

Constituent Coordination in Lexical-Functional Grammar*

Ronald M. Kaplan and John T. Maxwell III

Xerox Palo Alto Research Center

3333 Coyote Hill Road

Palo Alto, California 94304 USA

Abstract: This paper outlines a theory of constituent coordination for Lexical-Functional Grammar. On this theory LFG's flat, unstructured sets are used as the functional representation of coordinate constructions. Function-application is extended to sets by treating a set formally as the generalization of its functional elements. This causes properties attributed externally to a coordinate structure to be uniformly distributed across its elements, without requiring additional grammatical specifications.

Introduction

A proper treatment of coordination has long been an elusive goal of both theoretical and computational approaches to language. The original transformational formulation in terms of the Coordinate Reduction rule (e.g. Dougherty, 1970) was quickly shown to have many theoretical and empirical inadequacies, and only recently have linguistic theories (e.g. GPSG (Gazdar et al., 1985), Categorical grammar (e.g. Steedman, 1985)) made substantial progress on characterizing the complex restrictions on coordinate constructions and also on their semantic interpretations. Coordination has also presented descriptive problems for computational approaches. Typically these have been solved by special devices that are added to the parsing algorithms to analyze coordinate constructions that cannot easily be characterized in explicit rules of grammar. The best known examples of this kind of approach are SYSCONJ (Woods, 1973), LSP (Sager, 1981), and MSG (Dahl and McCord, 1983).

Coordination phenomena are usually divided into two classes, the so-called constituent coordinations where the coordinated elements look like otherwise well-motivated phrasal constituents (1), and nonconstituent coordination where the coordinated elements look like fragments of phrasal constituents (2).

- (1) (a) A girl saw Mary and ran to Bill. (Coordinated verb phrases)
- (b) A girl saw and heard Mary. (Coordinated verbs)
- (2) Bill went to Chicago on Wednesday and New York on Thursday.

Of course, what is or is not a well-motivated constituent depends on the details of the particular grammatical theory. Constituents in transformationally-oriented theories, for example, are units that simplify the feeding relations of transformational rules, whereas "constituents" in categorial grammars merely reflect the order of binary combinations and have no other special motivation. In lexical-functional grammar, surface constituents are taken to be the units of phonological interpretation. These may differ markedly from the units of functional or semantic interpretation, as shown in the analysis of Dutch cross serial dependencies given by Bresnan et al. (1982).

Nonconstituent coordination, of course, presents a wide variety of complex and difficult descriptive problems, but constituent coordination also raises important linguistic issues. It is the latter that we focus on in this brief paper.

To a first approximation, constituent coordinations can be analyzed as the result of taking two independent clauses and factoring out their common subparts. The verb coordination in (1b) is thus related to the fuller sentence coordination in (3). This intuition, which was the basis of the Coordinate Reduction Transformation, accounts for more complex patterns of acceptability such as (4) illustrates. The coordination in (4c) is acceptable because both (4a) and (4b) are, while (4e) is bad because of the independent subcategorization violation in (4d).

- (3) A girl saw Mary and a girl heard Mary.
- (4) (a) A girl dedicated a pie to Bill.
- (b) A girl gave a pie to Bill.

*An earlier version of this paper appeared in the Proceedings of COLING 88, Budapest, August 1988, pp. 303-305.

- (c) A girl dedicated and gave a pie to Bill.
- (d) *A girl ate a pie to Bill.
- (e) *A girl dedicated and ate a pie to Bill.

This first approximation is fraught with difficulties. It ensures that constituents of like categories can be conjoined only if they share some finer details of specification, but there are more subtle conditions that it does not cover. For example, even though (5a) and (5b) are both independently grammatical, the coordination in (5c) is unacceptable:

- (5) (a) The girl promised John to go.
 - (b) The girl persuaded John to go.
 - (c) *The girl promised and persuaded John to go.
- (Hint: Who is going?)

Another well-known difficulty with this approach is that it does not obviously allow for the necessary semantic distinctions to be made, on the assumption that the semantic properties of reduced coordinations are to be explicated in terms of the semantic representations of the propositional coordinations that they are related to. This is illustrated by the contrasting semantic entailments in (6): Sentence (6a) allows for the possibility that two different girls are involved while (6b) implies that a single (but indefinite) girl performed both actions.

- (6) (a) A girl saw Mary and a girl talked to Bill.
- (b) A girl saw Mary and talked to Bill.

