

Generation as Structure Driven Derivation

Jürgen Wedekind
 Institute for Natural Language Processing
 University of Stuttgart
 FRG

Abstract

This paper describes two algorithms which construct two different types of generators for lexical functional grammars (LFGs). The first type generates sentences from functional structures and the second from semantic structures. The latter works on the basis of extended LFGs, which contain a mapping from f-structures into semantic structures. Both algorithms can be used on all grammars within the respective class of LFG-grammars. Thus sentences can be generated from input structures by means of LFG-grammars and the same grammar formalism, although not necessarily the same grammar, can be used for both analysis and synthesis.

1 Introduction

Using the same grammar formalism, or even the same grammar, for both analysis and synthesis is usually regarded as an elegant, efficient and sometimes even as the psychologically most plausible approach to natural language parsing and generation. In this paper we want to show that this approach can be realized within the LFG-framework by defining two generation algorithms.¹ The first permits the construction of generators for LFGs which generate sentences from functional structures. The second constructs generators which generate sentences from semantic structures.

Both algorithms are based on concepts of derivation for LFGs which can be strengthened in such a way that the derivation can be driven by a given input structure. The principles which underlie the control mechanisms for the derivation are sufficiently general to also be applicable to other unification based formalisms which allow the derivation of functional and/or semantic structures in parallel to constituent structures (e.g. PATR (cf. e.g. [Shieber 83], [Karttunen 86])).

For the generation from semantic structures, a derivation concept of this type can be defined, if, following a proposal by Halvorsen and Kaplan (cf. [Halvorsen 87], [Kaplan 87]), projector equations are used to describe (co-describe) the semantic structure of a sentence. Since the projector mechanism is independent of the specific type of semantic theory, the algorithm works as long as this mechanism is used to build up semantic representations.

In addition, the derivation driven by the semantic structure can be directed by constraints over the f-structure. Usually, semantic information by itself is regarded as providing a basis for generation which is too weak to capture relevant distinctions in the surface form. This additional mechanism could contribute to make the generation more sensitive to syntactic and pragmatic information. Thus, additional functional information can enforce a specific syntactic realization, such as passivization, topicalization, extraposition or discontinuous realization of constituents.

It should be clear that the specific constraints which have to be or can be imposed on this kind of generation are subject to empirical studies on questions of syntax, discourse and dialogue.² Similarly, for machine translation, one will need interlingual comparative research on these subjects to establish what relevant functional information can be drawn directly from the f-structures of the source text, in order to guarantee a coherent target text. These important empirical questions will not be addressed directly in this paper. We consider it an essential requirement for an adequate generation algorithm, however, that it provides the respective possibilities of control.

2 Generation from functional structures

In this section we describe an algorithm which constructs a generator for an arbitrary given LFG³ which generates terminal strings from functional structures.⁴ Such an algorithm has to define for every LFG a relation $\Gamma_\phi(\Phi, s)$ (s is *generable* from Φ) between directed acyclic graphs (DAGs) and terminal strings.

$\forall \Phi \in DAG \forall s \in V_T^* (\Delta_\phi(s, \Phi) \text{ iff } C(\dots \Phi, s \dots))$

Depending on what the adequacy condition $C(\dots \Phi, s \dots)$ for this relation is, one will get different adequacy criteria for possible explications of what 'generation' can mean within the LFG-framework. We started from the

- perhaps too idealized - condition that is normally used for the relation between strings and f-structures Δ_ϕ , specified by an adequate parsing algorithm for LFG.⁵

$\forall \Phi \in DAG \forall s \in V_T^* (\Delta_\phi(s, \Phi) \text{ iff } s \text{ is derivable from the start symbol } S \text{ and } s \text{ has f-structure } \Phi)$

If we use this condition and insert it into the schema $C(\dots \Phi, s \dots)$,

(*) $\forall \Phi \in DAG \forall s \in V_T^* (\Gamma_\phi(\Phi, s) \text{ iff } s \text{ is derivable from the start symbol } S \text{ and } s \text{ has f-structure } \Phi)$

then the relation Δ_ϕ is simply defined inversely by an adequate generation algorithm ($\Gamma_\phi^{-1} = \Delta_\phi$ and also $\Delta_\phi^{-1} = \Gamma_\phi$). This means that a generator for an LFG G must accept an input structure Φ by building up a string s , iff s is derivable with f-structure Φ in G . Thus, the generator for an LFG constructed by the algorithm which satisfies this condition is simply a parser or transducer for the set of well-formed f-structures, which constructs for an input structure the set of all sentences which have this structure as their f-structure. This implies that the generator produces no output, if the input structure is not a well-formed f-structure.

2.1 Derivational background

The algorithm can be based on derivation concepts for LFGs which can be strengthened in such a way that the derivation can be driven by an input structure (*structure driven derivation*). However, the derivability conditions formulated in [Kaplan/Bresnan 82] cannot be used directly. According to [Kaplan/Bresnan 82] a terminal string s is regarded as well-formed iff it satisfies the following conditions:

(WFF)

1. There is a c-structure c for s that can be derived by the context-free base of the grammar.
2. There is an f-structure Φ and a mapping ϕ from the c-structure nodes to the nodes of Φ such that Φ is the unique minimal f-structure that satisfies the annotations associated with the c-structure nodes. (The f-description solution algorithm (fds-algorithm) constructs both ϕ and Φ .)⁶
3. All constraints in the f-description are satisfied by Φ .
4. Φ is complete and coherent.

These conditions are tested in the following order: 1. \prec 2. \prec 3., 4.. Thus, if the f-structure is built up after the derivation of the c-structure,⁷ it is impossible to use the functional information contained in an input structure for the control of the derivation of the c-structure. A decidable generation procedure presupposes the possibility of comparing the input structure with the partial f-structure of a derived partial c-structure. Thus, in order to drive the c-structure derivation by a given input structure it is necessary to derive the partial f-structure in parallel to a partial c-structure. This means that one can use only those derivation concepts which make f-description solutions for partial c-structures available in each step of the derivation. The concept on which the following algorithm is based is described in more detail in [Wedekind 86]. According to this concept, a derivation is a sequence of quadruples (c, Φ, ϕ, C^ϕ) . Each quadruple consists of

- c a partial c-structure,
- Φ a partial f-structure,
- ϕ a mapping from the c-structure nodes into the set of nodes of Φ , and
- C^ϕ a constraint set.

(Φ and ϕ would be the result of the fds-algorithm if it were applied to the corresponding annotated c-structure. C^ϕ corresponds to the set of instantiated constraining equations contained in the f-description of c . See the example in fig. 1.) I follow the usual convention of identifying the c-structure nodes with sequences of integers. The linear order of the edges of a tree is normally encoded by numbering the arcs, and every node is identified with the sequence of integers numbering the arcs along the path from the root to that node.

