow, 1980] Bonnie Lynn Webber and Robert J. Bobrow. Knowledge representation for syntactic/semantic processing. In *Proceedings of the 1st National Conference of the American Association for Artificial Intelligence*, pages 316–323, Stanford, Cal., August 1980.