1

# The Formal Architecture of Lexical-Functional Grammar

RONALD M. KAPLAN

**Abstract.** This paper describes the basic architectural concepts that underlie the formal theory of Lexical-Functional Grammar. The LFG formalism, which has evolved from previous computational, linguistic, and psycholinguistic research, provides a simple set of devices for describing the common properties of all human languages and the particular properties of individual languages. It postulates two levels of syntactic representation for a sentence, a constituent structure and a functional structure. These are related by a piece-wise correspondence that permits the properties of the abstract functional structure to be defined in terms of configurations of constituent structure phrases. The basic architecture crucially separates the three notions of structure, structural description, and structural correspondence. This paper also outlines some recent extensions to the original LFG theory that enhance its ability to express certain kinds of linguistic generalizations while remaining compatible with the underlying architecture. These include formal variations in the elementary linguistic structures, in descriptive notation, and in the arrangement of correspondences.

## 1 Introduction

Since it was first introduced by Kaplan and Bresnan (1982), the formalism of Lexical-Functional Grammar has been applied in the description of a wide range of linguistic phenomena. The basic features of the formalism

Earlier versions of this paper appeared in *Proceedings of ROCLING II*, ed. C.-R. Huang and K.-J. Chen (Taipei, Republic of China, 1989), 1–18, and *Journal of Information Science and Engineering*, vol. 5, 1989, 305–322.

*Formal Issues in Lexical-Functional Grammar.*
edited by
Mary Dalrymple
Ronald M. Kaplan
John T. Maxwell III
Annie Zaenen.

are quite simple: the theory assigns two levels of syntactic representation to a sentence, the constituent structure and functional structure. The c-structure is a phrase-structure tree that serves as the basis for phonological interpretation while the f-structure is a hierarchical attribute-value matrix that represents underlying grammatical relations. The c-structure is assigned by the rules of a context-free phrase structure grammar. Functional annotations on those rules are instantiated to provide a formal description of the f-structure, and the smallest structure satisfying those constraints is the grammatically appropriate f-structure.

This formal conception evolved in the mid-1970's from earlier work in computational and theoretical linguistics. Woods' (1970) Augmented Transition Networks demonstrated that a direct mapping between superficial and underlying structures was sufficient to encode the discrepancy between the external form of utterances and their internal predicate-argument relations. ATN grammars followed transformational grammar in using the same kind of mathematical structure, phrase-structure trees, as both surface and deep grammatical representations. Kaplan (1975) noticed that the strong transformational motivation for this commonality of representation did not exist in the ATN framework. Inputs and outputs of transformations had to be of the same formal type if rules were to feed each other in a derivational sequence, but a nonderivational approach imposed no such requirement. Thus, while hierarchical and ordered tree structures are suitable for representing the sequences of surface words and phrases, they are not particularly convenient for expressing more abstract relations among grammatical functions and features. Although the fact that *John* is the subject in *John saw Mary* can be formally represented in a tree in which *John* is the NP directly under the S node, there is no explanatory advantage in using such an indirect way of encoding this simple intuition. Kaplan (1975) proposed hierarchical attribute-value matrices, now familiar as f-structures, as a more natural way of representing underlying grammatical relations.

The ATN register setting operations enabled explicit reference to labels like Subject and Object. They were originally used to manipulate the temporary information that accumulated during the course of analyzing a sentence and which was reorganized at the end to form a traditional transformational deep structure. Kaplan (1975) saw no need for that reorganization, since the accumulated registers already contained all the significant grammatical information. But this change in register status from merely being a repository of necessary bookkeeping information to being the major target of linguistic analysis had far-reaching consequences. The exact nature of the register setting and accessing operations became issues of major theoretical importance, and theoretical commitments were also required for the particular configurations of register contents that the grammar associated with individual sentences. The LFG formalism emerged from a careful study of questions of this sort. The accumulated register information was formalized as monadic functions defined on the set of grammatical relation and feature names (SUBJ, OBJ, CASE), and the ATN computational operations for manipulating these functions evolved into the equational specifications in LFG's functional descriptions.

This formal machinery has served as backdrop for and has been refined by substantive investigations into the common properties of all human languages and the particular properties of individual languages. Early investigations established, for example, the universal character of grammatical functions like subject and object, general principles of control and agreement, and basic mechanisms for expressing and integrating lexical and syntactic information (see Bresnan 1982a,c; Bresnan and Kaplan 1982; Kaplan and Bresnan 1982; and other papers in Bresnan 1982b). These studies and more recent results have offered strong support for the general organization of the theory, but they have also uncovered problems that are difficult to handle in the theory as originally formulated. Thus, a number of extensions and revisions to LFG are currently under consideration, dealing with long-distance dependencies, coordination, word-order, and semantic and pragmatic interpretation. Some of these proposals may seem at first sight like radical departures from the details of traditional LFG. But the LFG formalism as presented by Kaplan and Bresnan (1982) was an expression of a general underlying architectural conception, and most recent proposals remain quite compatible with that basic perspective.