Despite its deficiencies, it has not been easy to find a satisfactory alternative to this first approximation. The theoretical challenge is to embed coordination in a grammatical system in a way that is independent of the other generalizations that are being expressed (e.g. actives correspond to passives, NP's in English can be followed by relative clauses, English relative clauses look like S's with a missing NP) but which interacts with those specifications in just the right ways. That is, a possible but unacceptable solution to this descriptive dilemma would be to add to the grammar new versions of all the basic rules designed specifically to account for the vagaries of coordination.

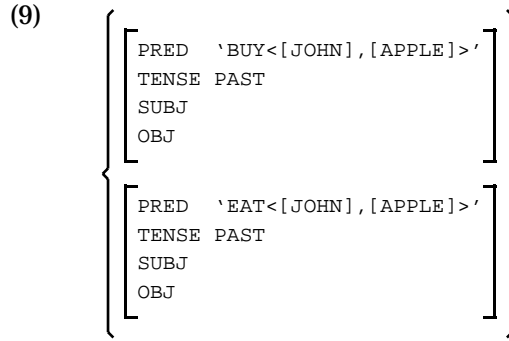
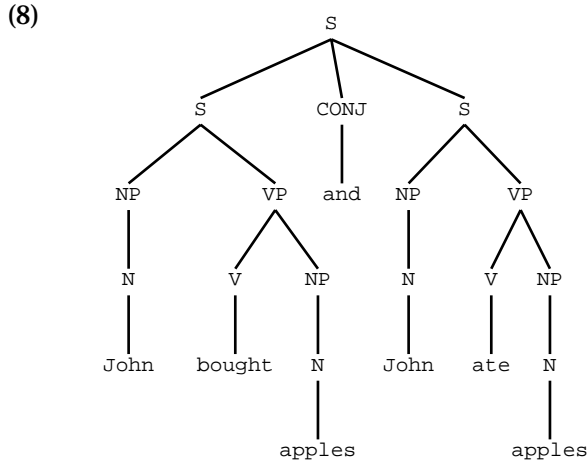
Coordination was not discussed in the original formulation of Lexical-Functional Grammar (Kaplan and Bresnan, 1982), although mathematical objects (finite sets of f-structures) were introduced to provide an underlying representation for grammatical constructions which, like the parts of a coordination, do not seem to obey the uniqueness condition that normally applies to grammatical functions and features. Adjuncts and other modifying constructions are the major example of this that Kaplan and Bresnan discussed, but they also suggested that the same mathematical representations might also be used in the analysis of coordination phenomena. In the present paper we extend the LFG formalism to provide a simple account of coordination that follows along the general lines of the Kaplan/Bresnan suggestion and does not involve detailed specifications of the coordination properties of particular constituents. We illustrate the consequences of this extension by discussing a small number of grammatical constructions; Bresnan, Kaplan, and Peterson (forthcoming) discuss a much wider range of phenomena and provide more general linguistic motivation for this approach.

Simple Coordination

A lexical-functional grammar assigns two syntactic levels of representation to each grammatical string in a language. The constituent structure, or c-structure, is a conventional tree that indicates the organization of surface words and phrases, while the functional structure (f-structure) is a hierarchy of attributes and values that represents the grammatical functions and features of the sentence. LFG assumes as a basic axiom that there is a piecewise function, called a structural correspondence or "projection", that maps from the nodes in the c-structure to the units in an abstract f-structure (see Kaplan and Bresnan (1982) and Kaplan (1987) for details). This means that the properties of the f-structure can be specified in terms of the mother-daughter and precedence relations in the c-structure, even though the f-structure is formally not at all a tree-like structure.

Now let us consider a simple example of coordination wherein two sentences are conjoined together (7). A plausible c-structure for this sentence is given in (8), and we propose (9) to represent the functional properties of this sentence.

- (7) John bought apples and John ate apples.



The structure in (9) is a set containing the f-structures that correspond to the component sentences of the coordination. As Bresnan, Kaplan, and Peterson (forthcoming) observe, sets constitute a plausible formal representation for coordination since an unlimited number of items can be conjoined in a single construction and none of those items dominates or has scope over the others. Neither particular functional attributes nor recursive embeddings of attributes can provide the appropriate representation that flat, unstructured sets allow.

