Published in O. Herzog, C. Rollinger (eds.), *Textunderstanding in LILOG: Integrating Computational Linguistics and Artificial Intelligence*, p. 439–452. Springer-Verlag, Berlin, Heidelberg, New York, 1991.

Attributive Description Formalisms ...and the Rest of the World

Bernhard Nebel and Gert Smolka
German Research Center for Artificial Intelligence
Saarbrücken,Germany

Abstract

Research in knowledge representation has led to the development of so-called terminological logics, the purpose of which is to support the representation of the conceptual and terminological part of Artificial Intelligence applications. Independently, in computational linguistics, so-called feature logics have been developed which are aimed at representing the semantic and syntactic information natural language sentences convey. Since both of these logics rely mainly on attributes as the primary notational primitives for representing knowledge, they can be jointly characterized as attributive description formalisms.

Although the intended applications for terminological logics and feature logics are not identical, and the computational services of systems based on the respective formalisms are quite different for this reason, the logical foundations turn out to be very similar – as we pointed out elsewhere. In this paper, we will show how attributive description formalisms relate to "the rest of the world." Recently, a number of formal results in the area of attributive description formalisms have been obtained by exploiting other research fields, such as formal language theory, automata theory, and modal logics. This connection between these different fields of formal research will be highlighted in the sequel.

1 Introduction

Terminological logics, which have their roots in the knowledge representation formalism KL-ONE [Brachman, 1979; Brachman and Schmolze, 1985], have been developed to support the representation of the conceptual and terminological part of Artificial Intelligence applications.

Starting with *primitive concepts* and *attributes* (in this context usually called *roles*), new concepts are defined by employing attributive descriptions. For instance, given the *concept* Human and the *attribute* child, the concept of a Parent can be defined by the description

a Human who has at least one child who in turn is a Human,

or, more formally,

$$\mathsf{Parent} = \mathsf{Human} \sqcap \exists \mathsf{child}\colon \mathsf{Human}.$$

The main computational services provided by terminological representation systems are the computation of the *concept hierarchy* according to the *subsumption* relation between concepts and the computation of *instance relationships* between concepts and objects of the application domain.

Feature logics grew out of research in computational linguistics. They form the constraint logic underlying the family of *unification grammars* that originated with Lexical Functional Grammar (LFG) [Kaplan and Bresnan, 1982] and Functional Unification Grammar (FUG) [Kay, 1979; Kay, 1985]. In unification grammars, syntactic and semantic objects are described by employing attributive descriptions. For instance, the class of linguistic objects that are

third-person singular noun phrases

can be described formally as follows [Shieber, 1986]:

$$\begin{bmatrix} \mathsf{cat:} & \mathsf{NP} \\ \mathsf{agreement:} & \begin{bmatrix} \mathsf{number:} & \mathsf{singular} \\ \mathsf{person:} & \mathsf{third} \end{bmatrix} \end{bmatrix}$$

or, in a linear notation as:

$$\mathsf{cat}\colon \mathsf{NP} \sqcap \mathsf{agreement}\colon (\mathsf{number}\colon \mathsf{singular} \sqcap \mathsf{person}\colon \mathsf{third}).$$

While parsing a sentence, such descriptions are combined by "unification," and, in the end, the combined descriptions provide the syntactic and semantic structure of the sentence. One main step during this process is the test whether a newly formed description is *satisfiable*, i.e., describes any linguistic structure at all.

As we pointed out in [Nebel and Smolka, 1990], terminological logics and feature logics are closely related. Although the intended applications are not identical, and for this reason, the computational services of systems based on the respective logics are quite different, the logical foundations turn out to be the same. Both logics employ restrictions on attributes as the primary notational primitives and are best formalized using a Tarski-style model theory. The main difference between terminological logics and feature logics is that the former permit set-valued attributes (called *roles*), while the latter permit only single-valued attributes (called *features*). This seemingly minor difference has drastic consequences as it amounts to computational complexity. Nevertheless, for a large range of problems, formal results apply to both kinds of logics.

In the the LILOG project, there two applications of attributive descriptions. The STUF formalism [Bouma *et al.*, 1988; Dörre and Seiffert, 1991] is based on feature logic and is employed in the linguistic components. The knowledge representation language L-LILOG [Pletat and von Luck, 1990; Pletat, 1991] is a hybrid formalism combining predicate logic and attributive descriptions.

The remainder of the paper is organized as follows. In the next section, we will briefly introduce the logical foundations of terminological and feature logics. Sect. 3 shows the applicability of results from *automata theory* to attributive description languages in terms of computational complexity results and algorithms. Sect. 4 summarizes a number of undecidability results which have been obtained by reductions using the *word problem for Thue systems*. In fact, for some proofs a slightly stronger condition is necessary, namely, that the semigroup generated by the Thue system is a group. In particular, we consider the problem of determining satisfiability for feature terms containing *functional uncertainty* in the case that the feature logic is propositionally complete. In Sect. 5, a correspondence between a certain terminological logic and the *propositional polymodal logic* $\mathrm{K}_{(m)}$ is considered, which leads to quite a number of interesting applications of results from modal and dynamic logic to attributive description formalisms. Finally, in the conclusion we will sketch some applications of results achieved in the area of attribute descriptions to other research fields. A summary of the relations discussed in the paper is shown in Figure 1.

2 Logical Foundations

While terminological logics were introduced originally with an informal semantics only, it quickly became obvious that a formal semantics is necessary to describe the intended meaning – and the obvious candidate, first-order predicate calculus and its associated model theory, was used for this purpose [Schmolze and Israel, 1983; Brachman and Levesque, 1984]. A similar process took place in the area of unification grammars [Kasper and Rounds, 1986; Johnson, 1987; Smolka, 1988].

