To appear in *Journal of Logic, Language, and Information*, 1993.

On the Expressivity of Feature Logics with Negation, Functional Uncertainty, and Sort Equations

Franz Baader, Hans-Jürgen Bürckert, Bernhard Nebel, Werner Nutt and Gert Smolka
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany

Abstract. Feature logics are the logical basis for so-called unification grammars studied in computational linguistics. We investigate the expressivity of feature terms with negation and the functional uncertainty construct needed for the description of long-distance dependencies and obtain the following results: satisfiability of feature terms is undecidable, sort equations can be internalized, consistency of sort equations is decidable if there is at least one atom, and consistency of sort equations is undecidable if there is no atom.

Key words: Feature Logic, Functional Uncertainty, Sort Equations, Undecidability

1 Introduction

Feature constraint grammars, also known as unification grammars, have become the predominant family of declarative grammar formalisms in Computational Linguistics (Kay, 1979; Kaplan and Bresnan, 1982; Shieber *et al.*, 1983; Shieber, 1986; Pollard and Sag, 1987). The common assumption of these formalisms is that linguistic objects can be described by means of their features, which are functional attributes. Figure 1, for instance, shows the description of a linguistic object that may represent the sentence "John sings a song". The features appear as edges of the graph. The terminal nodes are atoms representing primitive linguistic objects.

Kasper and Rounds (Kasper and Rounds, 1986; Rounds and Kasper, 1986) were the first to capture the relation between feature descriptions and linguistics objects in terms of a logic. Subsequently, Johnson (1988) and Smolka (1988; 1992) realized that feature logics can be modeled straightforwardly in Predicate Logic.[1] In this approach, which underlies the present paper, a domain of linguistic objects is called a feature algebra and is simply a structure that interprets atoms as pairwise distinct individuals and features as unary partial functions that are undefined on atoms. In addition, one can have sorts, which are interpreted as sets of individuals.

One popular syntax for feature descriptions are so-called feature terms (Kasper and Rounds, 1986; Rounds and Kasper, 1986; Smolka, 1992), which are expressions denoting sets in feature algebras. The basic feature term forms are given by

$$S \longrightarrow a \mid A \mid p{:}\, S \mid p{\downarrow}q \mid S \sqcap S' \mid S \sqcup S' \mid \neg S,$$

where a stands for atoms, A stands for sorts, and p and q stand for words over features. Given a feature algebra, a denotes the singleton consisting of the atom

[1] An alternative view on these formalisms is provided by the "modal perspective" (Blackburn and Spaan, 1992).

Journal of Logic, Language, and Information **0:** 1–1, 1995.

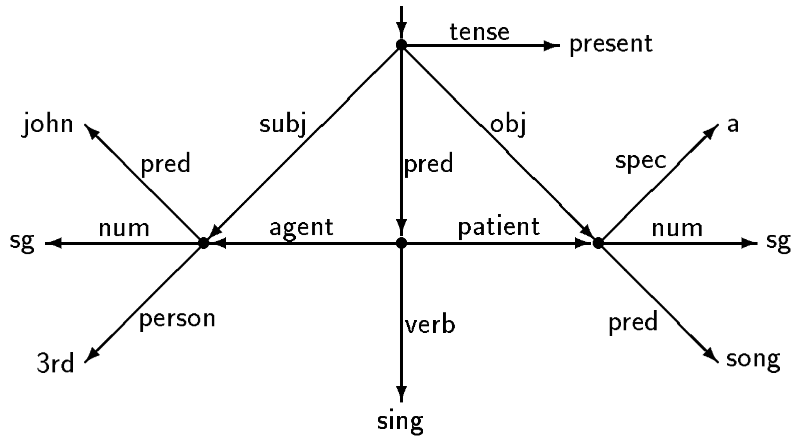

Fig. 1. A feature graph.

a, $p{:}\,S$ denotes the inverse image of S under p (where p is interpreted as unary partial function obtained as the composition of its features), $p \downarrow q$ denotes the set of all individuals for which p and q are defined and agree, $S \sqcap S'$ denotes the intersection of S and S', $S \sqcup S'$ denotes the union of S and S', and $\neg S$ denotes the complement of S. For applications it is important to decide whether a feature term is satisfiable, that is, whether it denotes a nonempty set in some feature algebra. The satisfiability problem for feature terms as given above is NP-complete (Smolka, 1992).

In this paper we investigate the expressivity of an additional feature term form that is known as functional uncertainty (Kaplan and Maxwell, 1989a; Kaplan and Maxwell, 1989b) and was invented for the convenient description of so-called long-distance dependencies in the grammar formalism LFG (Kaplan and Bresnan, 1982). It takes the form

$$\exists L(S),$$

where L is a finite description of a regular set of words over features and S is a feature term. A feature term $\exists L(S)$ denotes the set of all individuals d such that there exists a word $p \in L$ such that d is in the inverse image of S under p. One can think of $\exists L(S)$ as the possibly infinite union

$$p_1{:}\,S \sqcup p_2{:}\,S \sqcup p_3{:}\,S \sqcup \cdots,$$

where $p_1, p_2, p_3, \ldots$ are the words in L. Note that the form $p{:}\,S$ can be expressed with $\exists L(S)$ if one takes for L the singleton consisting of the word p.

So far, the computational properties of functional uncertainty are known only for one restricted special case. Kaplan and Maxwell (1989a) have shown that

$$
\begin{aligned}
\mathsf{S} &\doteq (\mathsf{subj}\colon \mathsf{NP}) \sqcap \mathsf{VP} \\
\mathsf{NP} &\doteq (\mathsf{D} \sqcap \mathsf{N}) \sqcup \mathsf{Name} \\
\mathsf{VP} &\doteq \mathsf{V} \sqcap (\mathsf{obj}\colon \mathsf{NP}) \\
\mathsf{D} &\doteq (\mathsf{num}\colon \mathsf{sg}) \sqcap (\mathsf{spec}\colon \mathsf{a}) \\
\mathsf{N} &\doteq (\mathsf{num}\colon \mathsf{sg}) \sqcap (\mathsf{pred}\colon \mathsf{song}) \\
\mathsf{Name} &\doteq (\mathsf{num}\colon \mathsf{sg}) \sqcap (\mathsf{person}\colon \mathsf{3rd}) \sqcap (\mathsf{pred}\colon \mathsf{john}) \\
\mathsf{V} &\doteq (\mathsf{subj\ num}\colon \mathsf{sg}) \sqcap (\mathsf{subj\ person}\colon \mathsf{3rd}) \sqcap (\mathsf{pred\ verb}\colon \mathsf{sing}) \sqcap \\
&\quad\ (\mathsf{tense}\colon \mathsf{present}) \sqcap (\mathsf{pred\ agent}{\downarrow}\mathsf{subj}) \sqcap (\mathsf{pred\ patient}{\downarrow}\mathsf{object})
\end{aligned}
$$

Fig. 2. A simple grammar.

satisfiability of feature terms built with the forms a, $p\downarrow q$, $S \sqcap S'$, and $\exists L(S)$ is decidable, provided a certain acyclicity condition is met. The decidability for more general cases has been an open problem, however.

We show below that satisfiability is undecidable in the general case, even if there are no atoms. However, our result depends crucially on the presence of the negation $\neg S$. Hence, our result presents only an upper bound for the more restricted class of feature terms actually used in practice. In particular, the problem is still open for feature terms that are built using only the forms a, $p\downarrow q$, $S \sqcap S'$, and $\exists L(S)$ without any additional conditions such as the acyclicity condition mentioned above.