That underlying architecture is the focus of the present paper. In the first section I review and explicate the fundamental notions that guided the development of the LFG formalism. These ideas provide a general view of the way in which different properties of an utterance can be represented and interrelated, and how constraints on those representations can be expressed. The second section surveys some of the recently proposed extensions to LFG, suggesting that they can be regarded as variations on the basic architectural theme.

## 2 Fundamental notions: Structures, descriptions, and correspondences

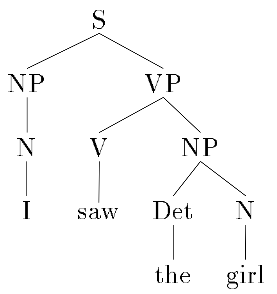
LFG posits two levels of syntactic representation for a sentence, and, as indicated above, these are of different formal types. This is a fundamental architectural presupposition of LFG and is the main point of departure for understanding the theory's formal organization. These different representations reflect the fact that there are different kinds of informational dependencies among the parts of a sentence, and that these are best expressed using different formal structures. The goal is to account for significant linguistic generalizations in a factored and modular way by means of related but appropriately dissimilar representations.

**Elementary structures.** We start with the simplest mathematical notion of a structure as a set of elements with some defined relations and properties. The strings that make up a sentence such as (1) are a trivial example: the elements of the set are the words and immediate precedence is the only native relation. The looser nonimmediate precedence relation is specified indirectly, as the transitive closure of immediate precedence.

(1) I saw the girl.

The phrase structure tree representing surface constituency configurations (2) is a slightly more complex example. The elements of this structure are nodes which are labeled by parts of speech and abstract phrasal categories and satisfy native relations of precedence (a partial order in this case) and immediate domination.

(2)




To put it in more explicit terms, a tree consists of a set of nodes N related by a labeling function $\lambda$ that takes nodes into some other finite labeling set L, a mother function M and that takes nodes into nodes, and a partial ordering $<$:

(3) N: set of nodes, L: set of category labels
$\mathrm{M}: \mathrm{N} \rightarrow \mathrm{N}$
$< \subseteq \mathrm{N} \times \mathrm{N}$
$\lambda: \mathrm{N} \rightarrow \mathrm{L}$

LFG admits only nontangled trees: for any nodes $n_1$ and $n_2$, if $\mathrm{M}(n_1) < \mathrm{M}(n_2)$, then $n_1 < n_2$.

Our third example is the functional structure illustrated in (4), which explicitly represents the primitive grammatical relations of subject, predicate, and object, as well as various kinds of agreement features.

(4)
$$\left[\begin{array}{ll}
\textsc{subj} & \left[\begin{array}{ll} \textsc{pred} & \text{`pro'} \\ \textsc{pers} & 1 \\ \textsc{num} & \textsc{sg} \end{array}\right] \\
\textsc{tense} & \textsc{past} \\
\textsc{pred} & \text{`see}\langle(\uparrow\ \textsc{subj}),\ (\uparrow\ \textsc{obj})\rangle\text{'} \\
\textsc{obj} & \left[\begin{array}{ll} \textsc{pred} & \text{`girl'} \\ \textsc{def} & + \\ \textsc{pers} & 3 \\ \textsc{num} & \textsc{sg} \end{array}\right]
\end{array}\right]$$

F-structures are defined recursively: they are hierarchical finite functions mapping from elements in a set of symbols to values which can be symbols, subsidiary f-structures, or semantic forms such as ‘see<SUBJ, OBJ>’. The set of f-structures F is characterized by the following recursive domain equation:

(5) A: set of atomic symbols, S: set of semantic forms
$$\mathrm{F} = (\mathrm{A} \rightarrow_f \mathrm{F} \cup \mathrm{A} \cup \mathrm{S})$$

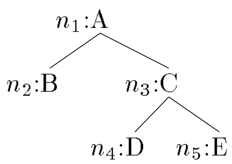
In effect, the only defining relation for f-structures is the argument-value relation of function application.