This logical reconstruction revealed in both cases that the formalisms correspond to subsets of ordinary first-order predicate logic. Although this correspondence is very helpful for understanding the meaning of the formalism and yields a firm base for extensions, it does not help much in determining the computational properties. Nevertheless, a logical foundation is a necessary prerequisite for an analysis of computational properties. In the following, the logical foundations of attributive description formalisms are briefly recalled.

In terminological logics, we start with an alphabet $\mathbf{C}$ of **concept symbols** (denoted by C) and an alphabet $\mathbf{R}$ of **role symbols** (denoted by R), which

Figure 1: Attributive description formalisms and the relation to the rest of the world

are disjoint. Concept symbols are intended to denote some subset of a domain, and role symbols are intended to denote unary, set-valued functions or, equivalently, two-place relations on the domain.[1] From concept and role symbols, complex **concept descriptions** (denoted by D) are composed using a variety of description-forming operations. In order to give an example, the language $\mathcal{ALC}$ will be specified, originally introduced by Schmidt-Schauß and Smolka [1991]:

$$D \longrightarrow C \mid \top \mid \bot \mid D \sqcap D' \mid D \sqcup D' \mid \neg D \mid \forall R\colon D \mid \exists R\colon D.$$

The formal meaning of concept descriptions built according to the above rule is given by an interpretation $\mathcal{I} = (\mathbf{D}^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\mathbf{D}^{\mathcal{I}}$ (the **domain**) is an arbitrary nonempty set and $\cdot^{\mathcal{I}}$ (the **interpretation function**) is a function such that:

$$\begin{aligned} C^{\mathcal{I}} &\subseteq \mathbf{D}^{\mathcal{I}} \\ R^{\mathcal{I}} &\subseteq \mathbf{D}^{\mathcal{I}} \times \mathbf{D}^{\mathcal{I}}. \end{aligned}$$

The denotation of complex concept descriptions is given inductively by:

$$\top^{\mathcal{I}} = \mathbf{D}^{\mathcal{I}}$$

[1] We will use both notations interchangeably.

$$\begin{aligned}
\bot^{\mathcal{I}} &= \emptyset \\
(D \sqcap D')^{\mathcal{I}} &= D^{\mathcal{I}} \cap D'^{\mathcal{I}} \\
(D \sqcup D')^{\mathcal{I}} &= D^{\mathcal{I}} \cup D'^{\mathcal{I}} \\
(\neg D)^{\mathcal{I}} &= \mathbf{D}^{\mathcal{I}} - D^{\mathcal{I}} \\
(\forall R\colon D)^{\mathcal{I}} &= \{d \in \mathbf{D}^{\mathcal{I}} \,|\, R^{\mathcal{I}}(d) \subseteq D^{\mathcal{I}}\} \\
(\exists R\colon D)^{\mathcal{I}} &= \{d \in \mathbf{D}^{\mathcal{I}} \,|\, R^{\mathcal{I}}(d) \cap D^{\mathcal{I}} \neq \emptyset\}.
\end{aligned}$$

Based on this semantics, the notion of *subsumption* mentioned above is defined as set-inclusion. A concept D is **subsumed** by another concept D', written $D \preceq D'$, iff $(D)^{\mathcal{I}} \subseteq (D')^{\mathcal{I}}$ for every interpretation $\mathcal{I}$. From this relation, a concept hierarchy can be computed. If the logic is extended to describe single objects by using role and concept symbols, then the notion of *instance relationship* can be formalized as set-membership in concepts.

Note that one can think of quite different terminological logics employing, for instance, role-forming operators, cardinality restrictions on roles, and so on. Indeed, quite a number of different representation systems have been built using a variety of terminological logics (for a survey, see [Nebel, 1990a]).

Turning now to feature logic, we notice that the formalization of so-called *feature terms* resembles the formalization of concept descriptions. In feature logics, we start with three pairwise disoint alphabets, namely, a set $\mathbf{S}$ of **sort symbols** (denoted by S), a set $\mathbf{F}$ of **feature symbols** (denoted by f), and a set $\mathbf{A}$ of **atoms** (denoted by a). Based on that, the following rule (see, e.g., [Smolka, 1988]) specifies how to built complex **feature terms** (denoted by F):

$$F \longrightarrow a \,|\, S \,|\, \top \,|\, \bot \,|\, F \sqcap F' \,|\, F \sqcup F' \,|\, \neg F \,|\, (f_1 \ldots f_n)\colon F \,|\, (f_{1,1} \ldots f_{1,m}) \downarrow (f_{2,1} \ldots f_{2,n}).$$

The formal meaning is provided by interpretations $\mathcal{I} = (\mathbf{D}^{\mathcal{I}}, \cdot^{\mathcal{I}})$, also called **feature algebras** in this context, where $\mathbf{D}^{\mathcal{I}}$ is a nonempty set and $\cdot^{\mathcal{I}}$ is a function such that:

$$\begin{aligned}
a^{\mathcal{I}} &\in \mathbf{D}^{\mathcal{I}} \\
S^{\mathcal{I}} &\subseteq \mathbf{D}^{\mathcal{I}} \\
f^{\mathcal{I}} &\subseteq \mathbf{D}^{\mathcal{I}} \times \mathbf{D}^{\mathcal{I}}.
\end{aligned}$$

Additionally, the restrictions

$$\begin{aligned}
(d, e), (d, e') \in f^{\mathcal{I}} &\implies e = e' \\
a \neq b &\implies a^{\mathcal{I}} \neq b^{\mathcal{I}} \\
a \in \mathbf{A}, f \in \mathbf{F}, d \in \mathbf{D}^{\mathcal{I}} &\implies (a^{\mathcal{I}}, d) \notin f^{\mathcal{I}},
\end{aligned}$$

have to be satisfied formalizing that features are functional, that different atoms denote different elements in the domain, and that atoms are never in the domain of a feature.