In order to characterize the expressivity of functional uncertainty, we relate it to another construct often used in feature constraint grammars, namely, sort equations. A sort equation is a pair $S \doteq S'$ consisting of two feature terms. A feature algebra is a model of a set of sort equations if for every equation both sides denote the same set.

Grammar rules in Functional Unification Grammar (Kay, 1979) and the more recent HPSG (Pollard and Sag, 1987) are stated by means of sort equations. Figure 2 shows a simple grammar in this style (sorts start with capital letters), which generates the single sentence "John sings a song", provided the right assumptions on word order are made. The basic idea is that in a model of the grammar the elements of a sort are the linguistic objects of the syntactic category expressed by the sort. Note that the graph in Figure 1 describes an element of the sort S in some model of the grammar in Figure 2.

There is a surprising connection between functional uncertainty and sort equations. We will exhibit an algorithm that, given a finite set $\mathcal{E}$ of sort equations and a feature term S, produces finitely many feature terms $S_1, \ldots, S_n$ such that S is satisfiable in a model of $\mathcal{E}$ if and only if $S_1, \ldots, S_n$ are satisfiable in some (arbitrary) feature algebra. This result says that, in the presence of functional uncertainty and negation, sort equations can be internalized and thus do not yield additional expressivity with respect to satisfiability. Since it is known that satisfiability with respect to sort equations is undecidable (Smolka, 1992), this result immediately implies that satisfiability of feature terms with functional uncertainty and negation is undecidable.

As an interesting byproduct of the internalization result for sort equations, we will show that, somewhat surprisingly, it is decidable whether a finite set of sort equations has a model, provided there is at least one atom. However, if we do not assume atoms, the consistency of sort equations becomes undecidable, even if we disallow feature terms with functional uncertainty.

The paper is organized as follows. Section 2 defines feature algebras, feature terms and sort equations and states basic properties. Section 3 shows that to decide satisfiability of feature terms it suffices to consider only the roots of rooted feature algebras, an auxiliary result on which the rest of the paper depends. Section 4 shows that consistency of sort equations is decidable if there is at least one atom while it is undecidable if there is no atom (our first main result). Section 5 shows how sort equations can be expressed with functional uncertainty (our second main result). Section 6 shows that satisfiability of feature terms with functional uncertainty and negation is undecidable (our third main result). Section 7 concludes.

2 Feature Algebras and Feature Terms

We assume three pairwise disjoint, possibly empty sets of symbols: **atoms** (denoted by a, b, c), **sorts** (denoted by A, B, C), and **features** (denoted by f, g, h). In the following, let $\mathbf{A}$ denote the set of all atoms, $\mathbf{S}$ the set of all sorts, and $\mathbf{F}$ the set of all features. We assume that there is at least one symbol, that is, $\mathbf{A} \cup \mathbf{S} \cup \mathbf{F} \neq \emptyset$.

A **feature algebra** is a pair $(\mathbf{D}^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consisting of a nonempty set $\mathbf{D}^{\mathcal{I}}$ (the **domain** of $\mathcal{I}$) and an **interpretation function** $\cdot^{\mathcal{I}}$ assigning to every atom a an element $a^{\mathcal{I}} \in \mathbf{D}^{\mathcal{I}}$, to every sort A a subset $A^{\mathcal{I}} \subseteq \mathbf{D}^{\mathcal{I}}$, and to every feature f a set of ordered pairs $f^{\mathcal{I}} \subseteq \mathbf{D}^{\mathcal{I}} \times \mathbf{D}^{\mathcal{I}}$ such that the following conditions are satisfied:

1. if (d, e) and (d, e') are in $f^{\mathcal{I}}$, then $e = e'$ *(features are functional)*
2. if $a \neq b$, then $a^{\mathcal{I}} \neq b^{\mathcal{I}}$ *(unique name assumption)*
3. if f is a feature and a is an atom, then there exists no $d \in \mathbf{D}^{\mathcal{I}}$ such that $(a^{\mathcal{I}}, d) \in f^{\mathcal{I}}$ *(atoms are primitive)*.

Note that we can see features equivalently either as functional binary relations or as unary partial functions. In place of $(d, e) \in f^{\mathcal{I}}$ we shall equivalently use the notation $e = df^{\mathcal{I}}$, which means that the partial function $f^{\mathcal{I}}$, if applied to d, yields the value e. If there exists no e such that $(d, e) \in f^{\mathcal{I}}$ we say that $df^{\mathcal{I}}$ is undefined. We use suffix notation for application of partial functions because we want to write composition of binary relations and partial functions from left to right, that is, $f^{\mathcal{I}} g^{\mathcal{I}}$ will mean apply first $f^{\mathcal{I}}$ and then $g^{\mathcal{I}}$.

A **path** is a word in $\mathbf{F}^*$, that is, a finite, possibly empty sequence of features. We shall use the letters p, q, and r for paths. Let $\mathcal{I}$ be a feature algebra and $p = f_1 \cdots f_n$ ($n \geq 0$) be a path. The empty path ε is interpreted as the identity on $\mathbf{D}^{\mathcal{I}}$. For $n \geq 1$, p is interpreted as the functional binary relation $p^{\mathcal{I}}$ which is obtained by composition of the functional binary relations $f_1^I, \ldots, f_n^I$, that is,

$$(d,e) \in (f_1 \cdots f_n)^{\mathcal{I}} \iff \exists d_0, \ldots, d_n\colon\ d = d_0 \wedge d_n = e \\ \wedge\ (d_0, d_1) \in f_1^{\mathcal{I}} \wedge \ldots \wedge (d_{n-1}, d_n) \in f_n^{\mathcal{I}}.$$

As for single features, we shall often write $e = d(f_1 \cdots f_n)^{\mathcal{I}}$ instead of $(d,e) \in (f_1 \cdots f_n)^{\mathcal{I}}$, and shall say $d(f_1 \cdots f_n)^{\mathcal{I}}$ is undefined if there exists no such e.

Regular sets of paths can be specified by finite means, for instance, by regular expressions over the alphabet of all features. The letter L will always denote a finite description of a regular set of paths, and we write $p \in L$ if the path p is in the regular set specified by L. As usual, we shall take $\emptyset$ as description of the empty set of paths, and for a path, p as description of the singleton $\{p\}$.

Feature terms are descriptions that denote sets in feature algebras. Here is the abstract syntax of **feature terms**:

$S, T \longrightarrow$	a	*atom*
	A	*sort*
	$p\colon S$	*selection*
	$p{\downarrow}q$	*agreement*
	$\bot$	*bottom*
	$\top$	*top*
	$S \sqcap T$	*intersection*
	$S \sqcup T$	*union*
	$\neg S$	*negation*
	$S - T$	*difference*
	$\exists L(S)$	*existential path quantification*
	$\forall L(S)$	*universal path quantification.*

Because of the symmetry with universal path quantification we prefer to call the functional uncertainty construct existential path quantification. We will see that universal path quantification can be expressed with existential path quantification and negation.

It is important to note that our feature term language is parameterized with respect to the three alphabets of atoms, sorts and features, and that each of these alphabets may be empty.