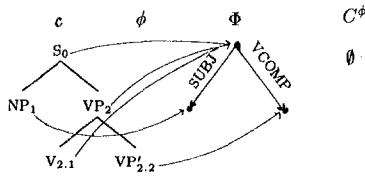


Fig. 1

The corresponding annotated c-structure is derivable by the following rules:

$$S \rightarrow NP \quad VP \quad VP \rightarrow V \quad VP'$$

$$(\uparrow \text{SUBJ}) = \downarrow \quad \uparrow = \downarrow \quad \uparrow = \downarrow \quad (\uparrow \text{VCOMP}) = \downarrow$$

The initial quadruple consists of an S-labeled root node c_S , an f-structure root node Φ_S , to which the S node is mapped by ϕ^S , and an empty constraint set C_S^ϕ .

$$\langle c_S, \Phi_S, \phi^S, C_S^\phi \rangle \quad c_S \quad \phi^S \quad \Phi_S \quad C_S^\phi$$

$$S_0 \longrightarrow \cdot \quad \emptyset$$

A grammar rule also introduces a quadruple.

$$V \rightarrow \langle c_r, \Phi_r, \phi^r, C_r^\phi \rangle$$

The functional part $\langle \Phi_r, \phi^r \rangle$ is obtained by applying the fds-algorithm to the annotated local tree represented by the rule, and by instantiating the metavariables in the constraining equations with the node indices of the local tree introduced by the rule. (The constraint set of a rule contains the constraints of the f-description of the local tree.) Fig. 2 gives an example.

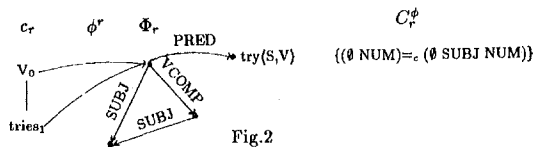


Fig. 2

This quadruple corresponds to the rule:

$$V \rightarrow \text{tries}$$

$$(\uparrow \text{PRED}) = \text{try}(\text{SUBJ}, \text{VCOMP})$$

$$(\uparrow \text{SUBJ}) = (\uparrow \text{VCOMP SUBJ})$$

$$(\uparrow \text{NUM}) =_c (\uparrow \text{SUBJ NUM})$$

Suppose that we have derived from the initial quadruple the quadruple given in fig. 1; then we can apply the V-rule, since the leaf 2.1 is labeled with V . The derived quadruple $\langle c', \Phi', \phi', C'^\phi \rangle$ consists of a c-structure which is the result of expanding 2.1 by c_r . The partial f-structure Φ' is the minimal extension of Φ which results from Φ by unifying the DAG Φ_r introduced by the rule with the substructure rooted by $\phi_{2.1}$. Since the new DAG Φ' is a homomorphic extension of Φ , the values of ϕ' for the old nodes (of c) are given by ϕ and the homomorphism. The values of ϕ' for the new nodes result from ϕ^r and from the value of ϕ for 2.1 (cf. the definition below). C'^ϕ contains besides C^ϕ (which is empty) the constraint $(2.1 \text{ NUM}) =_c (2.1 \text{ SUBJ NUM})$ which is constructed from $(\emptyset \text{ NUM}) =_c (\emptyset \text{ SUBJ NUM})$ by attaching 2.1 as a prefix to each node index within that constraint. The result is the constraint set of the f-description of the derived tree. Fig. 3 illustrates the result of the rule application.

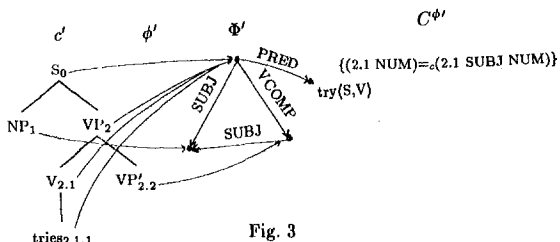


Fig. 3

If we reconstruct directed labeled connected rooted acyclic graphs (DAGs) as transition graphs of connected rooted acyclic finite state automata, whose leaves (states without leaving transitions) are labeled by a partial function m with atoms of a set A ($\Phi = (Q, L, \delta, q_0, A, m)$),⁸ then we can state the definition of the derivability relation Δ_ϕ as follows.

DEFINITION 2.1 A terminal string s is derivable with f-structure Φ_f ($\Delta_\phi(s, \Phi_f)$) iff there is a sequence $\omega_0 \dots \omega_n$ such that

- $\omega_0 = \langle c_S, \Phi_S, \phi^S, C_S^\phi \rangle$ and
- for all $\omega_i = \langle c, \Phi, \phi, C^\phi \rangle$, $\omega_{i+1} = \langle c', \Phi', \phi', C'^\phi \rangle$ ($0 \leq i < n$) there is a rule $V \rightarrow \langle c_r, \Phi_r, \phi^r, C_r^\phi \rangle$ and
- V is a label of a leaf μ of c

- c' is the result of expanding μ in c by c_r ,
- Φ' is the minimal extension of Φ which results from Φ by unifying Φ_r with the substructure rooted by ϕ_μ
- if ν is a node of c , whose ϕ -value is $\delta^*(q_0, p)$ in Φ , then its ϕ' -value in Φ' is $\delta'^*(q'_0, p)$
- if $\mu.j$ is a node of c' , not contained in c , the value of ϕ'_j is $\delta'^*(q'_0, q)$ and the value of ϕ for μ is $\delta^*(q_0, p)$, then the value of ϕ' for $\mu.j$ is $\delta'^*(q'_0, p.q)$ in Φ'
- C'^ϕ contains besides C^ϕ the constraints $(\mu.j p) =_c (\mu.i q)$ etc.⁹ for all $(j p) =_c (i q)$ etc. in C^ϕ

- s is the terminal string of the c-structure of ω_n ,
- Φ_f is equal to the DAG of ω_n (Φ_n)
- Φ_n satisfies all constraints in C_n^ϕ
- Φ_n is complete and coherent.¹⁰

2.2 Generation as f-structure driven derivation

In principle we could use this derivation concept for generation if we substituted the DAG in the initial quadruple by an input structure and mapped the S node to the root of the input structure. However, such a concept of generation would not satisfy the adequacy condition mentioned above. The derivation would not be adequately controlled by the input structure because it is not guaranteed that

- i) the information contained in the input structure is completely derived and
- ii) no additional information is introduced during the derivation.