The meaning of chains of features $f_1 \dots f_n$, also called **feature paths**, is the composition of functional relations:

$$(d,e) \in f_1 \dots f_n^{\mathcal{I}} \iff \exists d_0, \dots, d_n : d_0 = d \wedge d_n = e \wedge \bigwedge_{i=1}^{n} (d_{i-1}, d_i) \in f_i^{\mathcal{I}}$$

Feature paths will also be denoted by the letters p and q. Using these definitions, the denotation of complex feature terms is given inductively by:

$$\begin{aligned}
(a)^{\mathcal{I}} &= \{a^{\mathcal{I}}\} \\
\top^{\mathcal{I}} &= \mathbf{D}^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset \\
(F \sqcap F)^{\mathcal{I}} &= F^{\mathcal{I}} \cap F^{\mathcal{I}} \\
(F \sqcup F)^{\mathcal{I}} &= F^{\mathcal{I}} \cup F^{\mathcal{I}} \\
(\neg F)^{\mathcal{I}} &= \mathbf{D}^{\mathcal{I}} - F^{\mathcal{I}} \\
(p{:}\, F)^{\mathcal{I}} &= \{d \in \mathbf{D}^{\mathcal{I}} \,|\, \emptyset \neq p^{\mathcal{I}}(d) \subseteq F^{\mathcal{I}}\} \\
(p \downarrow q)^{\mathcal{I}} &= \{d \in \mathbf{D}^{\mathcal{I}} \,|\, p^{\mathcal{I}}(d) = q^{\mathcal{I}}(d) \neq \emptyset\}.
\end{aligned}$$

A feature term F is **satisfiable** iff there exists an interpretation such that $F^{\mathcal{I}} \neq \emptyset$.

If attributive description formalisms contain *intersection* "$\sqcap$" and **complement** "$\neg$," they are called **propositionally complete**. In such formalisms, the notions of satisfiability and subsumption are obviously closely related. More precisely, subsumption and unsatisfiability are linear time reducible to each other (see, e.g., [Nebel and Smolka, 1990]).

3 Regular Languages and Finite State Automata

As mentioned in the previous section, the logical semantics for attributive description formalisms proved to be quite useful in understanding the expressive power of these formalisms. Terminological logics as well as feature logics are obviously subsets of ordinary first-order logic. These subsets, however, were unexplored previously with respect to their computational properties. For instance, it was not known until 1988 whether there are undecidable terminological logics [Schild, 1988] and only in 1989 was it shown that subsumption in KL-ONE [Brachman and Schmolze, 1985; Schmidt-Schauß, 1989] and NIKL [Moser, 1983; Patel-Schneider, 1989b] is undecidable – a point we return to in the next section.

Since in knowledge representation and computational linguistics, efficiency is an important issue, decidability of a formalism is not the only concern. *Tractability*, i.e, solvability in polynomial time, is also relevant. As a matter of fact, Brachman and Levesque [1984] requested that knowledge representation formalisms should always permit polynomial time computations. They started an inquiry concerning the trade-off between expressiveness and tractability of representation formalisms, which led to a number of analyses of different terminological logics [Nebel, 1988; Patel-Schneider, 1989a; Schmidt-Schauß and Smolka, 1991]. However, only recently, terminological logics that are maximally expressive and still tractable have been identified [Donini *et al.*, 1991a] using the *constraint solving* technique introduced in [Schmidt-Schauß and Smolka, 1991].

Another open problem was whether the computational complexity of subsumption for tractable terminological logics is preserved under the introduction of *terminological axioms*. This problem was solved by discovering a correspondence between *nondeterministic finite state automata* and a particular, simple terminological logic. Exploiting complexity results from the theory of finite state automata, it was possible to show that the addition of terminological axioms increases the computational complexity considerably [Nebel, 1990b]. Further, the mentioned correspondence proved to be useful for characterizing the semantics of so-called *terminological cycles* [Baader, 1990; Nebel, 1991].

3.1 Terminological Axioms and the Lexicon

Investigations of the computational complexity of terminological logics are usually based on the semantics given in Sect. 2. They analyze what resources are necessary for checking subsumption between two concept descriptions. In particular, it is assumed that all concept symbols appearing in the descriptions are undefined. In existing systems, however, it is possible to assign a *name* to a concept description and to use this new name in other expressions instead of the original description. This aspect of the use of terminological logics can be straightforwardly formalized by the notion of **terminological axioms**, which have the following form:

$$C \doteq D$$

Usually, it is assumed that sets of such axioms, also called **terminologies** (denoted by T), satisfy two restrictions, namely,

1. a concept symbol C appears at most once on the left hand side of a terminological axiom, and
2. the terminology is cycle-free, i.e., there is a partial order on the set of concepts $\mathbf{C}$ such that for every terminological axiom $C \doteq D$, every concept symbol in D is strictly less than C.

Given such a terminology T, subsumption is relativized to this terminology, written as $D \preceq_T D'$, by considering set-inclusion of concept denotations only in interpretations that are *models* of the terminology. An interpretation $\mathcal{I}$ is a **model** of a set of terminological axioms iff for all axioms $C \doteq D$ the interpretation satisfies $C^{\mathcal{I}} = D^{\mathcal{I}}$.