Given a feature algebra $\mathcal{I}$, the **denotation** $S^{\mathcal{I}}$ of a feature term S in $\mathcal{I}$ is a subset of $\mathbf{D}^{\mathcal{I}}$ defined inductively as follows:

$$(a)^{\mathcal{I}} = \{a^{\mathcal{I}}\}$$
$$(p\colon S)^{\mathcal{I}} = \{d \in \mathbf{D}^{\mathcal{I}} \mid \exists e \in S^{\mathcal{I}}\colon\ (d,e) \in p^{\mathcal{I}}\}$$

$$
\begin{aligned}
(p{\downarrow}q)^{\mathcal{I}} &= \{d \in \mathbf{D}^{\mathcal{I}} \mid \exists e \in \mathbf{D}^{\mathcal{I}}\colon\ (d,e) \in p^{\mathcal{I}} \cap q^{\mathcal{I}}\} \\
\bot^{\mathcal{I}} &= \emptyset \\
\top^{\mathcal{I}} &= \mathbf{D}^{\mathcal{I}} \\
(S \sqcap T)^{\mathcal{I}} &= S^{\mathcal{I}} \cap T^{\mathcal{I}} \\
(S \sqcup T)^{\mathcal{I}} &= S^{\mathcal{I}} \cup T^{\mathcal{I}} \\
(\neg S)^{\mathcal{I}} &= \mathbf{D}^{\mathcal{I}} - S^{\mathcal{I}} \\
(S - T)^{\mathcal{I}} &= S^{\mathcal{I}} - T^{\mathcal{I}} \\
(\exists L(S))^{\mathcal{I}} &= \{d \in \mathbf{D}^{\mathcal{I}} \mid \exists p \in L\ \exists (d,e) \in p^{\mathcal{I}}\colon\ e \in S^{\mathcal{I}}\} \\
(\forall L(S))^{\mathcal{I}} &= \{d \in \mathbf{D}^{\mathcal{I}} \mid \forall p \in L\ \forall (d,e) \in p^{\mathcal{I}}\colon\ e \in S^{\mathcal{I}}\}.
\end{aligned}
$$

Note that if a feature term S is a sort, $S^{\mathcal{I}}$ is given directly by the feature algebra $\mathcal{I}$.

Two feature terms S and T are **equivalent** (written $S \sim T$) if $S^{\mathcal{I}} = T^{\mathcal{I}}$ for every feature algebra $\mathcal{I}$.

Many of the introduced feature term forms are redundant. By rewriting with the equivalences

$$
\begin{aligned}
p\colon S &\sim \exists p(S) \\
\forall L(S) &\sim \neg\exists L(\neg S) \\
\bot &\sim \exists\emptyset(S) \qquad \text{(where } S \text{ is an arbitrary feature term)} \\
\top &\sim \forall\emptyset(S) \qquad \text{(where } S \text{ is an arbitrary feature term)} \\
S \sqcup T &\sim \neg(\neg S \sqcap \neg T) \\
S - T &\sim S \sqcap \neg T
\end{aligned}
$$

the forms appearing as the left hand sides can be eliminated. Obviously, the equivalences for $\top$ and $\bot$ can only be used to eliminate these forms if there exists a feature term S containing neither $\top$ nor $\bot$. This is in fact the case since we assumed $\mathbf{A} \cup \mathbf{S} \cup \mathbf{F}$ to be nonempty.

Proposition 2.1 *For every feature term one can compute in linear time an equivalent feature term containing only the forms a, A, $p \downarrow q$, $\exists L(S)$, $S \sqcap T$, and $\neg S$.*

A feature term S is called **satisfiable** if there exists a feature algebra $\mathcal{I}$ such that $S^{\mathcal{I}} \neq \emptyset$. Due to the presence of negation, unsatisfiability and equivalence of feature terms are linear-time reducible to each other:

$$
\begin{aligned}
S \text{ unsatisfiable} &\iff S \sim \bot \\
S \sim T &\iff (S - T) \sqcup (T - S) \text{ unsatisfiable.}
\end{aligned}
$$

Until now we have defined satisfiability, equivalence and inclusion of feature terms with respect to all feature algebras. One can also use axioms to specify classes of feature algebras with respect to which satisfiability, equivalence and inclusion should be considered. As axioms we use so-called **sort equations** which take the form $S \doteq T$, where S and T are feature terms. A feature algebra $\mathcal{I}$ **satisfies** a sort equation $S \doteq T$ iff $S^{\mathcal{I}} = T^{\mathcal{I}}$. A feature algebra $\mathcal{I}$ is a **model** of a set $\mathcal{E}$ of sort equations iff it satisfies every sort equation in $\mathcal{E}$. A set of sort equations is called **consistent** iff it has at least one model. We say that a feature term S is **satisfiable** w.r.t. a set $\mathcal{E}$ of sort equations iff there exists a model $\mathcal{I}$ of $\mathcal{E}$ such that $\mathcal{S}^{\mathcal{I}} \neq \emptyset$. As for the case without sort equations, unsatisfiability, inclusion, and equivalence of feature terms w.r.t. a set of sort equations are linear-time reducible to each other.

Finitely many sort equations can always be equivalently expressed by a single sort equation of the form $S \doteq \bot$. In fact, a feature algebra $\mathcal{I}$ satisfies a sort equation $S \doteq T$ iff it satisfies $(S - T) \sqcup (T - S) \doteq \bot$; and $\mathcal{I}$ satisfies the sort equations $S_1 \doteq \bot$, ..., $S_n \doteq \bot$ iff it satisfies $S_1 \sqcup \ldots \sqcup S_n \doteq \bot$.

3 Rooted Feature Algebras

The purpose of this section is to define the notion of a "rooted feature algebra," and to derive some results for rooted feature algebras which will be useful in the following two sections. This notion is very similar to the notion of a "generated submodel" as introduced for modal and multimodal logic (see e.g. (Goldblatt, 1987)). However, because of the presence of atoms in our formalism, the definition of rooted feature algebras is more complex.

Let S be a satisfiable feature term, and let the feature algebra $\mathcal{I}$ together with the element $d \in \mathbf{D}^{\mathcal{I}}$ be a witness for the satisfiability of S, that is, let $d \in S^{\mathcal{I}}$. Then $\mathbf{D}^{\mathcal{I}}$ may contain "unreachable" elements that are not needed to verify $d \in S^{\mathcal{I}}$. As a consequence of the main theorem of this section we will see that, to decide satisfiability of feature terms, it suffices to consider only the roots of rooted feature algebras. This fact will be used in Section 5 to show that sort equations can be internalized, i.e., simulated by feature terms with path quantification. As a second consequence of the main theorem, one obtains a result on the behavior of atoms with respect to sorts and feature terms, which is used in Section 4 to show that consistency of sort equations is decidable, provided that one has at least one atom.

Let $\mathcal{I}$ be a feature algebra and let d be an element of $\mathbf{D}^{\mathcal{I}}$. We define

$$gen(d) := \{e \in \mathbf{D}^{\mathcal{I}} \mid \text{ there exists a path } p \text{ with } dp^{\mathcal{I}} = e\}$$

and say that an element of $gen(d)$ is generated by d. Obviously, $d \in gen(d)$, and $e \in gen(d)$ implies that $gen(e) \subseteq gen(d)$.

Our intention is now to restrict the domains of feature algebras to such sets $gen(d)$. However, we must keep in mind that atoms must always be interpreted somehow. Thus, if some elements of $\mathbf{A}^{\mathcal{I}} = \{a^{\mathcal{I}} \mid a \in \mathbf{A}\}$ are not contained in $gen(d)$ we cannot really restrict the domain to $gen(d)$, but only to $gen(d) \cup \mathbf{A}^{\mathcal{I}}$.