It is possible, for example, to derive additional adjuncts or not to derive all adjuncts represented in the input structure. Due to the unification part of the derivation process, it is only guaranteed that the f-structure of the generated sentence is compatible with the input structure. The requirements i) and ii), which will be referred to as *completeness* and *coherence*,¹¹ show that the input structure is in fact a complex constraint with a positive and a negative part. The positive part (*completeness*), which requires that the input structure (Φ_{in}) is subsumed by the derived structure (Φ_n) ($\Phi_{in} \sqsubseteq \Phi_n$), can be made explicit by two kinds of constraints: existential constraints, which demand that

COMP_a: all paths of the input structure are derived,

and reentrancy constraints, which demand that

COMP_b: all reentrancies of the input structure are derived.¹²

The negative part (*coherence*) which demands that

COH: the derived structure is subsumed by the input structure ($\Phi_n \sqsubseteq \Phi_{in}$)

ensures that the f-structure of the generated string is the unique minimal structure that satisfies the completeness constraints expressed by the input structure.

The central problem of generation designed as structure driven derivation is the control of the fulfillment of these conditions. Since this problem also occurs within other formalisms which build up DAG-structures during the derivation process, the solutions proposed here for LFG can also be applied in more or less the same way within the other formalisms.

i.) COMP_b. This condition is controllable if the input structure Φ_{in} is unfolded. The functional structure of the initial tuple is then an unordered tree Φ_t . Since the input structure is a (homomorphic) extension of the unordered tree ($\Phi_t \sqsubseteq \Phi_{in}$) and both structures have the same path set, the relating homomorphism is an 'onto'-mapping and therefore called *epimorphism*. Part B of fig. 4 gives an example.¹³

Now, since coherence has to be ensured during generation, the derived structure will never become an extension of the input structure and each generation step induces a new epimorphism from the derived structure to the input structure. The coherence condition guarantees that the epimorphisms induced in the generation steps always approximate an isomorphism. When the derived structure and the input structure are isomorphic, all reentrancies are derived.

ii.) COMP_a. The fulfillment of this condition can be controlled, if, apart from the root, all nodes of the DAG introduced by a rule are labeled by a '+'-marker. This additional labeling distinguishes the generator rules from the grammar rules. Fig. 5 shows the generator rule corresponding to the grammar rule of fig. 2. If the root of the unordered tree Φ_t is also +-labeled and all nodes of the structure that is derived from Φ_t are +-labeled, then all paths of the input structure are derived.¹⁴ The condition that all leaves of a well-formed f-structure are labeled by atomic values ensures that all atomic values of the input structure are derived.

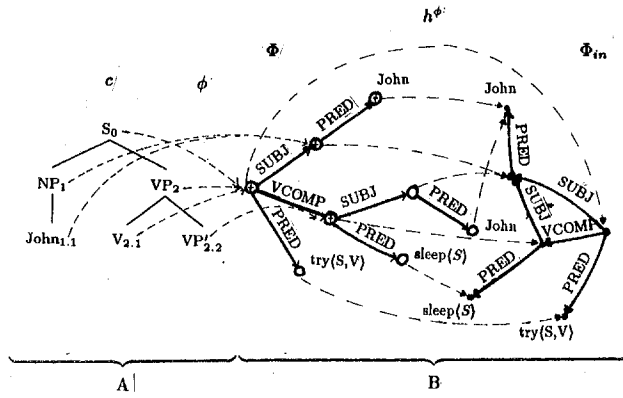


Fig. 4

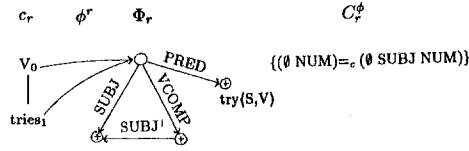


Fig. 5

iii.) COH It is possible to check this condition in each step of the derivation since the input structure is accessible by the epimorphism induced in each particular step. It is guaranteed that no additional information is introduced by the rule application, if the substructure to which the expanded c-structure node is mapped by ϕ and the epimorphism h^ϕ is (aside from the +-labels) an extension of the DAG introduced by the rule.¹⁵

If, for example, the constellation shown in fig. 4 (A and B) is generated and the rule in fig. 6 were to be applied to node 2.2, condition COH will be violated, since the substructure rooted by $h^\phi(\phi_{2.2})$ is not an extension of the (unlabeled) structure introduced by that rule. On the other hand, the substructure rooted by $h^\phi(\phi_{2.1})$ is an extension of the (unlabeled) structure introduced by the V-rule of fig. 5 and fulfills COH with respect to node 2.1.

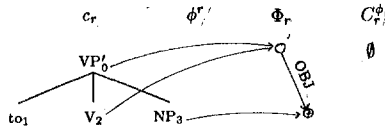


Fig. 6

Since the (functional) structure which is to be derived from Φ_i is equal (or isomorphic) to the input structure itself, it is possible to check

- the f-completeness and f-coherence of the input structure before the generation starts, and
- the constraints expressed by the rules simultaneously during the generation.

Although the V-rule would satisfy the coherence condition with respect to node 2.1, its application is ruled out, since the substructure rooted by $h^\phi(\phi_{2.1})$ does not satisfy the constraint expressed by the rule.¹⁶ Thus, the sequence of tuples which constitute the generation of a terminal string need not contain a constraint set.

The start entity of a generation is then a quintuple

$$\langle c_s, \Phi_i, \phi^s, \Phi_{in}, h_s^\phi \rangle.$$

Φ_{in} is an f-complete and f-coherent input structure, Φ_i is the unfolded input structure and h_s^ϕ is the relating epimorphism. The generability relation Γ_σ is then defined as follows.¹⁷

DEFINITION 2.2 A terminal string s is generable from an input structure Φ_{in} ($\Gamma_\sigma(\Phi_{in}, s)$) iff there is a sequence $\omega_0 \dots \omega_n$ such that

- ◊ $\omega_0 = \langle c_s, \Phi_i, \phi^s, \Phi_{in}, h_s^\phi \rangle$ and
- ◊ for all $\omega_i = \langle c, \Phi, \phi, \Phi_{in}, h^\phi \rangle$, $\omega_{i+1} = \langle c', \Phi', \phi', \Phi_{in}, h^{\phi'} \rangle$ ($0 \leq i < n$) there is a generator rule $V \rightarrow \langle c_r, \Phi_r, \phi_r, C_r^\phi \rangle$ and
 - V is the label of a leaf μ of c
 - ◊ the substructure rooted by $h^\phi(\phi_\mu)$ is (aside from the +-labels) an extension of Φ_r
 - ◊ for all $(j, p) =_c (i, q)$ etc. in C_r^ϕ , $\delta_{in}^*(h^\phi(\phi_{\mu,j}), p) = \delta_{in}^*(h^\phi(\phi_{\mu,i}), q)$ etc.