**Descriptions of structures.** Given a collection of well-defined structure-types whose defining relations can represent various kinds of linguistic dependencies, the problem of grammatical analysis is to ensure that all and only the appropriate structures are assigned to the sentences of the language. Structures can be assigned by *constructive* or *procedural* methods, by a set of operations that either analyze the properties of a string and build appropriate abstract representations that are consistent with these properties (as in the ATN approach) or that synthesize an abstract structure and systematically convert it to less abstract structures until the string is reached (the canonical interpretation of a transformational derivation). Alternatively, structures can be assigned by *descriptive*, *declarative*, or *model-based* methods. In this case, the properties of one structure (say, the string) are used to generate formal descriptions of other representations, in the form of a collection of constraints on the defining relations that those structures must possess. There are no operations for building more abstract or more concrete representations—any structures that satisfy all the propositions in the description are acceptable. These are the description's models.

The descriptive, model-based approach is, of course, the hallmark of LFG. This is motivated by the fact that particular properties of other representations are not neatly packaged within particular words or phrases. Rather, each word or phrase provides only some of the information that goes into defining an appropriate abstract representation. That information interacts with features of other words to uniquely identify what the abstract properties must be. The constraints on grammatical representations are distributed in partial and piecemeal form throughout a sentence—this is a second architectural presupposition of LFG theory. The descriptive method accommodates most naturally to this modular situation, since partial information can be assembled by a simple conjunction of constraints that can be verified by straightforward satisfiability tests.

We implement the descriptive approach in the most obvious way: a description of a structure can consist simply of a listing of its defining properties and relations. Taking a more formal example, we can write down a description of a tree such as (6) by introducing names ($n_1$, $n_2$ etc.) to stand for the various nodes and listing the propositions that those nodes satisfy. For this tree, the mother of $n_2$ is $n_1$, the label of $n_1$ is A, and so forth. A complete description of this tree is provided by the set of equations formulated in (7):

(6)




(7)
$$
\begin{array}{ll}
\mathrm{M}(n_2) = n_1 & \mathrm{M}(n_4) = n_3 \\
\lambda(n_1) = \mathrm{A} & \mathrm{M}(n_5) = n_3 \\
\lambda(n_2) = \mathrm{B} & \lambda(n_4) = \mathrm{D} \\
\mathrm{M}(n_3) = n_1 & \lambda(n_5) = \mathrm{E} \\
\lambda(n_3) = \mathrm{C} & n_4 < n_5 \\
n_2 < n_3 &
\end{array}
$$

This description is presented in terms of the tree-defining properties and relations given in (3).

We can also write down a set of propositions that a given f-structure satisfies. For the f-structure in (8), where the names $f_i$ are marked on the opening brackets, we note that $f_1$ applied to q is the value $f_2$, $f_2$ applied to s is t, and so forth.

(8)
$$
f_1{:}\begin{bmatrix} \mathrm{q} & f_2{:}\begin{bmatrix} \mathrm{s} & \mathrm{t} \\ \mathrm{u} & \mathrm{v} \end{bmatrix} \\ \mathrm{w} & \mathrm{x} \end{bmatrix}
$$

Using LFG's parenthetic notation for function application as defined in (9), the constraints in (10) give the properties of this f-structure.

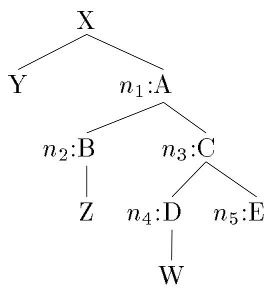
(9) (f a) = v iff <a v> $\in$ f, where f is an f-structure and a is an atomic symbol

(10) $(f_1\ \mathrm{q}) = f_2$
$(f_2\ \mathrm{s}) = \mathrm{t}$
$(f_2\ \mathrm{u}) = \mathrm{v}$
$(f_1\ \mathrm{w}) = \mathrm{x}$

Structures can thus be easily described by listing their properties and relations. Conversely, given a consistent description, the structures that satisfy it may be discovered—but not always. For the simple functional domain of f-structures, descriptions that involve only equality and function application can be solved by an attribute-value merging or unification operator, or other techniques that apply to the quantifier-free theory of equality (e.g. Kaplan and Bresnan 1982). But allowing more expressive predicates into the description language may lead to descriptions whose satisfiability cannot be determined. For example, I discuss below the proposal of Kaplan and Zaenen (1989b) to allow specifications of regular languages to appear in the attribute position of an LFG function-application expression. Their notion of "functional uncertainty" permits a better account of long-distance dependencies and other phenomena than the constituent-control theory of Kaplan and Bresnan (1982) provided. Kaplan and Maxwell (1988a) have shown that the satisfiability of uncertainty descriptions over the domain of acyclic f-structures is decidable, but the problem may be undecidable for certain types of cyclic f-structures (e.g. those that also satisfy constraints such as $(f\ \mathrm{x})=f$). This example indicates the need for caution when adding richer predicates to a descriptive formalism; so far, however, theoretically interesting description-language extensions have been well-behaved when applied in linguistically reasonable structural domains.