We say that a feature algebra $\mathcal{I}$ is **rooted** iff there exists $d \in \mathbf{D}^{\mathcal{I}}$ such that $\mathbf{D}^{\mathcal{I}} = gen(d) \cup \mathbf{A}^{\mathcal{I}}$. In this case, d is called a **root** of $\mathcal{I}$.

In order to show that it is sufficient to consider such rooted feature algebras when interested in satisfiability of feature terms, we need the following weak notion of restriction of a feature algebra. Let $\mathcal{I}$ be a feature algebra and let M be a subset of $\mathbf{D}^{\mathcal{I}}$. Then a feature algebra $\mathcal{J}$ is called a **quasi-restriction** of $\mathcal{I}$ to the subset M iff it satisfies the following properties:

1. $\mathbf{D}^{\mathcal{J}} = M \cup \mathbf{A}^{\mathcal{I}}$,
2. $a^{\mathcal{J}} = a^{\mathcal{I}}$ for all atoms a,
3. $A^{\mathcal{J}} \cap M = A^{\mathcal{I}} \cap M$ for all sorts A, and
4. $f^{\mathcal{J}} = f^{\mathcal{I}} \cap M \times M$ for all features f.

For a given feature algebra $\mathcal{I}$ and a subset M of $\mathbf{D}^{\mathcal{I}}$ there may exist more than one quasi-restriction of $\mathcal{I}$ to M. These quasi-restrictions may differ in the behavior of elements of $\mathbf{A}^{\mathcal{I}} - M$ with respect to sorts. Nevertheless, we shall often use the name $\mathcal{I}|_M$ for such a quasi-restriction. We call $\mathcal{J}$ quasi-restriction of $\mathcal{I}$ and not restriction because usually one has that restrictions are unique. However, defining the notion "quasi-restriction to a set M" in this non-unique way is necessary for the proof of Corollary 3.4, which in turn is important for the proofs of Proposition 4.1 and Lemma 5.1.

Lemma 3.1 *Let $\mathcal{I}|_M$ be a quasi-restriction of $\mathcal{I}$ to the subset M of $\mathbf{D}^{\mathcal{I}}$. For all feature terms S and all elements d of $\mathbf{D}^{\mathcal{I}}$ satisfying $gen(d) \subseteq M$ we have*

$$d \in S^{\mathcal{I}} \iff d \in S^{\mathcal{I}|_M}.$$

Proof. The lemma is proved by induction on the structure of S. Without loss of generality we may assume that S contains only the forms a, A, $p{\downarrow}q$, $\exists L(T)$, $T_1 \sqcap T_2$, $\neg T$.

1. $S = a$ for an atom a. Since $\mathcal{I}|_M$ is a quasi-restriction of $\mathcal{I}$ to M we have $a^{\mathcal{I}|_M} = a^{\mathcal{I}}$, and thus $d \in S^{\mathcal{I}}$ iff $d \in S^{\mathcal{I}|_M}$ is trivially satisfied.
2. $S = A$ for a sort A. We have $A^{\mathcal{I}|_M} \cap M = A^{\mathcal{I}} \cap M$ since $\mathcal{I}|_M$ is a quasi-restriction of $\mathcal{I}$ to M, and $d \in M$ since $gen(d) \subseteq M$. This yields $d \in S^{\mathcal{I}}$ iff $d \in S^{\mathcal{I}|_M}$.
3. $S = p{\downarrow}q$ for paths p, q. Let $p = f_1 ... f_k$ and $q = g_1 ... g_l$.
Assume that $d \in (p{\downarrow}q)^{\mathcal{I}}$, that is, $dp^{\mathcal{I}}$ and $dq^{\mathcal{I}}$ are both defined and equal. To be more precise, that means there exist d_1, ..., d_k, e_1, ..., e_l in $\mathbf{D}^{\mathcal{I}}$ such that $(d, d_1) \in f_1^{\mathcal{I}}$, $(d_1, d_2) \in f_2^{\mathcal{I}}$, ..., $(d_{k-1}, d_k) \in f_k^{\mathcal{I}}$, $(d, e_1) \in g_1^{\mathcal{I}}$, $(e_1, e_2) \in g_2^{\mathcal{I}}$, ..., $(e_{l-1}, e_l) \in g_l^{\mathcal{I}}$, and $d_k = e_l$. Obviously, d_1, ..., d_k, e_1, ..., e_l are all elements

of $gen(d)$, and thus of M. But then $dp^{\mathcal{I}|_M} = d_k = e_l = dq^{\mathcal{I}|_M}$, which shows $d \in (p\!\downarrow\! q)^{\mathcal{I}|_M}$.
Conversely, assume that $d \in (p \downarrow q)^{\mathcal{I}|_M}$, that is, $dp^{\mathcal{I}|_M} = e = dq^{\mathcal{I}|_M}$ for an element $e \in \mathbf{D}^{\mathcal{I}|_M}$. Obviously, this implies $dp^{\mathcal{I}} = e = dq^{\mathcal{I}}$, and thus $d \in (p\!\downarrow\! q)^{\mathcal{I}}$.

4. $S = \exists L(T)$ for a feature term T and a description L of a regular set of paths. Assume that $d \in (\exists L(T))^{\mathcal{I}}$, that is, there exists a path $p \in L$ and an element $e \in T^{\mathcal{I}}$ such that $dp^{\mathcal{I}} = e$. As above, $dp^{\mathcal{I}} = e$ implies $dp^{\mathcal{I}|_M} = e$. In addition, $e \in gen(d)$ yields $gen(e) \subseteq gen(d) \subseteq M$. Thus we can apply the induction hypothesis to T and e, and get $e \in T^{\mathcal{I}|_M}$. This shows $d \in (\exists L(T))^{\mathcal{I}|_M}$.
   The other direction can be proved in a similar way.

5. $S = T_1 \sqcap T_2$. By induction, we have for $i = 1, 2$ that $d \in T_i^{\mathcal{I}}$ iff $d \in T_i^{\mathcal{I}|_M}$. This yields $d \in (T_1 \sqcap T_2)^{\mathcal{I}}$ iff $d \in (T_1 \sqcap T_2)^{\mathcal{I}|_M}$.

6. $S = \neg T$. By induction, we have $d \in T^{\mathcal{I}}$ iff $d \in T^{\mathcal{I}|_M}$. This yields $d \in (\neg T)^{\mathcal{I}}$ iff $d \in (\neg T)^{\mathcal{I}|_M}$.

This completes the proof of the lemma. ∎

If we take $M = gen(d)$ in this lemma we get

Theorem 3.2 *Let $\mathcal{I}$ be a feature algebra, d be an element of $\mathbf{D}^{\mathcal{I}}$, and S be a feature term. Then*

$$d \in S^{\mathcal{I}} \iff d \in S^{\mathcal{I}|_{gen(d)}}$$

provided that $\mathcal{I}|_{gen(d)}$ is a quasi-restriction of $\mathcal{I}$ to $gen(d)$.

The theorem shows that one can restrict the attention to rooted feature algebras if one is interested in the satisfiability of a feature term.

Corollary 3.3 *A feature term S is satisfiable if and only if there exists a rooted feature algebra $\mathcal{I}$ with root $d \in \mathbf{D}^{\mathcal{I}}$ such that $d \in S^{\mathcal{I}}$.*

As another consequence of Theorem 3.2 one gets that the behavior of an atom with respect to feature terms only depends on its behavior with respect to sorts.