To obtain the representation of coordination shown in (8) and (9), all we need is the following alternative way of expanding S:

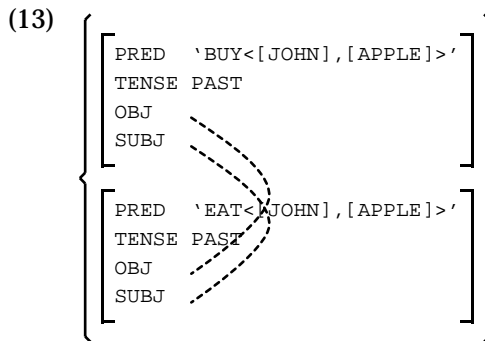
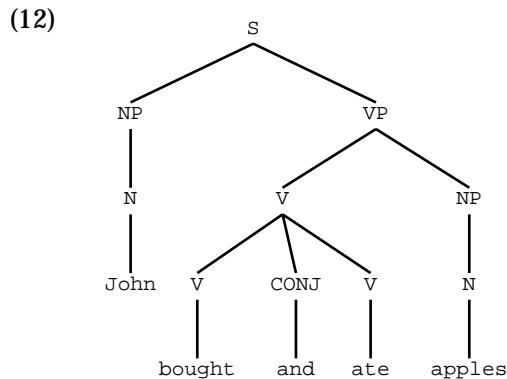
$$(10) \quad S \rightarrow \begin{array}{ccc} S & \text{CONJ} & S \\ \downarrow \in \uparrow & & \downarrow \in \uparrow \end{array}$$

This rule says that a conjoined sentence consists of a sentence followed by a conjunction followed by another sentence, where the f-structures of each sub-sentence is an element of the f-structure that represents their coordination.

Coordination with Distribution

The next step is to consider constituent coordinations where some parts of the sentence are shared by the coordinated constituents. Consider the following sentence:

(11) John bought and ate apples.



The desired c-structure and f-structure for (11) are shown in (12) and (13) respectively. Notice that the subjects and objects of BUY and EAT are linked, so that the f-structure is different from the one in (9) for *John bought apples and John ate apples*. The identity links in this structure account for the different semantic entailments of sentences (7) and (11) as well as for the differences in (6a) and (6b).

This is an example of verb coordination, so the following alternative is added to the grammar:

$$(14) \quad V \rightarrow \begin{array}{ccc} V & \text{CONJ} & V \\ \downarrow \in \uparrow & & \downarrow \in \uparrow \end{array}$$

This rule permits the appropriate c-structure configuration but its functional specifications are no different than the ones for simple sentential coordination. How then do the links in (13) arise? The basic descriptive device of the LFG formalism is the function application expression:

$$(15) \quad (f a) = v$$

As originally formulated by Kaplan and Bresnan (1982), this equation (15) holds if and only if f denotes an f-structure which yields the value v when applied to the attribute a . According to the original definition, the value of an application expression is undefined when f denotes a set of f-structures instead of a single function and an equation such as (15) would therefore be false. Following Bresnan, Kaplan, and Peterson (forthcoming), we propose extending the function-application device so that it is defined for sets of functions. If s denotes a set of functions, we say that $(s a)=v$ holds if and only if v is the *generalization* of all the elements of s applied to a :

$$(16) \quad (s a) = \prod (f a) \text{ if } (f a) \text{ is defined for all } f \in s.$$

The generalization $f_1 \prod f_2$ of two functions or f-structures f_1 and f_2 is defined recursively as follows:

$$(17) \quad \text{If } f_1=f_2 \text{ then } f_1 \prod f_2 = f_1.$$

$$\text{If } f_1 \text{ and } f_2 \text{ are f-structures, then } f_1 \prod f_2 = \{ \langle a, (f_1 a) \prod (f_2 a) \rangle \mid a \in \text{DOM}(f_1) \cap \text{DOM}(f_2) \}.$$

$$\text{Otherwise, } f_1 \prod f_2 = \perp.$$

The generalization is the greatest lower bound in the subsumption ordering on the f-structure lattice. The symbol \perp denotes the bottom-most element in this lattice, the element that subsumes all other elements and about which no information is known. With this element explicitly represented, we simplify the determinacy and completeness conditions of Kaplan and Bresnan (1982) by defining minimal solutions containing \perp to be invalid.