If the restrictions spelled out above are satisfied, subsumption relative to a terminology can easily be reduced to subsumption over concept descriptions relative to "the empty terminology" by *expanding* all defined concepts by their definitions. However, in the worst case, this can lead to an exponential increase of the size of a concept description [Nebel, 1990b]. Thus, even when subsumption determination for a particular terminological logic is tractable, this does not mean that subsumption determination relative to terminologies is also tractable. On the other hand, all results on the complexity of subsumption seem to have assumed that the reduction from $\preceq_T$ to $\preceq$ can be done in polynomial time – and in applications this reduction did not seem to be a source of computational problems, provided some caching is performed [Lipkis, 1982].

Finally, it turned out that there is indeed a "hidden computational cliff." The minimal terminological language abstract syntax rule

$$D \longrightarrow C \mid D \sqcap D' \mid \forall R\colon D$$

is closely related to nondeterministic finite state automata and, by this, to regular expressions – provided terminological axioms are permitted.

Suppose we are given two **nondeterministic finite state automata** $\mathcal{A}_1, \mathcal{A}_2$ with $\mathcal{A}_i = (\Sigma, \mathcal{Q}^i, \delta^i, q_0^i, \mathcal{F}^i)$, where Σ is the alphabet, $\mathcal{Q}^i$ are the sets of states, where we assume without loss of generality that $\mathcal{Q}^1 \cap \mathcal{Q}^2 = \emptyset$, $\delta^i \subseteq \mathcal{Q}^i \times (\Sigma \cup \{\epsilon\}) \times \mathcal{Q}^i$ are the transition functions, ${q_0}^i \in \mathcal{Q}^i$ are the initial states, and $\mathcal{F}^i \subseteq \mathcal{Q}^i$ are the sets of accepting states. The **language accepted by these automata** is denoted by $\mathcal{L}(\mathcal{A}_i)$. If such automata are *cycle-free*, a cycle-free terminology can be specified such that language inclusion corresponds to subsumption relative to the terminology [Nebel, 1990b]:

Automata	Terminology
$\mathcal{A}_1$, $\mathcal{A}_2$	T
Σ	$\mathbf{R} = \Sigma$
$(\mathcal{Q}^1 \cup \mathcal{Q}^2)$	$\mathbf{C} = (\mathcal{Q}^1 \cup \mathcal{Q}^2) \uplus \{F\}$
$q \in \mathcal{F}^1 \cup \mathcal{F}^2$	$q \doteq \ldots \sqcap F \sqcap \ldots$
ϵ-transition from q to q'	$q \doteq \ldots \sqcap q' \sqcap \ldots$
s-transition from q to q'	$q \doteq \ldots \sqcap \forall s\colon q' \sqcap \ldots$
$\mathcal{L}(\mathcal{A}_1) \supseteq \mathcal{L}(\mathcal{A}_2)$	$q_0^1 \preceq_T q_0^2$

Since inclusion of languages accepted by cycle-free automata is known to be co-NP-complete [Garey and Johnson, 1979], it follows that $\preceq_T$ is co-NP-hard.

Interestingly, this correspondence also works the other way around. Given a terminology and two concepts, we can construct two automata such that subsumption coincides with language inclusion, which gives us co-NP-completeness for $\preceq_T$ in the language considered.

Note that for the proof of this correspondence the set-valued nature of attributes in terminological logics is inessential. The same arguments are valid for functional attributes, which gives us an interesting corollary in the area of unification grammars. Satisfiability of feature terms relative to a lexicon – which is nothing else than a cycle-free terminology for a feature logic [Nebel and Smolka, 1990] – is also NP-hard, even if satisfiability for the underlying feature logic is polynomial. For instance, adopting the ψ-terms introduced in [Aït-Kaci, 1984], for which satisfiability can be decided in quasi-linear time, leads to an NP-complete satisfiability problem if a lexicon is added.

This intractability result does not seem to show up in practical applications very often, however. As a matter of fact, it is not easy to construct a terminology that exhibits exponential time behavior when an efficient algorithm is used that resembles the language inclusion algorithm for finite automata, such as the one described in [Lipkis, 1982]. Nevertheless, it shows us that provable tractability is hardly achievable in the area of attributive description formalisms.

3.2 Terminological Cycles

The correspondence between automata and terminologies not only helped to solve the problem concerning the complexity of subsumption relative to a terminology, but also provides a good tool to analyze so-called **terminological cycles**. Such cycles appear when the second restriction on terminologies mentioned above is dropped. In this case, the definition of a concept refers, either directly or indirectly, to the concept itself. Such constructions present problems because neither the right semantics nor the computational properties are obvious.

Based on the correspondence spelled above, Baader [1990] shows that the three possible styles of semantics, namely, *descriptive, least fixpoint, and greatest fixpoint semantics* [Nebel, 1990a; Nebel, 1991], can be characterized by finite state automata. In particular, the greatest fixpoint semantics has an elegant characterization, because it corresponds to automata isomorphic to the terminology.

Besides confirming the conjecture in [Nebel, 1990b] that subsumption becomes PSPACE-complete for least and greatest fixpoint semantics, this characterization also led directly to sound and complete subsumption algorithms for these cases. In addition, this result gave rise to the idea of extending the expressive power of terminological logics by adding regular expressions over roles [Baader, 1991].

4 Thue Systems

For feature logics, the computational complexity was analyzed quite early. The feature logic described in Sect. 2 without union "$\sqcup$" and complement "$\neg$," which give essentially the ψ-terms mentioned above, was shown to have a quasi-linear satisfiability problem [Aït-Kaci, 1984]. The addition of union or complement leads to NP-completeness, as shown in [Kasper, 1987; Johnson, 1987; Smolka, 1988].