Corollary 3.4 *Let b be an atom, and let $\mathcal{I}$ and $\mathcal{J}$ be feature algebras such that $b^{\mathcal{I}} \in A^{\mathcal{I}}$ if and only if $b^{\mathcal{J}} \in A^{\mathcal{J}}$ holds for all sorts A. Then we have $b^{\mathcal{I}} \in S^{\mathcal{I}}$ if and only if $b^{\mathcal{J}} \in S^{\mathcal{J}}$ for all feature terms S.*

Proof. Without loss of generality we may assume that $\mathbf{A}^{\mathcal{I}} = \mathbf{A}^{\mathcal{J}}$ (otherwise we could rename the domains of the interpretations appropriately). For an atom b the set $gen(b^{\mathcal{I}})$ is always a singleton set consisting of the element $b^{\mathcal{I}}$ alone. Thus any quasi-restriction $\mathcal{I}_b$ of the feature algebra $\mathcal{I}$ to $gen(b^{\mathcal{I}})$ has the set $\mathbf{A}^{\mathcal{I}}$ as its domain. By the definition of quasi-restrictions, all the features are interpreted as empty relations in $\mathcal{I}_b$. For all sorts A we have $b^{\mathcal{I}} \in A^{\mathcal{I}}$ iff $b^{\mathcal{I}_b} \in A^{\mathcal{I}_b}$, but the behavior of elements $a^{\mathcal{I}_b}$ for $a \neq b$ with respect to sorts is arbitrary. Since $\mathcal{I}_b$ has $\mathbf{A}^{\mathcal{I}} = \mathbf{A}^{\mathcal{J}}$ as its domain, the assumption of the corollary implies that $\mathcal{I}_b$ can be seen as a quasi-restriction of $\mathcal{J}$ as well. Together with Theorem 3.2 this observation completes the proof of the corollary. ■

As already pointed out earlier, this corollary will be important for the proofs of Proposition 4.1 and Lemma 5.1.

4 Consistency of Sort Equations

When working with a finite set of sort equations, it is important to know whether this set is consistent, i.e., whether it has a model. With respect to an inconsistent set of sort equations, all feature terms are unsatisfiable and thus equivalent to $\perp$. Surprisingly, decidability of the consistency problem for sort equations depends on the existence of atoms in the feature term language. In the first part of this section we will show that consistency is decidable if the language contains at least one atom. In principle, the reason for this is that, in the presence of atoms, whenever there is a model, there is a rather trivial model, consisting of the denotation of atoms only. In the second part of the section, it will be shown that consistency is in general undecidable if there are no atoms.

As pointed out before, it is sufficient to consider a single sort equation of the form $S \doteq \perp$. First, assume that there exists at least one atom. In this case, consistency of the sort equation $S \doteq \perp$ can be characterized as follows.

Proposition 4.1 *Assume that* $\mathbf{A} \neq \emptyset$. *Then the sort equation* $S \doteq \perp$ *is consistent if and only if for all atoms* a *the feature term* $\neg S \sqcap a$ *is satisfiable.*

Proof. Let $\mathcal{I}$ be a model of the sort equation $S \doteq \perp$. This means that $S^{\mathcal{I}} = \emptyset$, and thus $(\neg S)^{\mathcal{I}} = \mathbf{D}^{\mathcal{I}}$. Consequently, we have for any atom a that $a^{\mathcal{I}} \in \mathbf{D}^{\mathcal{I}} = (\neg S)^{\mathcal{I}}$. But then $a^{\mathcal{I}} \in (\neg S \sqcap a)^{\mathcal{I}}$, which shows that this feature term is satisfiable.

On the other hand, assume that for any atom a there is a feature algebra $\mathcal{I}_a$ such that $(\neg S \sqcap a)^{\mathcal{I}_a} \neq \emptyset$. This means that for any atom a we have $a^{\mathcal{I}_a} \in (\neg S)^{\mathcal{I}_a}$. We define a new feature algebra $\mathcal{I}$ as follows: $\mathbf{D}^{\mathcal{I}} := \{a^{\mathcal{I}} \mid a \text{ is an atom}\}$ where the $a^{\mathcal{I}}$ are assumed to be different individuals; for all features f we define $f^{\mathcal{I}} := \emptyset$; and for all sorts A we define $A^{\mathcal{I}} := \{a^{\mathcal{I}} \mid a^{\mathcal{I}_a} \in A^{\mathcal{I}_a}\}$. Obviously, $\mathcal{I}$ is a feature algebra. By Corollary 3.4 we get for all atoms a that $a^{\mathcal{I}} \in (\neg S)^{\mathcal{I}}$ because $a^{\mathcal{I}_a} \in (\neg S)^{\mathcal{I}_a}$. This shows that $(\neg S)^{\mathcal{I}} = \mathbf{D}^{\mathcal{I}}$, and thus $S^{\mathcal{I}} = \emptyset$. ■

This proposition reduces the question of consistency of sort equations to the problem of satisfiability of feature terms of the form $T \sqcap a$. The following lemma shows that we can further restrict ourselves to the case where T does not contain path quantifications.

Lemma 4.2 *Let T be a feature term and a be an atom. Then there exists a feature term T' without path quantifications such that $T \sqcap a \sim T' \sqcap a$.*

Proof. It is easy to see that a term of the form $\forall L(T_1) \sqcap a$ or $\exists L(T_1) \sqcap a$ is equivalent to the term $T_1 \sqcap a$ if the empty word ε is in L. If $\varepsilon \notin L$, then $\forall L(T_1) \sqcap a$ is equivalent to $\top \sqcap a$ and $\exists L(T_1) \sqcap a$ is equivalent to $\bot \sqcap a$. Using this fact, the lemma can easily be proved by induction. Please note that, if T starts with a negation, then this negation can be pushed into the term with the help of de Morgan's rules and the fact that $\neg\forall L(S) \sim \exists L(\neg S)$ and $\neg\exists L(S) \sim \forall L(\neg S)$. ■

Since satisfiability of feature terms not containing path quantifications is decidable (Smolka, 1992), the proposition and the lemma yield

Theorem 4.3 *Assume that $\mathbf{A} \neq \emptyset$, and let $\mathcal{E}$ be a finite set of sort equations. Then it is decidable whether $\mathcal{E}$ is consistent or not.*

Proof. The only remaining problem is that, if the set of atoms is infinite, we should have to consider infinitely many terms of the form $\neg S \sqcap a$ in order to check the condition of Proposition 4.1. However, the sort equations contain only finitely many atoms. It is easy to see that it is enough to consider these finitely many atoms, and only one of the other atoms as specimen. ■

Now let us consider the case where $\mathbf{A} = \emptyset$, that is, there is no atom. We shall show that consistency of sort equations may become undecidable, even if the terms occurring in the sort equations do not contain path quantifications. This result will be proved by a reduction of the word problem for groups. To this purpose we rephrase the word problem in such a way that it fits into our framework.

Let Σ be a nonempty set of symbols, Σ^* be the set of words over Σ, and ε be the empty word. Under concatenation of words, Σ^* is a monoid whose neutral element is ε. A **congruence** is an equivalence relation $\sim$ on Σ^* such that $p \sim q$ implies $rpr' \sim rqr'$ for all p, q, r, $r' \in \Sigma^*$. If $\sim$ is clear from the context, we use $\overline{p}$ to denote the equivalence class of a word $p \in \Sigma^*$ with respect to $\sim$. The quotient $\Sigma^*/\sim$ is again a monoid under the operation $\overline{p}\,\overline{q} := \overline{pq}$.