A set of propositions in a given structural description is usually satisfied by many structures. The description (7) is satisfied by the tree (6) but it is also satisfied by an infinite number of larger trees (e. g. (11)). It is true of this tree that the mother of $n_2$ is $n_1$ and, indeed, all the equations in (7) are true of it. But this tree has nodes beyond the ones described in (7) and it satisfies additional propositions that the tree in (6) does not satisfy.

(11)




In general, structures that satisfy descriptions form a semi-lattice that is partially ordered by the amount of information they contain. The minimal structure satisfying the description may be unique if the description itself is determinate, if there are enough conditions specified and not too many unknowns. The notion of minimality figures in a number of different ways within the LFG theory, to capture some intuitions of default, restriction, and completeness.

LFG clearly distinguishes the mathematical structures that comprise linguistic representations from the propositions in a description language that characterize those structures, that those structures serve as models for. This is an important difference between LFG and other so-called "unification-based" theories of grammar, such as Kay's (1979, 1984) Functional Unification Grammar. If the only descriptions are simple conjunctions of defining properties and relations, then there is an isomorphic mapping between the descriptions and the objects being described. Further, combining two descriptions by a unification operation yields a resulting description that characterizes all objects satisfying both those descriptions. Thus, in simple situations the distinction between descriptions and objects can safely be ignored, as Kay proposed. But the conflation of these two notions leads to conceptual confusions when natural extensions to the description language do not correspond to primitive properties of domain objects. For example, there is no single primitive object that naturally represents the negation or disjunction of some collection of properties, yet it is natural to form descriptions of objects by means of such arbitrary Boolean combinations of defining propositions. Kay's FUG represents disjunctive constraints as sets of descriptions: a set of descriptions is satisfied if any of its member descriptions is satisfied. This contrasts with the equally plausible interpretation that a set of descriptions is satisfied by a collection of more basic structures, one satisfying each of the elements of the description set. The Kasper and Rounds (1986) logic for feature structures clarified this issue by effectively resurrecting for FUG the basic distinction between objects and their descriptions.

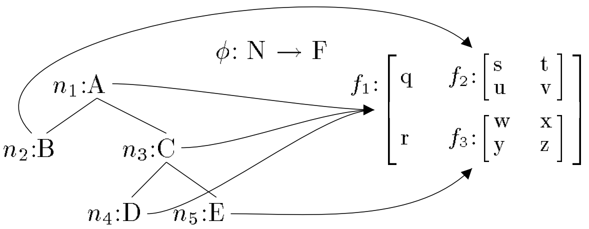
As another example of the importance of this distinction, no single object can represent the properties of long-distance dependencies that Kaplan and Zaenen (1989b) encode in specifications of functional uncertainty. As discussed below, they extend the description language to include constraints such as:

$$(f \text{ COMP} * \{\text{SUBJ} \mid \text{OBJ}\}) = (f \text{ TOPIC})$$

The regular expression in this equation denotes an infinite set of alternative strings, and such a set does not exist in the domain of basic structures. The Kaplan/Zaenen approach to long-distance dependencies is thus incompatible with a strict structure/description isomorphism.

**Structural correspondences.** We have seen that structures of different types can be characterized in different kinds of description languages. It remains to correlate those structures that are properly associated with a particular sentence. Clearly, the words of the sentence and their grouping and ordering relationships carry information about (or supply constraints on) the linguistic dependencies that more abstract structures represent. In the LFG approach, this is accomplished by postulating the existence of other very simple formal devices, correspondence functions that map between the elements of one (usually more concrete) structure and those of another; the existence of structural correspondences is the third architectural presupposition of LFG. The diagram in (12) illustrates such an element-wise correspondence, a function $\phi$ that goes from the nodes of a tree into units of f-structure space.

(12)




This function maps nodes $n_1$, $n_3$, and $n_4$ into the outer f-structure $f_1$, and nodes $n_2$ and $n_5$ to the subsidiary f-structures $f_2$ and $f_3$, respectively. A correspondence by itself only establishes a connection between the pieces of its domain and range structures, unlike a more conventional interpretation function that might also at the same time derive the desired formal properties of the range. But nothing more than these simple correspondence connections is needed to develop a description of those formal properties. Previously we described an f-structure by specifying only f-structure properties and elements, independent of any associated c-structure. The structural correspondence now permits descriptions of range f-structures to be formulated in terms of the elements and native relations of the tree. In other words, the element-wise structural correspondence allows the mother-daughter relationships in the tree to constrain the function-application properties in the f-structure, even though those formal properties are otherwise completely unrelated.