The situation in terminological logics was more problematical because of the variety of possible concept- and role-forming operators. As mentioned above, for a long time it remained an open problem whether there are terminological logics such that subsumption is undecidable. The first undecidability result [Schild, 1988] considered a language containing role complements – which do not have practical relevance. Subsequently, Schmidt-Schauß [1989] proved a small subset of KL-ONE to be undecidable using a reduction from the word problem in *invertible Thue systems* to subsumption. Since this result proved to be quite fruitful for solving other related problems, we will briefly describe the correspondence.

4.1 Feature Agreement and Role-Value-Maps

In the presentation of the logical foundations of attributive descriptions, we mentioned already that other terminological logics than $\mathcal{ALC}$ are conceivable. The reader might have noticed already that *feature-path agreement* $p \downarrow q$ has no counterpart in the presented terminological logic. As a matter of fact, some terminological logics support such an operator, for instance, KL-ONE and NIKL. Let us consider a subset of those formalisms as specified below:

$$D \longrightarrow C \mid D \sqcap D' \mid \forall R\colon D \mid (R_{1,1} \ldots R_{1,m}) \downarrow (R_{2,1} \ldots R_{2,n}),$$

where the denotation of role chains is identical to the denotation of feature chains, i.e., relational composition, and role chains are denoted by P and Q. The agreement of such role chains, often called **role-value-map** is defined by:

$$(P \downarrow Q)^{\mathcal{I}} \quad = \quad \{d \in \mathbf{D}^{\mathcal{I}} \mid P^{\mathcal{I}}(d) = Q^{\mathcal{I}}(d)\}.$$

Such a construct could be used, for instance, to define the concept of a father such that all his children have the same surname as the father:

$$\mathsf{Father} \sqcap (\mathsf{surname}) \downarrow (\mathsf{child\ surname}).$$

Although a very useful construct, it leads unfortunately to undecidability of subsumption. This means that as long as our attributes are functional, subsumption stays decidable (NP-complete for the feature logic considered in this paper or even quasi-linear for the more restricted ψ-terms). If we allow for set-valued attributes, subsumption becomes undecidable. This result follows from a reduction from the word problems for a special class of Thue systems. A **Thue system** $\mathcal{T}$ over an alphabet Σ is a finite set of pairs of words $u_i, v_i \in \Sigma^*$: $\mathcal{T} = \Big\{\{u_i, v_i\}\Big\}$. Such a Thue system defines a binary relation $\overset{\mathcal{T}}{\leftrightarrow}$ on Σ^* by:

$$u \overset{\mathcal{T}}{\leftrightarrow} v \iff \exists w_1, w_2 \in \Sigma^* \; \exists \{u_i, v_i\} \in \mathcal{T}\colon \; u = w_1 u_i w_2 \wedge v = w_1 v_i w_2.$$

The symbol $\overset{\mathcal{T}}{\sim}$ is used to denote the transitive and reflexive closure of $\overset{\mathcal{T}}{\leftrightarrow}$. The **word problem** is the problem to decide $u \overset{\mathcal{T}}{\sim} v$ for given $\mathcal{T}$ and words $u, v \in \Sigma^*$.

An **invertible Thue system** is a Thue system such that for each $s \in \Sigma$ there exists $r \in \Sigma$ such that $sr \overset{\mathcal{T}}{\sim} \epsilon$, where ϵ is the empty word. In other words, the quotient $\mathcal{T}/\overset{\mathcal{T}}{\sim}$ is a group under concatenation. It is known that there exist invertible Thue systems such that the word problem is undecidable [Boone, 1959]. Undecidability of subsumption in the above mentioned terminological logic can now be shown by using the following correspondence:

Invertible Thue system	Terminological logic
Σ	$\mathbf{R} = \Sigma \uplus \{R\}$
$\mathcal{T} = \Big\{\{u_i, v_i\}\Big\}$	$D = \prod_{s\in\Sigma}(R\ \ s) \downarrow (R) \sqcap \prod_i \forall R\colon (u_i \downarrow v_i)$
$u \overset{\mathcal{T}}{\sim} v$	$D \preceq \forall R\colon (u \downarrow v)$

4.2 Arbitrary Axioms

Since, on one hand, agreements of role-chains are a very useful construction, and on the other hand, they lead to undecidability in case of set-valued attributes, it seems to be a good idea to restrict agreements to chains of functional attributes. Indeed, the terminological logic employed in the CLASSIC system [Borgida *et al.*, 1989; Brachman *et al.*, 1991] is based on this insight. Beside ordinary roles also functional attributes are supported and agreements are only permitted on the latter kind of attribute.

While such a move preserves decidability for the terminological logic [Hollunder and Nutt, 1990], it leads to problems if terminologies containing cycles are allowed. Using a similar reduction as above, Smolka [1989] shows that ψ-terms plus cyclic terminological axioms result in undecidability of satisfiability of feature terms w.r.t. terminological axioms. This result can be easily reformulated for the corresponding terminological logics, and it turns out that subsumption for descriptive and greatest fixpoint semantics becomes undecidable [Nebel, 1991]. For this reason, CLASSIC does not support terminological cycles.