A **Thue equation** over Σ is a set $\{p, q\}$ consisting of two words $p, q \in \Sigma^*$. A **Thue system** over Σ is a finite set T of Thue equations over Σ. Every Thue system T over Σ defines a binary relation $\leftrightarrow_T$ on Σ^* by

$$u \leftrightarrow_T v :\Longleftrightarrow \exists\, w_1, w_2 \in \Sigma^* \;\exists\, \{p, q\} \in T\colon \;\; u = w_1 p w_2 \;\wedge\; v = w_1 q w_2.$$

We use $\sim_T$ to denote the reflexive and transitive closure of $\leftrightarrow_T$ on Σ^*. It is easy to see that $\sim_T$ is a congruence on Σ^*. To be more precise, $\sim_T$ is the least congruence $\sim$ such that $p \sim q$ for every Thue equation $\{p, q\}$ in T.

It is known that there exists a Thue system T consisting of seven equations over a two-element alphabet such that it is undecidable for two words p, q whether $p \sim_T q$ holds or not (see, for instance, (Boone, 1959)). In the following we shall need a stronger version of this undecidability result, which is due to Novikov and Boone (see (Novikov, 1955; Boone, 1959; Stillwell, 1982)): there is a finite set of symbols Σ and a Thue system $G = \{\{p_1, \varepsilon\}, \ldots, \{p_n, \varepsilon\}\}$ such that

1. for every $f \in \Sigma$ there is some $q \in \Sigma^*$ such that G contains the Thue equation $\{fq, \varepsilon\}$
2. the set of words p with $p \not\sim_G \varepsilon$ is not recursively enumerable.

In particular, it is undecidable whether $p \sim_G \varepsilon$ or not. Note that property (1) implies that $\Sigma^*/\sim_G$ is a group.

Theorem 4.4 *Assume that* $\mathbf{A} = \emptyset$*, that is, there is no atom. Then there exists a feature term* S *of the form*

$$p_1 \downarrow \varepsilon \sqcap \ldots \sqcap p_n \downarrow \varepsilon$$

such that the set of paths p *for which the sort equation*

$$S \sqcap \neg(p \downarrow \varepsilon) \doteq \top$$

is consistent is not recursively enumerable. In particular, it is undecidable whether the sort equation $S \sqcap \neg(p \downarrow \varepsilon) \doteq \top$ *is consistent or not.*

Proof. Suppose that Σ is a set of symbols and $G = \{\{p_1, \varepsilon\}, \ldots, \{p_n, \varepsilon\}\}$ a Thue system over Σ with the properties stated in the theorem by Novikov and Boone. We regard elements of Σ as features and words over Σ as paths. Let S be the feature term

$$p_1 \downarrow \varepsilon \sqcap \ldots \sqcap p_n \downarrow \varepsilon.$$

To prove our claim it suffices to show that for every $p \in \Sigma^*$ the sort equation $S \sqcap \neg(p \downarrow \varepsilon) \doteq \top$ is consistent if and only if $p \not\sim_G \varepsilon$.

"$\Rightarrow$" Suppose $p \not\sim_G \varepsilon$. We construct a feature algebra $\mathcal{I}$ satisfying $S \sqcap \neg(p \downarrow \varepsilon) \doteq \top$ as follows:

$$\begin{aligned} \mathbf{D}^{\mathcal{I}} &:= \Sigma^*/\sim_G \\ \overline{q}\, f^{\mathcal{I}} &:= \overline{qf} \quad \text{for every } f \in \Sigma \text{ and } q \in \Sigma^*. \end{aligned}$$

Since $\sim_G$ is the congruence generated by G, we have $\overline{p}_i = \overline{\varepsilon}$ for every Thue equation $\{p_i, \varepsilon\}$ in G. This implies $\overline{q}\, p_i^{\mathcal{I}} = \overline{qp_i} = \overline{q\varepsilon} = \overline{q}$ for every $\overline{q} \in \mathbf{D}^{\mathcal{I}}$. Hence, $\mathcal{I}$ satisfies $S \doteq \top$.

Assume that $\mathcal{I}$ does not satisfy $\neg(p \downarrow \varepsilon) \doteq \top$. Then there is some $\overline{q} \in \mathbf{D}^{\mathcal{I}}$ such that $\overline{q}\, p^{\mathcal{I}} = \overline{q}$, which implies that $\overline{q} = \overline{qp} = \overline{q}\,\overline{p}$. Since $\Sigma^*/\sim_G$ is a group, the element $\overline{q}$ has an inverse $\overline{q'}$. Then $\overline{p} = \overline{q'}\,\overline{q}\,\overline{p} = \overline{q'q} = \overline{\varepsilon}$, that is $p \sim_G \varepsilon$. We thus have obtained a contradiction to the fact that $p \not\sim_G \varepsilon$. Hence, $\overline{q}\,p^{\mathcal{I}} \neq \overline{q}$ for all $\overline{q} \in \mathbf{D}^{\mathcal{I}}$, which implies that $\mathcal{I}$ satisfies $\neg(p\!\downarrow\!\varepsilon) \doteq \top$. Since $\mathcal{I}$ satisfies $S \doteq \top$ and $\neg(p\!\downarrow\!\varepsilon) \doteq \top$, it follows that $\mathcal{I}$ satisfies $S \sqcap \neg(p\!\downarrow\!\varepsilon) \doteq \top$.

"$\Leftarrow$" Suppose $p \sim_G \varepsilon$. Assume there is a feature algebra $\mathcal{I}$ that satisfies $S \sqcap \neg(p\!\downarrow\!\varepsilon) \doteq \top$. We define an equivalence relation $\sim$ on Σ^* by

$$q \sim q' \;:\Longleftrightarrow\; \mathcal{I} \text{ satisfies } q\!\downarrow\!q' \doteq \top.$$

Since $\mathcal{I}$ satisfies $S \sqcap \neg(p\!\downarrow\!\varepsilon) \doteq \top$, it follows that $\mathcal{I}$ satisfies $p_i\!\downarrow\!\varepsilon \doteq \top$ for every Thue equation $\{p_i, \varepsilon\}$ in G. By property (1) of G, for every $f \in \Sigma$ there is some $q \in \Sigma^*$ such that $\mathcal{I}$ satisfies $fq\!\downarrow\!\varepsilon \doteq \top$. This means that for every $d \in \mathbf{D}^{\mathcal{I}}$ we have $df^{\mathcal{I}}q^{\mathcal{I}} = d$. Hence, every $f^{\mathcal{I}}$ is a total function on $\mathbf{D}^{\mathcal{I}}$. We conclude that for all q, q', r, $r' \in \Sigma^*$ the feature algebra $\mathcal{I}$ satisfies $rqr'\!\downarrow\!rq'r' \doteq \top$ if $\mathcal{I}$ satisfies $q\!\downarrow\!q' \doteq \top$. Thus, $\sim$ is a congruence.