- c' is the result of expanding μ in c by c_r
- Φ' is the minimal extension of Φ which results from Φ by unifying Φ_r with the substructure rooted by ϕ_μ
- if ν is a node of c , whose ϕ -value is $\delta^*(q_0, p)$ in Φ , then its ϕ' -value in Φ' is $\delta^{**}(q'_0, p)$.
if μ, j is a node of c' , not contained in c , the value of ϕ'_j is $\delta_r^*(q'_0, q)$ and the value of ϕ for μ is $\delta^*(q_0, p)$, then the value of ϕ' for μ, j is $\delta^{**}(q'_0, p, q)$ in Φ' .
- ◊ $\forall p \in L^*(\langle q'_0, p \rangle \in \text{Dom}(\delta^{**}) \rightarrow h^{\phi'}(\delta^{**}(q'_0, p)) = \delta_{in}^*(q'_0, p))$
- ◊ each node of the functional structure Φ_n of ω_n is +-labeled
- ◊ Φ_n is isomorphic to Φ_{in} (h_n^ϕ is an isomorphism)
- s is the terminal string of the c-structure of ω_n .

Lemma 1 follows from the above.

LEMMA 1 $\forall \Phi \in \text{DAG} \forall s \in V_T^*(\Gamma_\sigma(\Phi, s) \leftrightarrow \Delta_\sigma(s, \Phi))$

This lemma can easily be proved, since in each step of the generation of a sentence the applied rule can be applied exactly in the same way in the corresponding derivation step of a derivation of that sentence (and vice versa). So the substructure which includes all +-labeled nodes of a generated functional structure corresponds exactly to that partial f-structure which is derived up to that step (and vice versa). Thus, the derived c-structure is identical to the generated c-structure, the derived f-structure is equal to the generated f-structure and thus identical (isomorphic) to the input structure. Since the constraints in the constraint set of a derivation must in fact be the (instantiated) constraints of all rules applied during the generation, the input structure satisfies all constraints iff the derived structure does.

3 Generation from semantic structures

In this section we use the ideas described in section 2 to develop an algorithm that constructs generators which generate terminal strings from semantic structures. These ideas are applicable if we can ensure, that

- the semantic structures are representable as DAGs,
- the only operation which is used to construct the semantic structures is the unification operation, and
- the semantic structure of a sentence can be built up simultaneously with the derivation of the c- and f-structure of that sentence.

That a) and b) can be ensured for most of the current semantic theories, like Montague Semantics (MS), Discourse Representation Theory (DRT) and Situation Semantics (SS), is illustrated, for example, in [Reyle 88] (MS, DRT), [Halvorsen 87] (SS) and in works concerning categorial grammars which are augmented by a unification component.¹⁸

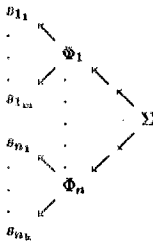
Condition c) is satisfiable if we follow a proposal by Halvorsen who describes a possible extension of LFGs such that the semantic representation of a sentence can be "simultaneously described (co-described) with the functional structure" ([Halvorsen 87], p.9). Halvorsen extends the formalism to include a new type of equation, which is used to build up a semantic representation and establishes an additional (partial) mapping from the f-structure nodes into the node set of the semantic structure.

Since the semantic structures are represented as DAGs, we can use for the generation from semantic structures, a condition like (*) as an adequacy condition, which refers to a semantic (σ -) structure instead of an f-structure.

(**) $\forall \Sigma \in \text{DAG} \forall s \in V_T^*(\Gamma_\sigma(\Sigma, s) \text{ iff } s \text{ is derivable from the start symbol } S \text{ and } s \text{ has } \sigma\text{-structure } \Sigma)$

Since this condition implicitly assumes that Γ_σ^{-1} is determined by an adequate parsing algorithm, this extension of the formalism is neutral with respect to the problem of the creativity¹⁹ of the extension. This neutrality is desirable, since the algorithm should be definable independent of a specific semantic theory. The question whether a semantic component is (or should be) a creative or a conservative extension of the syntactic theory (LFG), on the other hand, depends crucially on the specific semantic theory and on the format of the rules which prescribe how the compositionality principles of the chosen theory are to be translated into this new type of equation.

According to condition (**), the generator constructed by the algorithm for an (extended) LFG is a parser or transducer for well-formed semantic structures which, in principle, constructs for an input structure all sentences which have this structure as its semantic structure, analogously to section 2. A semantic structure alone, however, is usually regarded as too poor with respect to the syntactic information relevant for 'adequate generation results' within a natural language system. We therefore integrate the additional possibility of driving the derivation by syntactic (functional) information. This is possible because the f-structure of a sentence is built up in parallel to the σ -structure driven derivation of that sentence. If we assume that sentences s_i and f-structures Φ_i are related to an input structure Σ according to the following schema:



we can ensure by additional constraints on the f-structures that only those sentences are generable whose f-structures satisfy these constraints. Thus, we can drive the derivation in a way which is more sensitive to syntactic information. We can ensure, for example, that only a passive realization or a realization with a specific topic/focus structure is generable. Since not every input structure has a surface realization with an f-structure that fulfills these additional constraints, these constraints can be highly creative from a formal point of view. Therefore, a generator which drives the derivation by a semantic structure with additional constraints and also satisfies condition (**), must be capable to loosen the f-structure constraints if no output sentence can be derived whose f-structure satisfies the constraints, i.e., to drop constraints successively until an output sentence can be derived. The control of such a dropping procedure is again dependent on the application domain²⁰ and can be determined only by empirical investigations.

3.1 The derivation of semantic structures

According to Halvorsen's proposal, the semantic structures are constructable by means of additional equations which are formulated with a projector σ which "can be prefixed to any expression denoting a functional structure" ([Halvorsen 87], p.8). Fig. 7 A gives a slightly simplified example of a lexical entry from [Halvorsen 87], written as the expansion of a lexical category symbol.

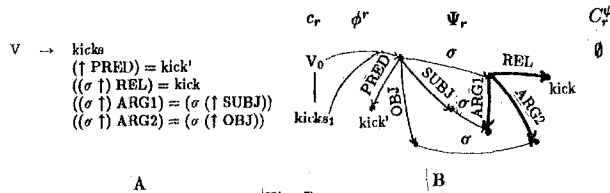


Fig. 7

In Part A the two equations at the bottom are called *inter-modular equations*.

We will first give the syntax of this new type of equation. Let us assume a distinction between functional attributes (L_ϕ) and values (A_ϕ) and semantic attributes (L_σ) and values (A_σ) (with all four sets pairwise disjoint). Then the set of *semantic* (σ -) *designators* contains the *simple* σ -designators A_σ and the *complex* σ -designators. A complex σ -designator is a term $\sigma(l_\phi q)$ consisting of a complex f-designator (a metavariable (\uparrow or \downarrow) followed by a possibly empty sequence of functional attributes), and a possibly empty sequence q of semantic attributes. The σ -equations, then, are those equations which have a complex σ -designator in the first argument position and a complex or simple σ -designator in the second.