Nevertheless, in the CLASSIC system, **implicational rules** are supported. These rules are interpreted procedurally, and they act on a database of objects that are described using concept and role symbols. Given such a rule of the form

$$C(x) \Rightarrow C'(x),$$

any object which the system has classified to belong to the denotation of the concept C will be asserted to belong also to the denotation of C'. If this assertion leads to an inconsistency, i.e., to a situation where an object is interpreted to belong to the denotation of $\perp$, the system signals this contradiction. Although these rules are not identical to axioms, we have the following restriction. A CLASSIC database can be consistently "completed," i.e., allow to be mentioned explicitly all objects that have to exist because of terminological axioms, only if the database plus the terminology have a model. This in turn, however, is equivalent to satisfiability of the terminological axioms plus the implicational rules, which is undecidable in the general case by the above result. This means it is undecidable whether a CLASSIC database has a consistent completion.

4.3 Functional Uncertainty

Another interesting application of the undecidability of the word problem in Thue systems is a reduction from the word problem to satisfiability of feature terms that contain **functional uncertainty** [Kaplan and Maxwell, 1988]. This term-forming operator was invented for the concise description of so-called long-distance dependencies in LFG [Kaplan and Bresnan, 1982]. It has the form

$$\exists(L)F,$$

where L is some finitely represented regular set of words over $\mathbf{F}$. It denotes all individuals $d \in \mathbf{D}^{\mathcal{I}}$ such that there is some feature path $p \in L$ and an element $e \in F^{\mathcal{I}}$, where $e \in p^{\mathcal{I}}(d)$. One can think of $\exists(L)F$ as an infinite union: $p_1\colon F \sqcup p_2\colon F \sqcup \ldots \sqcup p_i\colon F \sqcup \ldots$, where all p_i are elements of L. Formally, the denotation of functional uncertainty is defined as

$$(\exists(L)F)^{\mathcal{I}} \quad = \quad \{d \in \mathbf{D}^{\mathcal{I}} \mid \exists p \in L\colon \emptyset \neq p^{\mathcal{I}}(d) \subseteq F^{\mathcal{I}}\}$$

Decidability of the satisfiability of feature terms containing functional uncertainty has been an open problem. A restricted version of the problem was addressed in [Kaplan and Maxwell, 1988], where a partial solution involving an acyclicity condition is given.

Recalling from Sect. 3.2 the fact that terminological cycles under the greatest fixpoint semantics are closely related to terminological logics that permit regular expressions over roles, one would expect that undecidability would show up again in this case. In fact, if the feature logic specified in Sect. 2 is extended by functional uncertainty, then satisfiability of feature terms is undecidable [Baader *et al.*, 1991].[2] An even stronger result can be shown. Satisfiability of a feature term relative to a set of arbitrary axioms can be reduced to satisfiability of a feature term without axioms [Baader *et al.*, 1991].[3] However, these results strongly depend on the presence of the complement operator. Thus, decidability for functional uncertainty in weaker feature logics – feature logics that are not propositionally complete – is still an open problem.

5 Modal Logics

The most surprising connection between attributive description formalisms and other research areas was recently discovered by Schild [1991]. He showed that a

[2] Note that no terminological axioms are involved here!

[3] A similar result for terminological logics is shown in [Schild, 1991].

large number of possible terminological logics are notational variants of different propositional modal and dynamic logics. Exploiting this correspondence, a number of interesting properties for the latter logics, such as finite model properties, complexity results, and algorithms, can be straightforwardly applied to the corresponding terminological logics. In order to demonstrate the connection between the different fields, we will focus on the correspondence between the terminological logic $\mathcal{ALC}$ [Schmidt-Schauß and Smolka, 1991] introduced in Sect. 2 and the **propositional polymodal logic** $\mathrm{K}_{(m)}$ [Halpern and Moses, 1985].

Given a set of **atomic propositions** $\Psi = \{a, b, c, \ldots\}$, the constants $\top$ and $\bot$ denoting the truth-values `true` and `false`, a set of m operators $K_1, \ldots, K_m$, the set of well-formed $\mathrm{K}_{(m)}$-formulas (denoted by ϕ) is defined by

$$\phi \longrightarrow a \mid \top \mid \bot \mid \phi \wedge \phi' \mid \phi \vee \phi' \mid \neg\phi \mid K_i\phi.$$

Satisfiability of such formulas is defined with respect to **Kripke structures**

$$M = (S, \pi, \kappa_1, \ldots, \kappa_m),$$

where S is a set of states, $\pi(s)$ is a truth-assignment for all atomic propositions in Ψ at the state $s \in S$, and $\kappa_i \subseteq S \times S$ are the accessibility relations. A formula ϕ is said to be **satisfied at a world** (M, s), written $(M, s) \models \phi$, under the following conditions:

$$\begin{array}{lcl}
(M,s) \models a & \iff & \pi(s)(a) = \mathtt{true} \\
(M,s) \models \top & & \\
(M,s) \not\models \bot & & \\
(M,s) \models \phi \wedge \phi' & \iff & (M,s) \models \phi \text{ and } (M,s) \models \phi' \\
(M,s) \models \phi \vee \phi' & \iff & (M,s) \models \phi \text{ or } (M,s) \models \phi' \\
(M,s) \models \neg\phi & \iff & (M,s) \not\models \phi \\
(M,s) \models K_i\phi & \iff & \forall t \in \kappa_i(s)\colon (M,t) \models \phi
\end{array}$$

A $\mathrm{K}_{(m)}$-formula ϕ is **satisfiable**, iff there exists a world (M, s) that satisfies ϕ. ϕ is **valid**, written $\models \phi$, iff all worlds satisfy ϕ.