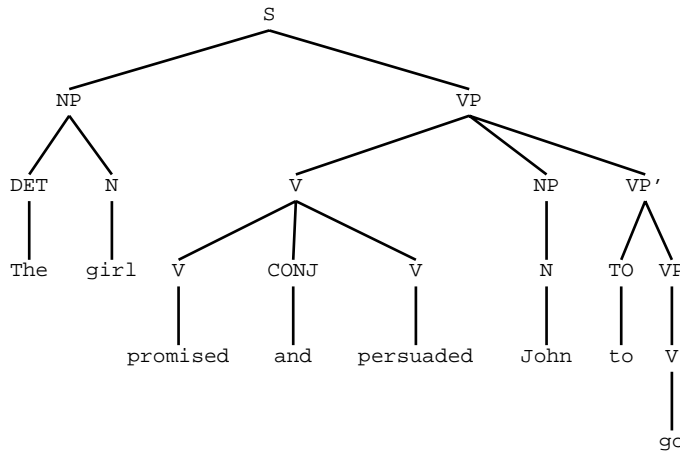
These definitions have two consequences. The first is that v subsumes $(f a)$ for all $f \in s$. Thus the properties asserted on a set as a whole must be distributed across the elements of the set. This explains why the subject and object of (11) are distributed across both verbs without having to change the VP rule in (18) below. The equation on the object NP of (18) is $(\uparrow \text{OBJ}) = \downarrow$. The meta-variable \uparrow denotes a set because the f-structure of the VP node is the same as the f-structure of the conjoined V node, which by (14) is a set. Therefore the effect of rule (18) is that each of the elements of the \uparrow set will have an OBJ attribute whose value is subsumed by the f-structure corresponding to *apples*.

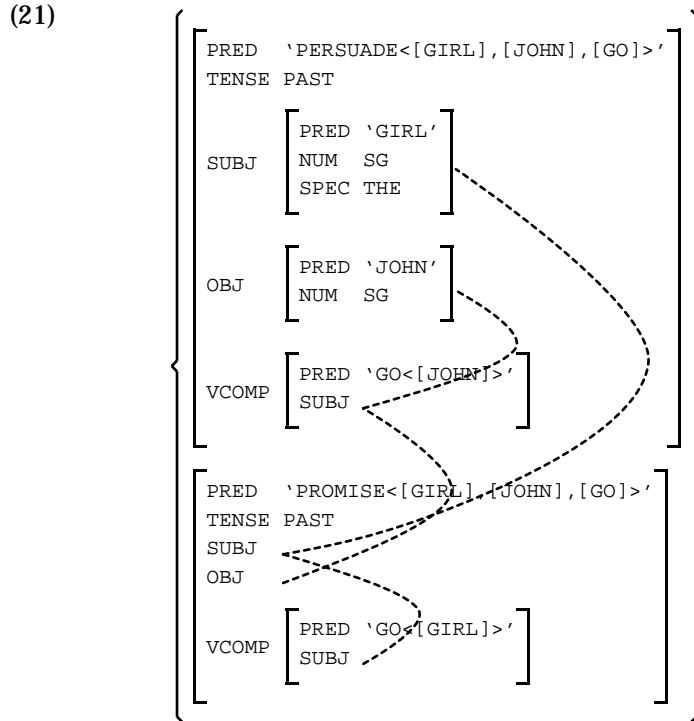
$$(18) \quad \text{VP} \rightarrow \begin{array}{cc} \text{V} & \text{NP} \\ \downarrow = \uparrow & (\uparrow \text{OBJ}) = \downarrow \end{array}$$

The second consequence of (16) is that v takes on the attributes and values that all of the $(f a)$ have in common. This is useful in explaining the ungrammaticality of the *promise and persuade* sentence in (4). (We are indebted to Andreas Eisele and Stefan Momma for calling our attention to this example.) The analysis for this sentence is in (20) and (21):

(19) *The girl promised and persuaded John to go.

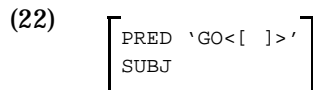
(20)





At first glance, (21) seems to provide a perfectly reasonable analysis of (19). PROMISE and PERSUADE share an object, a subject, and a verb complement. The verb complements have different subjects as a result of the different control equations for PROMISE and PERSUADE (The lexical entry for PROMISE specifies subject control (\uparrow VCOMP SUBJ) = (\uparrow SUBJ), while PERSUADE specifies object control (\uparrow VCOMP SUBJ) = (\uparrow OBJ)). There is no inconsistency, incompleteness or incoherence in this structure.