If we assume that

- i. the *fds*-algorithm is used to solve the projector equations and to construct the semantic structure, and
- ii. constraints are expressible,²¹

then we can add the following two conditions to the WFF-condition:

(WFF σ)

(WFF) and

5. There is a σ -structure Σ and a partial mapping σ from the f-structure nodes to the nodes of Σ , such that Σ is the only minimal σ -structure that satisfies the projector equations associated with the c-structure nodes. (The *fds*-algorithm constructs also σ and Σ .)
6. All σ -constraints are satisfied by Φ , σ and Σ .²²

These additional conditions must be tested in the order 3. \leq 5. \prec 6..

Since the description-solution algorithm is also used for solving the projector equations, it is easy to simulate the projector-mechanism by using an additional singular attribute. The solution algorithm enforces and preserves the function property (*uniqueness*) of the projector, i.e., if $(\phi_\mu p)$ and $(\phi_\nu p')$ are terms designating some nodes in the f-structure which are mapped by

σ to some nodes in the σ -structure $(\sigma(\phi_\mu p), \sigma(\phi_\nu p'))$, then the identity of $(\phi_\mu p)$ and $(\phi_\nu p')$, which might be established by some other equations, will also enforce the identity of $(\sigma(\phi_\mu p))$ and $(\sigma(\phi_\nu p'))$. Thus, we can simulate the projector by using a singular attribute σ (not contained in L_ϕ , A_ϕ , L_σ , and A_σ) that is inserted between the f-designator and the semantic attributes of a semantic designator according to the following rewrite schemata.

$$(\sigma(\uparrow p) q) \Rightarrow (\uparrow p \sigma q)$$

$$(\sigma(\downarrow p) q) \Rightarrow (\downarrow p \sigma q)$$

The identity of $(\phi_\mu p)$ and $(\phi_\nu p')$ will then also enforce the identity of the values of $(\phi_\mu p \sigma)$ and $(\phi_\nu p' \sigma)$.²³

If we simulate the projector mechanism in this way, the derivation of a σ -structure in parallel to a c- and f-structure closely resembles the derivation process described in section 2.1. A rule introduces a quadruple, where Φ , is replaced by the DAG Ψ , which is obtained by solving the functional and σ -annotations of the local tree introduced by the rule. The constraint set C_Ψ^σ contains the instantiated f- and σ -constraints expressed by that rule.

$$V \rightarrow \langle c_r, \Psi_r, \phi_r, C_r^\sigma \rangle$$

Fig. 7 shows in part B the solution of the equation system for the local tree introduced by the rule of part A. The initial quadruple consists of a DAG Ψ_S which has one σ transition.

$$\langle c_s, \Psi_s, \phi_s, C_s^\sigma \rangle \quad S_0 \xrightarrow{\sigma} \emptyset$$

The only additional condition that has to be satisfied by the derived DAG-structures results from the fact that the semantic substructures of a derived DAG are not necessarily *connected*, i.e. that not every substructure which is a value of a σ -attribute must necessarily be a substructure of the topmost σ -attribute of Ψ . On the one hand, the syntax for the σ -equations permits us to formulate rules which introduce unconnected semantic structures (fig. 8.1 gives an example). On the other hand, rules can introduce grammatical functions without a σ -attribute (cf. fig. 8.2 and 8.3). This can lead to unconnected semantic structures, if the expansion of a c-structure node which is associated with such a grammatical function introduces a semantic structure. The motivation for not excluding these two sources for the unconnectedness is that the *semantic function* of a constituent or grammatical function can be uncertain within the local context given by a rule and has to be determined by another constituent not introduced in that rule.²⁴ In traditional LFG it is usually assumed that the semantic function of subcategorized constituents such as SUBJ and OBJ is determined by the governing verb (PRED) and that the assignment of grammatical functions to semantic functions may be affected by lexical rules. These assignments can be established by inter-modular equations in the lexical entry for the verb, as illustrated in fig. 7, while at the same time leaving open the assignments of semantic functions to the NPs in the rules of fig. 8.2 and 8.3 by not annotating them with σ -equations.

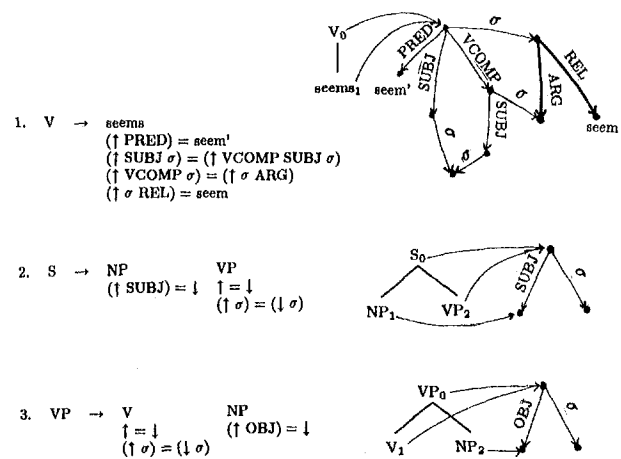


Fig. 8

Thus, unconnected semantic structures can become connected through inter-modular equations which take into account the semantic structure of those constituents whose semantic function is not determined by the context represented by the local tree of a rule.

Since the DAG of a derived sentence has to contain a connected semantic structure, we have to add to the well-formedness conditions of that DAG, that the value of each σ -transition must be accessible from the σ -value of the

root of that DAG. The derived semantic structure is the substructure of the derived DAG which is rooted by the value of the uppermost σ -transition. This leads immediately to the definition of the derivability relation.

DEFINITION 3.1 A terminal string s is *derivable* with σ -structure Σ ($\Delta_\sigma(s, \Sigma)$) iff there is a sequence $\omega_0 \dots \omega_n$ such that