This notion of satisfiability is obviously closely related to satisfiability of $\mathcal{ALC}$-concepts. Indeed, there is a one-to-one correspondence between $\mathcal{ALC}$ and $\mathrm{K}_{(m)}$, as can be seen from the following table:

Polymodal logic $K_{(m)}$	Terminological logic $\mathcal{ALC}$
Ψ	$\mathbf{C} = \Psi$
$\{1, \ldots, m\}$	$\mathbf{R} = \{R_1, \ldots, R_m\}$
$\top$	$\top$
$\bot$	$\bot$
$\phi \wedge \phi'$	$\phi \sqcap \phi'$
$\phi \vee \phi'$	$\phi \sqcup \phi'$
$\neg\phi$	$\neg\phi$
$K_i\phi$	$\forall R_i \colon \phi$
$\neg K_i \neg\phi$	$\exists R_i \colon \neg\phi$
ϕ satisfiable	ϕ is a satisfiable $\mathcal{ALC}$-concept
$\models \neg\phi \vee \phi'$	$\phi \preceq \phi'$

PSPACE-completeness of subsumption in $\mathcal{ALC}$ follows immediately, because satisfiability in $K_{(m)}$ is known to be PSPACE-complete [Halpern and Moses, 1985]. Hence, we have an alternative proof of the complexity of subsumption to the one presented in [Schmidt-Schauß and Smolka, 1991]. The most interesting aspect of this close correspondence is that it also works for other variants of propositional modal and dynamic logics [Schild, 1991], giving us a large number of complexity results and algorithms for free. This correspondence also applies to feature logics. In this context, deterministic dynamic logics are the right kind of logics to establish the correspondence. However, although these correspondences can be used to solve a number of open problems, there are aspects which have not been considered in modal and dynamic logics. For instance, agreements of feature paths do not have a counterpart in modal or dynamic logics.

6 Conclusion

We have demonstrated that the study of formal properties of attributive description formalisms, which jointly characterize terminological and feature logics, is quite closely connected to other areas of formal research. In particular, we have shown how the theory of finite state automata helps in solving some open problems in terminological logics, and how the word problem for Thue systems is applied to a number of problems to prove undecidability. Finally, we have examined the close correspondence between attributive description formalisms on one side and modal and dynamic logics on the other.

Interestingly, the study of attributive description formalisms is not only a sink for results in other areas, but also provides insights which can be applied elsewhere. For instance, complex object data models, such as O_2 [Lécluse *et al.*, 1989], are closely related to attributive description formalisms, so that the techniques are applicable. Such an application reveals that the subtype-inference algorithm specified in [Lécluse *et al.*, 1989] is incomplete, and that the subtype-inference problem is PSPACE-complete [Bergamaschi and Nebel, 1990]. Further, the study of sublanguages of $\mathcal{ALC}$ [Donini *et al.*, 1991b; Donini *et al.*, 1991a] can be directly applied to sublogics of $\mathrm{K}_{(m)}$. For example, if only negation of propositional atoms is allowed and there is no disjunction, then satisfiability of a $\mathrm{K}_{(m)}$-formula is co-NP-complete. Finally, the undecidability result for subsumption constraints in feature logics yields the undecidability of semi-unification over rational trees [Dörre and Rounds, 1990].

Acknowledgement

We would like to thank Franz Baader and Werner Nutt for a number of helpful comments on an earlier version of this paper.

References

[Aït-Kaci, 1984] Hassan Aït-Kaci. *A Lattice-Theoretic Approach to Computations Based on a Calculus of Partially Ordered Type Structures.* PhD thesis, University of Pennsylvania, Philadelphia, PA, 1984.

[Baader *et al.*, 1991] Franz Baader, Hans-Jürgen Bürckert, Bernhard Nebel, Werner Nutt, and Gert Smolka. On the expressivity of feature logics with negation, functional uncertainty, and sort equations. DFKI Report RR-91-01, German Research Center for Artificial Intelligence (DFKI), Saarbrücken, January 1991.

[Baader, 1990] Franz Baader. Terminological cycles in KL-ONE-based knowledge representation languages. In *Proceedings of the 8th National Conference of the American Association for Artificial Intelligence*, pages 621–626, Boston, MA, 1990.

[Baader, 1991] Franz Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1991. To appear.

[Bergamaschi and Nebel, 1990] Sonia Bergamaschi and Bernhard Nebel. Theoretical foundations of complex object data models. Technical Report 74, CIOC-CNR, Bologna, Italy, December 1990.

[Boone, 1959] W. W. Boone. The word problem. *Ann. of Math.*, 70(2):207–265, 1959.

[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, OR, June 1989.

[Bouma *et al.*, 1988] Gosse Bouma, Esther König, and Hans Uszkoreit. A flexible graph-unification formalism and its application to natural-language processing. *IBM Journal of Research and Development*, 32(2):170–184, March 1988.

[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, TX, 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.*, 1991] Ronald J. Brachman, Deborah L. McGuinness, Peter F. Patel-Schneider, Lori Alperin Resnick, and Alexander Borgida. Living with CLASSIC: When and how to use a KL-ONE-like language. In J. Sowa, editor, *Principles of Semantic Networks*. Morgan Kaufmann, San Mateo, CA, 1991.

[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, NY, 1979.

[Donini *et al.*, 1991a] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. Tractable concept languages. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, August 1991. To appear.

[Donini *et al.*, 1991b] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Werner Nutt. The complexity of concept languages. In J. A. Allen, R. Fikes, and E. Sandewall, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 2nd International Conference*, pages 151–162, Cambridge, MA, April 1991.

[Dörre and Rounds, 1990] Jochen Dörre and William C. Rounds. On subsumption and semi-unification in feature algebras. In *Proceedings of the 5th IEEE Symposium on Logic in Computer Science*, pages 301–310, Philadelphia, PA, 1990.