However, in LFG the completeness conditions apply to the f-structures mapped from all the c-structure nodes, whether or not they are part of the structure corresponding to the root node. And if we look at the f-structure that corresponds to the verb-complement node, we discover that it contains \perp and thus is incomplete:



This f-structure is the generalization of (s VCOMP) for the set given in (21). Everything that the two VCOMPs have in common is given by this f-structure. However, the subject of the f-structure has \perp for its predicate. It thus violates the "semantic completeness" condition of LFG, which in essence requires that something must be known about the predicate of every thematic function. If the VCOMPs had had a subject in common (as in the sentence *The girl urged and persuaded John to go*) then the sentence would have been perfectly legal.

Interactions with Long-Distance Dependencies

Under certain circumstances a shared constituent plays different roles in the conjoined constituents. For instance, in (23) *The robot* is the object for *Bill gave Mary*, and it is the oblique object for *John gave a ball to*.

(23) The robot that Bill gave Mary and John gave a ball to

This variation reflects a more general uncertainty about what role the head of a relative clause can play in the relative clause, as illustrated in (24):

(24) The robot that Bill gave Mary
 The robot that gave Bill Mary
 The robot that John said Bill gave Mary
 The robot that Tom claimed John said Bill gave Mary, etc.

In fact, the number of roles that the head of a relative clause can play is theoretically unbounded.

To deal with these possibilities, the notion of *functional uncertainty* has been introduced into LFG theory (Kaplan and Zaenen, 1989; Kaplan and Maxwell, 1988). With functional uncertainty the attribute of a functional equation is allowed to consist of a (possibly infinite) regular set of attribute strings. For instance, normally the role that a constituent plays in the f-structure is given by a simple equation such as (25):

$$(25) \quad (f_1 \text{ OBJ}) = f_2$$

A functionally uncertain equation that could be used to express the relationship between the head of a relative clause and the role that it plays in the clause might look like (26):

$$(26) \quad (f_1 \text{ COMP}^* \text{ GF}) = f_2$$

Equation (26) says that the functional relationship between f_1 and f_2 could consist of any number of COMPS followed by a grammatical function, such as SUBJ or OBJ.

The definition of functional uncertainty given by Kaplan and Zaenen (1989) is essentially as follows:

$$(27) \quad \text{If } \alpha \text{ is a regular expression, then } (f\alpha) = v \text{ holds if and only if} \\ ((fa) \text{ Suff}(a, \alpha)) = v \text{ for some symbol } a, \text{ where} \\ \text{Suff}(a, \alpha) \text{ is the set of suffix strings } y \text{ such that } ay \in \alpha.$$

We will not discuss functional uncertainty further in this paper, except to show how it fits into our model for sets. To achieve the proper interaction between sets and regular expressions, we merge (27) with (16):

$$(28) \quad (s \alpha) = v = \prod (f_i \alpha), \text{ for all } f_i \in s \\ = \prod ((f_i a_i) \text{ Suff}(a_i, \alpha)), \text{ for all } f_i \in s$$

Allowing different a_i to be chosen for each f_i provides the variation needed for (23). The uncertainty can be realized by a different functional path in each of the coordinated elements, but the uncertainty must be resolved somehow in each element and this accounts for the familiar Across the Board and Coordinate Structure Constraints.

Representing the Conjunction

We have not yet indicated how the identity of the particular conjunction is represented. If we look at rule (14) again, we notice that it is missing any equation to tell us how the f-structure for CONJ is related to \uparrow :

$$(29) \quad \begin{array}{ccccc} V & \rightarrow & V & \text{CONJ} & V \\ & & \downarrow \in \uparrow & ? & \downarrow \in \uparrow \end{array}$$

If we replace the ? with $\uparrow = \downarrow$, then the f-structure for CONJ will be identified with the set corresponding to \uparrow , which will have the effect of distributing all of its information across the f-structures corresponding to the conjoined verbs. As was pointed out to us by researchers at the University of Manchester (UMIST), this arrangement leads to inconsistencies when coordinations of different types (*and* vs. *or*) are mutually embedded. On the other hand, if we replace the ? with $\downarrow \in \uparrow$, then the f-structure for CONJ will be another element of the set, on a par with the f-structures corresponding to the conjoined verbs. This is clearly counterintuitive and also erroneously implies that the shared elements will be distributed across the conjunction as well as the elements of the set.