For every Thue equation $\{p_i, \varepsilon\}$ in G, the feature algebra $\mathcal{I}$ satisfies $p_i\!\downarrow\!\varepsilon \doteq \top$, which implies $p_i \sim \varepsilon$. By definition, $\sim_G$ is the least congruence with this property. Therefore, $p \sim_G \varepsilon$ implies $p \sim \varepsilon$, that is, $\mathcal{I}$ satisfies $p\!\downarrow\!\varepsilon \doteq \top$. This contradicts our assumption that $\mathcal{I}$ satisfies $\neg(p\!\downarrow\!\varepsilon) \doteq \top$. We conclude that $S \sqcap \neg(p\!\downarrow\!\varepsilon) \doteq \top$ is unsatisfiable. ∎

Since consistency of an equation $S \doteq \top$ is equivalent to consistency of the equation $\neg S \doteq \bot$ we get the following immediate consequence of the theorem.

Corollary 4.5 *Assume that* $\mathbf{A} = \emptyset$. *Then consistency of a sort equation* $S \doteq \bot$ *is undecidable.*

5 Internalizing Sort Equations

The purpose of this section is to show that sort equations do not enhance the expressive power of a feature term language that allows for path quantification. In fact, it will turn out that satisfiability of a feature term with respect to a finite set of sort equations is equivalent to pure satisfiability of a set of feature terms. We call this process of encoding sort equations into feature terms with path quantification "internalization." Similar result have been obtained for propositional dynamic logic (see e.g. (Kozen and Tiuryn, 1990), p.805, Proposition 16), but again, the presence of atoms makes the formulation and the proof of our result more complex.

As mentioned before, it is sufficient to consider only one sort equation of the form $S \doteq \bot$. Recall that we denote by $\mathbf{F}^*$ the set of all paths, that is, the set of all words over $\mathbf{F}$. Obviously, a feature term T can be satisfiable with respect to $S \doteq \bot$ only if this equation is consistent. For this reason, the condition for consistency of sort equations in the presence of atoms occurs in the formulation of the following internalization lemma.

Lemma 5.1 *The feature term T is satisfiable w.r.t. the sort equation $S \doteq \bot$ if and only if the feature term $T \sqcap \forall \mathbf{F}^*(\neg S)$, and the feature terms $\neg S \sqcap a$ for all atoms a are satisfiable.*

Proof. Assume that $\mathcal{I}$ is a feature algebra such that $S^{\mathcal{I}} = \bot^{\mathcal{I}} = \emptyset$ and $T^{\mathcal{I}} \neq \emptyset$. Obviously, $S^{\mathcal{I}} = \emptyset$ means that all the elements of $\mathbf{D}^{\mathcal{I}}$ are in $(\neg S)^{\mathcal{I}}$. In particular, we have $a^{\mathcal{I}} \in (\neg S)^{\mathcal{I}}$ for all atoms a. This shows that $(\neg S \sqcap a)^{\mathcal{I}} \neq \emptyset$.

Let $d \in \mathbf{D}^{\mathcal{I}}$ be such that $d \in T^{\mathcal{I}}$. In order to prove that $d \in (T \sqcap \forall \mathbf{F}^*(\neg S))^{\mathcal{I}}$ it is enough to show that $d \in (\forall \mathbf{F}^*(\neg S))^{\mathcal{I}}$. Let $p \in \mathbf{F}^*$ and $e \in \mathbf{D}^{\mathcal{I}}$ be such that $dp^{\mathcal{I}} = e$. Since all the elements of $\mathbf{D}^{\mathcal{I}}$ are in $(\neg S)^{\mathcal{I}}$ we have $e \in (\neg S)^{\mathcal{I}}$, which completes the proof of the "only-if" part of the lemma.

Conversely, assume that the feature term $T \sqcap \forall \mathbf{F}^*(\neg S)$ and the feature terms $\neg S \sqcap a$ for all atoms a are satisfiable. Let $\mathcal{I}$ be a feature algebra such that $(T \sqcap \forall \mathbf{F}^*(\neg S))^{\mathcal{I}} \neq \emptyset$, and for all atoms a let $\mathcal{I}_a$ be a feature algebra such that $(\neg S \sqcap a)^{\mathcal{I}} \neq \emptyset$. Let $d \in \mathbf{D}^{\mathcal{I}}$ be such that $d \in (T \sqcap \forall \mathbf{F}^*(\neg S))^{\mathcal{I}}$. We want to define a quasi-restriction $\mathcal{I}|_{gen(d)}$ of $\mathcal{I}$ to $gen(d)$ which satisfies the sort equation $S \doteq \bot$ and interprets the feature term T as nonempty set. For that purpose we have to fix the interpretation of the sorts on $RA := \mathbf{A}^{\mathcal{I}} - gen(d)$ in an appropriate way. This can be done as follows: For all sorts A we define $A^{\mathcal{I}|_{gen(d)}} := (A^{\mathcal{I}} \cap gen(d)) \cup \{a^{\mathcal{I}} \mid a^{\mathcal{I}} \in RA \text{ and } a^{\mathcal{I}_a} \in A^{\mathcal{I}_a}\}$.

By Theorem 3.2 we have $d \in (T \sqcap \forall \mathbf{F}^*(\neg S))^{\mathcal{I}|_{gen(d)}}$ since $d \in (T \sqcap \forall \mathbf{F}^*(\neg S))^{\mathcal{I}}$. In particular, this yields $d \in T^{\mathcal{I}|_{gen(d)}}$. It remains to be shown that $S^{\mathcal{I}|_{gen(d)}} = \emptyset$. Assume that there exists $e \in \mathbf{D}^{\mathcal{I}|_{gen(d)}} = gen(d) \cup RA$ such that $e \in S^{\mathcal{I}|_{gen(d)}}$.

If $e \in gen(d)$ then there exist a path $p \in \mathbf{F}^*$ such that $e = dp^{\mathcal{I}}$. But then $e \in S^{\mathcal{I}|_{gen(d)}}$ contradicts $d \in (\forall \mathbf{F}^*(\neg S))^{\mathcal{I}|_{gen(d)}}$.

Assume that $e \in RA$, that is, $e = a^{\mathcal{I}}$ for an atom a such that $a^{\mathcal{I}} \notin gen(d)$. We have defined $\mathcal{I}|_{gen(d)}$ such that $e = a^{\mathcal{I}|_{gen(d)}} \in A^{\mathcal{I}|_{gen(d)}}$ iff $a^{\mathcal{I}_a} \in A^{\mathcal{I}_a}$ holds for all sorts A. By Corollary 3.4 we get $e = a^{\mathcal{I}|_{gen(d)}} \in (\neg S \sqcap a)^{\mathcal{I}|_{gen(d)}}$ since $a^{\mathcal{I}_a} \in (\neg S \sqcap a)^{\mathcal{I}_a}$. This is a contradiction to our assumption that $e \in S^{\mathcal{I}|_{gen(d)}}$. ∎

If there are no atoms, that is, if $\mathbf{A} = \emptyset$, then the condition "the feature terms $\neg S \sqcap a$ for all atoms a are satisfiable" is void. This yields

Theorem 5.2 *Assume that $\mathbf{A} = \emptyset$. Then the feature term T is satisfiable w.r.t. the sort equation $S \doteq \bot$ if and only if the feature term $T \sqcap \forall \mathbf{F}^*(\neg S)$ is satisfiable.*

On the other hand, if there exists at least on atom, the condition is equivalent to the consistency of the sort equation $S \doteq \bot$.