- ◊ $\omega_0 = \langle c_S, \Psi_S, \phi^S, C_S^\psi \rangle$ and
- ◊ for all $\omega_i = \langle c, \Psi, \phi, C^\psi \rangle$, $\omega_{i+1} = \langle c', \Psi', \phi', C'^\psi \rangle$ ($0 \leq i < n$) there is a rule $V \rightarrow \langle c_r, \Psi_r, \phi^r, C_r^\psi \rangle$ and
 - V is a label of a leaf μ of c
 - c' is the result of expanding μ in c by c_r
 - Ψ' is the minimal extension of Ψ which results from Ψ by unifying Ψ_r with the substructure rooted by ϕ_μ
 - if ν is a node of c , whose ϕ -value is $\delta^*(q_0, p)$ in Ψ , then its ϕ' -value in Ψ' is $\delta^*(q'_0, p)$
 - if $\mu.j$ is a node of c' , not contained in c , the value of ϕ'_j is $\delta^*(q'_0, q)$ and the value of ϕ for μ is $\delta^*(q_0, p)$, then the value of ϕ' for $\mu.j$ is $\delta^*(q'_0, p.q)$ in Ψ'
 - C'^ψ contains besides C^ψ the constraints $(\mu.j \ p) =_c (\mu.i \ q)$ etc. for all $(j \ p) =_c (i \ q)$ etc. in C_r^ψ
- Ψ_n satisfies all constraints in C_n^ψ
- Ψ_n is f-complete and f-coherent
- ◊ $\forall p \in L_\sigma^*(\langle q_0^0, p, \sigma \rangle \in \text{Dom}(\delta_n^*)) \rightarrow \exists q \in L_\sigma^*(\delta_n^*(q_0^0, \sigma, q) = \delta_n^*(q_0^0, p, \sigma))$ (connectedness)
- ◊ Σ is the substructure of Ψ_n rooted by $\delta_n(q_0^0, \sigma)$
- s is the terminal string of the c-structure of ω_n .

3.2 Generation as semantic structure driven and constrained derivation

The mechanisms for controlling *completeness* and *coherence*, which have been developed in section 2, together with the defining conditions of the different types of derivation concepts can now be used to design an adequate algorithm for the generation from semantic structures, which in addition can be directed by functional constraints.

The process of generating from a semantic input structure Σ_{in} has to start with a complex entity

$$\langle c_S, \Psi_t, \phi^S, \Sigma_{in}, h_S^\sigma, C_S^\psi \rangle$$

where Ψ_t is an unordered tree consisting of only one σ -transition, whose value is \downarrow -labeled and dominates the unfolded input structure Σ_{in} . Fig. 9 shows the start entity for the generation from a simplified semantic input structure. h_S^σ is the induced epimorphism from the substructure rooted by the σ -value of Ψ_t onto Σ_{in} . The start entity contains an empty set of constraints, since not all constraints expressed by the applied rules can be checked immediately (e.g. pure functional constraints) and have to be added to this set until they can be checked.

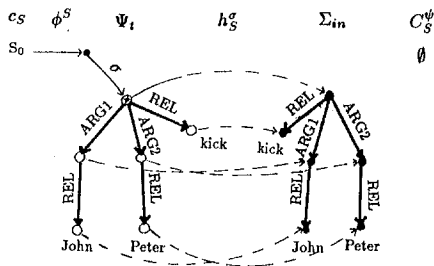


Fig. 9

The derivation can be constrained by a set $C^{\phi \oplus \sigma}$ of pure functional and/or inter-modular constraints of the form

$$(\emptyset \ p) =_c (\emptyset \ p') \text{ etc. and } (\emptyset \ \sigma \ q) =_c (\emptyset \ p \ \sigma)$$

which enforce the derivation of those sentences, whose f- and σ -structures fulfill these constraints. (These constraints must eventually be retracted if no sentence is generable.) E.g. the set

$$C_1^{\phi \oplus \sigma} = \left\{ \begin{array}{l} (\emptyset \ \text{SUBJ} \ \sigma) =_c (\emptyset \ \sigma \ \text{ARG2}) \\ (\emptyset \ \text{BYOBJ} \ \sigma) =_c (\emptyset \ \sigma \ \text{ARG1}) \end{array} \right\}$$

will enforce a passive realization of the input structure given in fig. 9 ('Peter is kicked by John'), while the set

$$C_2^{\phi \oplus \sigma} = \left\{ \begin{array}{l} (\emptyset \ \text{SUBJ} \ \sigma) =_c (\emptyset \ \sigma \ \text{ARG2}) \\ (\emptyset \ \text{BYOBJ} \ \sigma) =_c (\emptyset \ \sigma \ \text{ARG1}) \\ (\emptyset \ \text{BYOBJ} \ \sigma) =_c (\emptyset \ \text{TOPIC}) \end{array} \right\}$$

enforces a passive realization with the BYOBJ in the TOPIC-position ('By John, Peter is kicked').

The generator rules are constructed analogously to section 2 by an additional \downarrow -labeling of all nodes which are values of semantic attributes (not of the σ -attribute!) within the DAG-structure introduced by the rule. Fig. 10 gives an example.

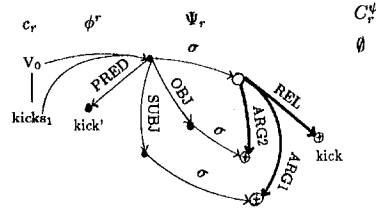


Fig. 10

Again a crucial restriction follows from the fact that these rules permit the derivation of structures with an unconnected semantic substructure, which becomes connected in later steps of the derivation by means of inter-modular equations. If a constellation is derived by means of these rules, as it is given schematically in fig. 11,

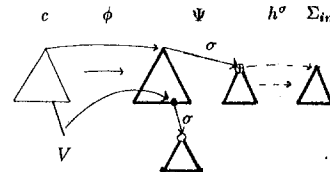


Fig. 11

the generation by expansion of V proceeds without control by the input structure as long as the semantic structure introduced by V remains unaccessible. The *coherence*-condition is undermined and sentences can be generated whose semantic structure is not subsumed by the input structure.

The *coherence*-condition can be checked, however, if only those rules are applied which preserve the connectedness of the semantic structure derived so far. This restriction on the order-freeness of the rule application does not affect the adequacy condition (**), because in each derivation step of a sentence with a connected semantic structure there must be at least one node of the partial c-structure which is expandable in such a way that the expansion preserves the connectedness of the partial semantic structure. If the expansion of every terminal node violated this connectedness condition, the semantic structure of that sentence could not be connected. Hence, at least one rule must be applicable in such a way.

Since it is possible that pure semantic constraints of the rule's constraint set refer to semantic substructures whose semantic function is uncertain up to that generation step, not all pure semantic constraints expressed by the rules can be tested immediately. This applies to pure semantic constraints which are constructable from inter-modular constraints by conversion. These constraints contain terms of the form $(i \ \sigma \ p)$ (which correspond to $(\downarrow \ \sigma \ p)$ annotations) and they are not testable immediately if i is the value for $(\emptyset \ q)$ (q non-empty) (which corresponds to a $(\uparrow \ q) = \downarrow$ annotation) and $\phi_\mu(q)$ has no σ -attribute. Hence, the pure semantic constraints which cannot be tested in a particular step, the pure functional constraints, and the inter-modular constraints of the rule's constraint set have to be collected in C^ψ . The (*constraint-driven*) *generability* relation has to then be defined as follows.