We observe, however, that the identity of the particular conjunction does not seem to enter into any syntactic or functional generalizations, and therefore, that there is no motivation for including it in the functional structure at all. Instead, it is necessary to encode this information only on the semantic level of representation, as defined by a semantic structural correspondence or "projection" (Kaplan, 1987). A projection is a piecewise function mapping from the units of one kind of structure to the units of another. The projection that is most central to LFG theory is the ϕ projection, the one that maps from constituent structure nodes into functional structures. But other projections are being introduced into LFG theory so that generalizations about various other subsystems of linguistic information can be formalized. In particular, Halvorsen and Kaplan (1988) have discussed the σ projection that maps from f-structures into a range of semantic structures. Given the projection concept, the various linguistic levels can be related to one another through "codescription", that is, the equations that describe the mapping between f-structures and s-structures (semantic

structures) are generated in terms of the same c-structure node configurations as the equations that map between c-structures and f-structures. This means that even though the s-structure is mapped from the f-structure, it may contain information that is not computable from the f-structure but is strongly correlated with it via codescription. We exploit this possibility to encode the identity of the conjunction only in semantic structure.

Consider a modified version of (29) that has equations describing the semantic structures corresponding to the f-structure units:

$$(30) \quad V \quad \rightarrow \quad \begin{array}{c} V \\ \downarrow \in \uparrow \\ \sigma \downarrow = (\sigma \uparrow \text{ ARG1}) \end{array} \quad \begin{array}{c} \text{CONJ} \\ (\sigma \uparrow \text{ REL}) = \sigma \downarrow \end{array} \quad \begin{array}{c} V \\ \downarrow \in \uparrow \\ \sigma \downarrow = (\sigma \uparrow \text{ ARG2}) \end{array}$$

Rule (30) says that the unit of semantic structure corresponding to the f-structure of the conjoined verb contains the conjunction as its main relation (REL), plus two ARGs consisting of the semantic structures corresponding to the f-structures that correspond to the individual V's. The semantic structure generated by (30) is something like this:

$$(31) \quad \left[\begin{array}{l} \text{REL AND} \\ \text{ARG1} \left[\begin{array}{l} \text{REL BUY} \\ \dots \dots \end{array} \right] \\ \text{ARG2} \left[\begin{array}{l} \text{REL EAT} \\ \dots \dots \end{array} \right] \end{array} \right]$$

It describes the conjoined verb as a relation, AND, which is applied to arguments consisting of the relation BUY and the relation EAT. Each of these relations also has arguments, the semantic structures corresponding to the shared subject and object of the sentence. Notice how this structure differs from the one that we find at the functional level (e.g. (13)). Rule (30) does not assign any functional role to the conjunction, yet all the necessary syntactic and semantic information is available in the complex of corresponding structures assigned to the sentence.

References

- Bresnan, J., Kaplan, R. M., and Peterson, P. Forthcoming. Coordination and the flow of information through phrase structure.
- Dahl, V. and McCord, M. 1983. Treating coordination in logic grammars. *Computational linguistics* 9, 69-91.
- Dougherty, R. C. 1970. A grammar of coordinate conjoined structures, I. *Language* 46, 850-898.
- Dowty, D. 1985. Type raising, functional composition, and non-constituent coordination. Paper presented to the Tucson Conference on Categorical Grammar, May 31-June 2 1985.
- Gazdar, G., Klein, E., Pullum, G. and Sag, I. 1985. *Generalized phrase structure grammar*. Cambridge: Harvard University Press.
- Halvorsen, P.-K. and Kaplan, R. M. 1988. Projections and Semantic Description in Lexical-Functional Grammar. Xerox PARC.
- Kaplan, R. M. 1987. Three seductions of computational psycholinguistics. In P. Whitelock, M. Wood, H. Somers, R. Johnson, and P. Bennett (eds.), *Linguistic theory and computer applications*. London: Academic Press, 149-188.
- Kaplan, R. M. and Bresnan, J., 1982. Lexical-functional grammar: A formal system for grammatical representation. In J. Bresnan (ed.), *The mental representation of grammatical relations*. Cambridge: MIT Press, 173-281.
- Kaplan, R. M. and Maxwell, J. T. 1988. An algorithm for functional uncertainty. COLING 88.
- Kaplan, R. M. and Zaenen, A. 1989. Long-distance dependencies, constituent structure, and functional uncertainty. In M. Baltin and A. Kroch (eds.), *Alternative Conceptions of Phrase Structure*. Chicago: Chicago University Press, 17-42.
- Sager, N. 1981. *Natural language information processing*. Reading, Mass.: Addison-Wesley.
- Steedman, M. 1985. Dependency and coordination in the grammar of Dutch and English. *Language* 61, 523-568.
- Woods, W. 1980. An experimental parsing system for transition network grammars. In R. Rustin (ed.), *Natural language processing*. New York: Algorithmics Press, 111-154.