[Dörre and Seiffert, 1991] Jochen Dörre and Roland Seiffert. STUF – formalism and workbench. In O. Herzog and C. Rollinger, editors, *Textunderstanding in LILOG: Integrating Computational Linguistics and Artificial Intelligence*. Springer-Verlag, Berlin, Heidelberg, New York, 1991. To appear.

[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, CA, 1979.

[Halpern and Moses, 1985] Josef Y. Halpern and Yoram Moses. A guide to the modal logics of knowledge and belief: Preliminary draft. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 480–490, Los Angeles, CA, August 1985.

[Hollunder and Nutt, 1990] Bernhard Hollunder and Werner Nutt. Subsumption algorithms for concept languages. DFKI Report RR-90-04, German Research Center for Artificial Intelligence (DFKI), Kaiserslautern, 1990.

[Johnson, 1987] Mark Johnson. *Attribute-Value Logic and the Theory of Grammar.* CSLI Lecture Notes 16. Center for the Study of Language and Information, Stanford University, Stanford, CA, 1987.

[Kaplan and Bresnan, 1982] Ronald M. Kaplan and Joan Bresnan. Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–381. MIT Press, Cambridge, MA, 1982.

[Kaplan and Maxwell, 1988] Ronald M. Kaplan and John T. Maxwell, III. An algorithm for functional uncertainty. In *Proceedings of the 12th International Conference on Computational Linguistics*, pages 297–302, Budapest, Hungary, 1988.

[Kasper and Rounds, 1986] Robert T. Kasper and William C. Rounds. A logical semantics for feature structures. In *Proceedings of the 24th Annual Meeting of the ACL*, pages 257–265, New York, NY, 1986.

[Kasper, 1987] Robert T. Kasper. *Feature Structures: A Logical Theory with Applications to Language Analysis.* PhD thesis, University of Michigan, Ann Arbor, MI, 1987.

[Kay, 1979] Martin Kay. Functional grammar. In *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistics Society*, Berkeley, CA, 1979. Berkeley Linguistics Society.

[Kay, 1985] Martin Kay. Parsing in functional unification grammars. In D. Dowty and L. Karttunen, editors, *Natural Language Parsing.* Cambridge University Press, Cambridge, UK, 1985.

[Lécluse *et al.*, 1989] Christophe Lécluse, Philippe Richard, and Fernando Velez. Modeling complex structures in object-oriented databases. In *Proceedings of the 8th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database-Systems*, pages 360–367, March 1989.

[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, MA, 1982. The proceedings have been published as BBN Report No. 4842 and Fairchild Technical Report No. 618.

[Moser, 1983] M. G. Moser. An overview of NIKL, the new implementation of KL-ONE. In *Research in Knowledge Representation and Natural Language Understanding, BBN Report No. 5421*, pages 7–26. Bolt, Beranek, and Newman Inc., Cambridge, MA, 1983.

[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*, volume 418 of *Lecture Notes in Artificial Intelligence*, pages 112–139. Springer-Verlag, Berlin, Heidelberg, New York, 1990.

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

[Nebel, 1990a] Bernhard Nebel. *Reasoning and Revision in Hybrid Representation Systems*, volume 422 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, Heidelberg, New York, 1990.

[Nebel, 1990b] Bernhard Nebel. Terminological reasoning is inherently intractable. *Artificial Intelligence*, 43:235–249, 1990. Also available as IWBS Report 82, IWBS, IBM Germany, Stuttgart, October 1989.

[Nebel, 1991] Bernhard Nebel. Terminological cycles: Semantics and computational properties. In J. Sowa, editor, *Principles of Semantic Networks*. Morgan Kaufmann, San Mateo, CA, 1991.

[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.

[Pletat and von Luck, 1990] Udo Pletat and Kai von Luck. Knowledge Representation in LILOG. In Karl-Hans Bläsius, Uli Hedtstück, and Claus Rollinger, editors, *Sorts and Types in Artificial Intelligence*, volume 418 of *Lecture Notes in Artificial Intelligence*, pages 140–164. Springer-Verlag, Berlin, Heidelberg, New York, 1990.

[Pletat, 1991] Udo Pletat. The knowledge representation language L-LILOG. In O. Herzog and C. Rollinger, editors, *Textunderstanding in LILOG: Integrating Computational Linguistics and Artificial Intelligence*. Springer-Verlag, Berlin, Heidelberg, New York, 1991. To appear.

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

[Schild, 1991] Klaus Schild. A correspondence theory for terminological logics: Preliminary report. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, Sydney, Australia, 1991. To appear.

[Schmidt-Schauß and Smolka, 1991] Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 48:1–26, 1991. Also available as IWBS Report 68, IWBS,IBM Germany, Stuttgart, 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, *Principles of Knowledge Representation and Reasoning: Proceedings of the 1st International Conference*, pages 421–431, Toronto, ON, May 1989.

[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, MA, 1983.

[Shieber, 1986] Stuart M. Shieber. *An Introduction to Unification-Based Approaches to Grammar*. CSLI Lecture Notes 4. Center for the Study of Language and Information, Stanford University, Stanford, CA, 1986.

[Smolka, 1988] Gert Smolka. A feature logic with subsorts. LILOG Report 33, IWBS, IBM Germany, Stuttgart, May 1988. To appear in: J. Wedekind, C. Rohrer (eds.), Unification in Grammar. MIT Press, 1991.

[Smolka, 1989] Gert Smolka. Feature constraint logics for unification grammars. IWBS Report 93, IWBS, IBM Germany, Stuttgart, November 1989. To appear in the Journal of Logic Programming.