DEFINITION 3.2 A terminal string s is *generable* from an input structure Σ_{in} under a constraint set $C^{\phi \oplus \sigma}$ ($\Gamma_\sigma^c(\Sigma_{in}, s, C^{\phi \oplus \sigma})$) iff there is a sequence $\omega_0 \dots \omega_n$ such that

- ◊ $\omega_0 = \langle c_S, \Psi_t, \phi^S, \Sigma_{in}, h_S^\sigma, C_S^\psi \rangle$ and
- ◊ for all $\omega_i = \langle c, \Psi, \phi, \Sigma_{in}, h^\sigma, C^\psi \rangle$, $\omega_{i+1} = \langle c', \Psi', \phi', \Sigma_{in}, h^{\sigma'}, C'^\psi \rangle$ ($0 \leq i < n$) there is a generator rule $V \rightarrow \langle c_r, \Psi_r, \phi^r, C_r^\psi \rangle$ and
 - V is a label of a leaf μ of c
 - ◊ the substructure rooted by $h^\sigma(\phi_\mu, \sigma)$ is (apart from the \downarrow -labels) an extension of Σ_r (the substructure of Ψ_r rooted by $\delta_r(q_0^0, \sigma)$)
 - ◊ for all pure semantic constraints $(j \ \sigma \ p) =_c (i \ \sigma \ q)$ etc. in C_r^ψ for which $\delta(\phi_{\mu,j}, \sigma)$ and $\delta(\phi_{\nu,i}, \sigma)$ is defined, $\delta_{in}^*(h^\sigma(\delta(\phi_{\mu,j}, \sigma)), p) = \delta_{in}^*(h^{\sigma'}(\delta(\phi_{\nu,i}, \sigma)), q)$ etc.

- c' is the result of expanding μ in c by c_r .
- Ψ' is the minimal extension of Ψ which results from Ψ by unifying Ψ_r with the substructure rooted by ϕ_μ
- ◇ $\forall p \in L_\phi^*(q_0, p, \sigma) \in \text{Dom}(\delta^*) \rightarrow \exists q \in L_\sigma^*(q_0', \sigma, q) = \delta^*(q_0', p, \sigma)$
- if ν is a node of c , whose ϕ -value is $\delta^*(q_0, p)$ in Ψ , then its ϕ' -value in Ψ' is $\delta^*(q_0', p)$.
- if μ, j is a node of c' , not contained in c , the value of ϕ_j^r is $\delta_r^*(q_0', q)$ and the value of ϕ for μ is $\delta^*(q_0, p)$, then the value of ϕ' for μ, j is $\delta^*(q_0', p, q)$ in Ψ' .
- ◇ $C^{\psi'}$ contains besides C^ψ the pure semantic constraints which cannot be tested in this step, the pure functional and the inter-modular constraints of C_r^ψ with μ attached as a prefix to the node indices of the constraints
- ◇ $\forall p \in L_\sigma^*(q_0', \sigma, p) \in \text{Dom}(\delta^*) \rightarrow h^{\sigma'}(\delta^*(q_0', \sigma, p)) = \delta_{in}^n(q_0^n, p)$
- ◇ each node of the substructure of Ψ_n rooted by $\delta_n(q_0^n, \sigma)$ (the semantic structure) is $+$ -labeled
- ◇ the substructure of Ψ_n rooted by $\delta_n(q_0^n, \sigma)$ is isomorphic to Σ_{in} (h_n^σ is an isomorphism)
- Ψ_n is f-complete and f-coherent
- Ψ_n satisfies all constraints in C_n^ψ
- ◇ Ψ_n satisfies all constraints in $C^{\psi \oplus \sigma}$
- s is the terminal string of the c-structure of ω_n .

If we define the (unconstrained) generability relation Γ_σ as

$$\Gamma_\sigma = \{ \langle \Sigma, s \rangle \mid \langle \Sigma, s, \theta \rangle \in \Gamma_\sigma^* \}$$

we can prove

$$\text{LEMMA 2 } \forall \Sigma \in \text{DAG } \forall s \in V_T^*(\Gamma_\sigma(\Sigma, s) \leftrightarrow \Delta_\sigma(s, \Sigma))$$

The proof from left to right is carried out in analogy to Lemma 1. The right-to-left half follows from the fact that the order of the rule application during the derivation of a sentence with a connected semantic structure can always be rearranged in such a way that in each step of the derivation the partial semantic structure is connected. Otherwise the derived sentence would not have a connected semantic structure.

4 Acknowledgements

The research reported in this paper was partially supported by the EUROTRA-D accompanying research (BMFT grant No. 101 3207 0) and the ESPRIT project ACORD (project P393). The author would like to thank Mark Johnson, Bob Kasper, Stefan Momma, Klaus Netter, Klaus Reinhardt and the members of the working group on Unification at the University of Stuttgart (Inge Bethke, Jochen Dörre, Andreas Eisele and Henk Zeevat) who read carefully through earlier versions of this paper and helped a lot in improving both the algorithm and its presentation. All remaining errors are of course my own.

Footnotes

¹For parsing cf. e.g. [Eisele/Dörre 86].

²For the description of dialogue structures within LFG cf. [Netter 88].

³The algorithm described here only works for those grammars which do not have regular expressions in the annotations (cf. [Kaplan/Maxwell/Zaenen 86]).

⁴The algorithm has been implemented by Jochen Dörre and Stefan Momma, cf. [Momma/Dörre 87].

⁵This condition can be weakened in different ways (cf. fn.15). However, since it depends on the domain of application how this is done in a particular case, and since we want to describe the algorithm independently of specific applications, we submit it to this strong condition.

⁶Detailed information on solution algorithms can be found in [Johnson 86] and [Johnson 87].

⁷I was reminded by an anonymous reviewer to relate this paper to a paper by Block ([Block 86]), who must have had this procedure in mind, when he claimed that there is no way to derive c-structures from f-structures.

"On the other hand, an LFG grammar, which does not allow transformations is forced to generate on two different levels f-structure and c-structure and then to compare the two, looking for a match. This is the case because the theory specifically forbids carrying out any operations on either f- or c-structures (cf. Kaplan and Bresnan 1982:180). Thus, there is no way to derive c-structures from f-structures." ([Block 86], p.3)

I will attempt to clarify this issue in this paper by 'unforbidden' means.

⁸In fact, functional structures are commonly regarded as equivalence classes of isomorphic transition graphs. We can use representatives of these classes within the definitions without loss of generality. Furthermore, all leaves have to be labeled by m , if a functional structure is well-formed.