Theorem 5.3 *Assume that $\mathbf{A} \neq \emptyset$. Then the feature term T is satisfiable w.r.t. the sort equation $S \doteq \bot$ if and only if the sort equation $S \doteq \bot$ is consistent and the feature term $T \sqcap \forall \mathbf{F}^*(\neg S)$ is satisfiable.*

6 Satisfiability is Undecidable

As an easy consequence of the undecidability result of Section 4 we get that satisfiability of feature terms w.r.t. sort equations is undecidable, if we have no atoms. In fact, the feature term $\top$ is satisfiable w.r.t. a finite set of sort equations $\mathcal{E}$ if and only if $\mathcal{E}$ is consistent. Please note that in the proof of the undecidability result we did not use path quantification.

On the other hand, we have seen that consistency of sort equations is decidable, if we have at least one atom. But satisfiability of feature terms w.r.t. sort equations is nevertheless undecidable in this case. This is shown in (Smolka, 1992) in the presence of three features, two atoms and one sort. Again, the sort equations and the feature term constructed in (Smolka, 1992) do not contain path quantifications.

Taking the two results together we thus have

Theorem 6.1 *Satisfiability of feature terms w.r.t. sort equations is undecidable. This holds even if path quantifications are disallowed, and it does not depend on whether* $\mathbf{A} = \emptyset$ *or* $\mathbf{A} \neq \emptyset$.

In the light of Section 5, this theorem shows that satisfiability of feature terms with path quantifications is undecidable, independently on whether we have atoms or not.

Theorem 6.2 *Satisfiability of feature terms with path quantification is undecidable. This result does not depend on whether* $\mathbf{A} = \emptyset$ *or* $\mathbf{A} \neq \emptyset$.

Proof. Assume that satisfiability of feature terms with path quantifications is decidable. Then the characterizations of satisfiability of feature terms w.r.t. sort equations given in Theorem 5.2 (for $\mathbf{A} = \emptyset$) or Theorem 5.3 (for $\mathbf{A} \neq \emptyset$) would yield a decision criterion for satisfiability w.r.t. sort equations. This is a contradiction to Theorem 6.1. ■

7 Conclusion

We have studied the expressivity of functional uncertainty in a feature term language with negation and obtained two main results: satisfiability is undecidable and sort equations can be internalized.

For practical applications in grammar formalisms the language studied in this paper is probably too expressive since general negation is not needed. Thus it would be interesting to find out whether satisfiability of feature terms built from the forms a, A, $p{\downarrow}q$, $\exists L(S)$, and $S \sqcap S'$ is decidable.

Feature logics are closely related to terminological logics (Brachman and Schmolze, 1985; Levesque and Brachman, 1987; Nebel and Smolka, 1990; Schmidt-Schauß and Smolka, 1991), which are employed in knowledge representation and grew out of research in semantic networks and frame systems. The essential difference between these two formalisms is that in terminological logics attributes can be nonfunctional while they must be functional in feature logics.

Baader (1991) studies a terminological logic that can be obtained from the feature logic in this paper by three changes: disallow atoms and agreements, and admit also interpretations that interpret features as nonfunctional binary relations. He shows that in this logic satisfiability of "feature terms" (which are called concept terms in this context) is decidable. Since concept equations (i.e., the equivalent of the sort equations of the present paper) can also be internalized with the help of path quantifications, the algorithm given in (1991) also yields a decision procedure for satisfiability w.r.t. concept equations. Baader's algorithm can easily be adapted to the case where one allows only functional binary relations. This means that the feature logic of the present paper becomes decidable if agreements and atoms are disallowed.

Similar results for terminological logics have independently been obtained by Schild (1991) as byproducts of the correspondence he exhibits between terminological logics and dynamic logics. In addition, he shows that this correspondence also yields complexity results for the terminological logic considered by Baader, and for our feature logic if agreements and atoms are disallowed. In both cases, one has an EXPTIME-complete satisfiability problem.

References

F. Baader. Augmenting concept languages by transitive closure of roles: An alternative to terminological cycles. *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 446–451, Sydney, Australia, 1991.

P. Blackburn and E. Spaan. A modal perspective on the computational complexity of attribute value grammar. Logic Group Preprint Series No. 77, Department of Philosophy, Utrecht University, The Netherlands, 1992.

W. W. Boone. The word problem. *Ann. of Mat.*, 70:207–265, 1959.

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

R. Goldblatt. *Logics of Time and Computation.* CSLI Lecture Notes 7. Center for the Study of Language and Information, Stanford University, Cal., 1987.

M. Johnson. *Attribute-Value Logic and the Theory of Grammar.* CSLI Lecture Notes 16. Center for the Study of Language and Information, Stanford University, Cal., 1988.

R. M. Kaplan and J. 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, Mass., 1982.

R. M. Kaplan and J. T. Maxwell III. An algorithm for functional uncertainty. In *Proceedings of the 12th International Conference on Computational Linguistics*, pages 297–302, Budapest, Hungary, 1989.

R. M. Kaplan and J. T. Maxwell III. Constituent coordination in lexical-functional grammar. In *Proceedings of the 12th International Conference on Computational Linguistics*, pages 303–305, Budapest, Hungary, 1989.

R. T. Kasper and W. C. Rounds. A logical semantics for feature structures. In *Proceedings of the 24th Annual Meeting of the ACL, Columbia University*, pages 257–265, New York, N.Y., 1986.

M. Kay. Functional grammar. In *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistics Society*, Berkeley, Cal., 1979. Berkeley Linguistics Society.

R. Kozen and J. Tiuryn. Logics of programs. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, pages 789–840. Elsevier, Amsterdam and MIT Press, Cambridge, Mass., 1990.

H. J. Levesque and R. J. Brachman. Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3:78–93, 1987.

B. Nebel and G. Smolka. Representation and reasoning with attributive descriptions. In K. Bläsius, U.Hedtstück, and C.-R. Rollinger, editors, *Sorts and Types in Artificial Intelligence*, volume 418 of *Lectures Notes in Computer Science*, pages 112–139. Springer-Verlag, Berlin, Germany, 1990.

P. S. Novikov. On the algorithmic unsolvability of the word problem (in Russian). *Dokl. Akad. Nauk SSSR Mathematičeskii Institut Trudy*, 44, 1955. Moscow.

C. Pollard and I. A. Sag. *Information-Based Syntax and Semantics*, volume 13 of *CSLI Lecture Notes*. Center for the Study of Language and Information, Stanford University, Cal., 1987.

W. C. Rounds and R. T. Kasper. A complete logical calculus for record structures representing linguistic information. In *Proceedings of the 1st IEEE Symposium on Logic in Computer Science*, pages 38–43, Boston, Mass., 1986.

K. Schild. A correspondence theory for terminological logics: Preliminary report. *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 466–471, Sydney, Australia, 1991.

M. Schmidt-Schauß and G. Smolka. Attributive concept descriptions with complements. *Artificial Intelligence*, 47, 1991.

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

S. M. Shieber, H. Uszkoreit, F. C. Pereira, J. Robinson, and M. Tyson. The formalism and implementation of PATR-II. In B. J. Grosz and M. Stickel, editors, *Research on Interactive Acquisition and Use of Knowledge*. SRI International, Artificial Intelligence Center, Menlo Park, Cal., 1983.

G. Smolka. A feature logic with subsorts. LILOG Report 33, IWBS, IBM Germany, Stuttgart, Germany, May 1988.

G. Smolka. Feature constraint logics for unification grammars. *Journal of Logic Programming*, 12:51–87, 1992.

J. Stillwell. The word problem and the isomorphism problem for groups. *Bulletin AMS*, 6:33–56, 1982.