⁹This prefixing procedure runs analogously for all other types of constraints. In LFG there are constraints of the form $(\nu p) = c$ (μq), (νp) , $(\nu p) = c$ and their negations

(μ and ν are node indices, q and p are sequences of attributes and z is an atomic value). The satisfaction relation between DAGs and these types of constraints is usually defined as:

$$\Phi \models (\nu p) = c \text{ (} \mu q \text{)} \leftrightarrow \delta^*(\phi_{\nu, p}) = \delta^*(\phi_{\mu, q})$$

$$\Phi \models (\nu p) \leftrightarrow \delta^* \text{ is defined for } \langle \phi_{\nu, p} \rangle$$

$$\Phi \models (\nu p) = c \leftrightarrow m(\delta^*(\phi_{\nu, p})) = z$$

For the negative constraints the satisfaction relation is defined by the negated conditions. For the sake of brevity, I will refer to only one type in the following, whenever constraints are involved.

¹⁰The relation that is specified by the WFF-condition between terminal strings and DAGs is equal to Δ_ϕ . Both procedures are equivalent, since the order which is given by the respective sequence of applications of rules is imposed onto the fds-algorithm, which otherwise is orderfree.

¹¹It should be noted that these conditions are different from the conditions that an f-structure must be complete and coherent. To avoid confusion the latter ones are called f-complete and f-coherent in the following.

¹²The completeness constraints can be made explicit by the path set (COMP_a), existential constraints) and the elements of the irreflexive subset of the Nerode relation over the path set (COMP_b, reentrancy constraints), cf. [Kasper/Rounds 86] and [Kasper 87].

¹³The purpose of the $+$ -labels will be explained below.

¹⁴This presupposes that the unification operation preserves labels, but this was already the case for the atomic values.

¹⁵Within the actual implementation the coherence condition can be weakened. Atomic-valued features can be selected which are excluded from the coherence check and thus can be added during the generation process. Again, it has to be determined in a language specific way which features are entirely predictable either from grammatical rules (e.g. structural case in German, which under a specific analysis could be covered by a structural rule associating nominative with SUBJ and accusative with OBJ) or from the lexical entries associated with certain predicates (e.g. grammatical gender in German which has to be specified for every noun).

¹⁶But that is only because the input structure is slimmed down for reasons of clarity.

¹⁷Defining conditions which did not occur in previous definitions are itemized by σ .

¹⁸For categorial unification grammars (CUG) cf. [Uszkoreit 86] and for unification categorial grammars (UCG) cf. [Zeevat/Klein/Calder 87].

¹⁹The extension of a syntactic theory by a semantic theory is called creative iff there are grammars where not all grammatically well-formed sentences are also semantically well-formed; otherwise the extension is called conservative.

²⁰Within a machine translation system this dropping of constraints could be controlled in such a way that a maximal similarity with respect to the stylistic features of the f-structure of the source language sentence has to be ensured. What 'maximal similarity with respect to the stylistic features' means again presupposes some empirical work.

²¹One possible additional condition would be to require that each c-structure constituent must be interpreted within the semantic structure, i.e. each c-structure node must be mapped by $\phi \circ \sigma$ (the product) into the semantic structure. Since we wanted to formulate the algorithm as general as possible, we dispense with this condition.

²²Since it is possible to express constraints referring to the mapping σ between Φ and Σ (inter-modular constraints, for example), the reference to Φ , σ and Σ is necessary.

²³To separate the f- and σ -structure from each other we only have to eliminate the σ -transitions.

²⁴The semantic function of a grammatical function is identified with the semantic attribute, whose value corresponds to the σ -value of that grammatical function.

References

- [Block 86] Block, R.: Can a "Parsing Grammar" Be Used for Natural Language Generation? The Negative Example of LFG. Ms. Hamburg 1986
- [Eisele/Dörre 86] Eisele, A., Dörre, J.: A Lexical Functional Grammar System in Prolog. In: *Proceedings of COLING 86*, Bonn 1986
- [Halvorsen 87] Halvorsen, P.-K.: Situation Semantics and Semantic Interpretation in Constraint-Based Grammars. CSLI Report No. CSLI-87-101, Stanford 1987
- [Johnson 86] Johnson, M.: Computing with Regular Path Formula. Ms. Stanford 1986
- [Johnson 87] Johnson, M.: Attribute-Value Logic and the Theory of Grammar. Ph.D. thesis, Stanford University, Stanford 1987
- [Kaplan/Bresnan 82] Kaplan, R., Bresnan, J.: Lexical Functional Grammar: A Formal System for Grammatical Representation. In: J. Bresnan (ed.): *The Mental Representation of Grammatical Relations*. Cambridge, Mass. 1982
- [Kaplan/Maxwell/Zaenen 86] Kaplan, R., Maxwell, J., Zaenen, A.: Functional Uncertainty. CSLI monthly vol. 2, no. 4, Stanford 1986
- [Kaplan 87] Kaplan, R.: Three Seductions of Computational Psycholinguistics. in: Whitelock, P. et al. (eds.): *Linguistic Theory and Computer Applications*, New York 1987
- [Karttunen 86] Karttunen, L.: D-PATR: A Development Environment for Unification-Based Grammars. In: *Proceedings of COLING 86*, Bonn 1986
- [Kasper 87] Kasper, R.T.: Feature Structures: A Logical Theory with Application to Language Analysis. Ph.D. thesis. University of Michigan, Ann Arbor 1987
- [Kasper/Rounds 86] Kasper, R.T., Rounds, W.: A Logical Semantics for Feature Structures. In: *Proceedings of the 24th Annual Meeting of the ACL*, New York 1986
- [Momma/Dörre 87] Momma, S., Dörre, J.: Generation from F-Structures. In: Klein, E., van Benthem, J. (eds.): *Categories, Polymorphism and Unification*, Edinburgh, Amsterdam 1987
- [Netter 88] Netter, K.: Syntactic Aspects of LFG-Based Dialogue Parsing. ESPRIT ACORD Project 393, Deliverable T2.7(a), Stuttgart 1988
- [Reyle 88] Reyle, U.: Compositional Semantics for LFG. In: Reyle, U., Rohrer, C. (eds.): *Natural Language Parsing and Linguistic Theories*. Dordrecht 1988
- [Shieber 83] Shieber, S. et al.: The Formalism and Implementation of PATR-II. In: Grosz, B., Stickel, M. (eds.): *Research on Interactive Acquisition and Use of Knowledge*, SRI International, Menlo Park 1983
- [Uszkoreit 86] Uszkoreit, H.: Categorial Unification Grammars. In: *Proceedings of COLING 86*, Bonn 1986
- [Wedekind 86] Wedekind, J.: A Concept of Derivation for LFG. In: *Proceedings of COLING 86*, Bonn 1986
- [Zeevat/Klein/Calder 87] Zeevat, H., Klein, E., Calder, J.: An Introduction to Unification Categorial Grammar. In: Haddock, N. et al. (eds.): *Edinburgh Working Papers in Cognitive Science*, Edinburgh 1987