The f-structure in (12), for example, satisfies the condition that $(f_1\ \mathrm{q}){=}f_2$, a constraint in the f-structure description language. But $f_1$ and $f_2$ are the f-structures corresponding to $n_1$ and $n_2$, respectively, so this condition can be expressed by the equivalent $(\phi(n_1)\ \mathrm{q}) = \phi(n_2)$. Finally, noting that $n_1$ is the mother of $n_2$, we obtain the equation $(\phi(\mathrm{M}(n_2))\ \mathrm{q}){=}\phi(n_2)$, which establishes a dependency between a node configuration in part of the tree and value of the q attribute in the corresponding f-structure. Systematically replacing the $f_i$ identifiers in the usual description of the f-structure by the equivalent $\phi(n_i)$ expressions and making use of the mother-daughter tree relations leads to an alternative characterization of (12):

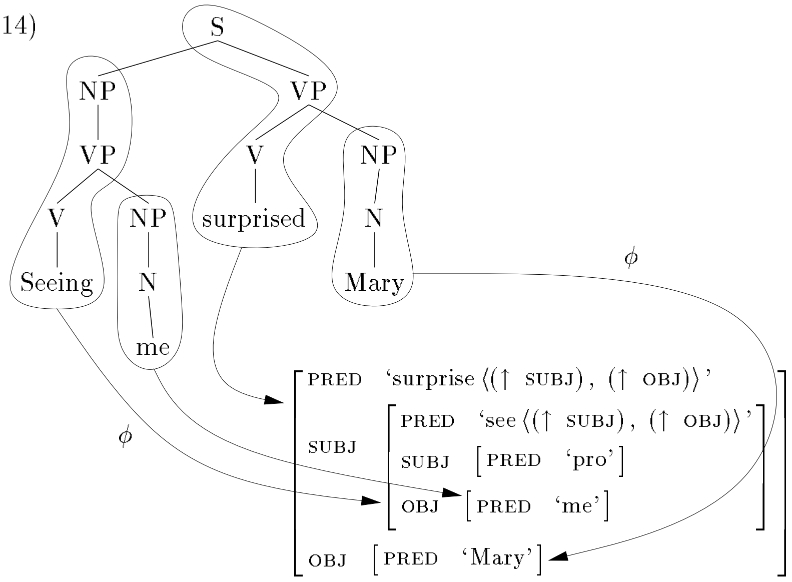
(13)
$$
\begin{array}{ll}
(\phi(\mathrm{M}(n_2))\ \mathrm{q}) = \phi(n_2) & \mathrm{M}(n_2) = n_1 \\
(\phi(n_2)\ \mathrm{s}) = \mathrm{t} & \\
(\phi(n_5)\ \mathrm{y}) = \mathrm{z} & \\
\phi(\mathrm{M}(n_3)) = \phi(n_3) & \mathrm{M}(n_3) = n_1 \\
\phi(\mathrm{M}(n_4)) = \phi(n_4) & \mathrm{M}(n_4) = n_3 \\
(\phi(\mathrm{M}(n_5))\ \mathrm{r}) = \phi(n_5) & \mathrm{M}(n_5) = n_3 \\
\text{etc.} &
\end{array}
$$

Thus, our notions of structural description and structural correspondence combine in this way so that the description of a range structure can involve not only its own native relations but also the properties of a corresponding domain structure.

We require a structural correspondence to be a function but it is not required to be one-to-one. As illustrated in (12), the correspondence $\phi$ maps the nodes $n_1$, $n_3$, and $n_4$ all onto the same f-structure $f_1$. When several nodes map onto the same f-structure, that f-structure can be loosely interpreted as the equivalence class or quotient of nodes induced by the correspondence. Conceptually, it represents the folding together or normalization of information carried jointly by the individual nodes that map onto it. Many-to-one configurations appear in many linguistic analyses. Lexical heads and their dominating phrasal categories, for example, usually map to the same f-structure, encoding the intuition that a phrase receives most of its functional properties from its head. Discontinuous constituents, functional units whose properties are carried by words in noncontiguous parts of the string, can be characterized in this way, as demonstrated by the Bresnan et al. (1982) analysis of Dutch cross-serial dependencies.

A structural correspondence also need not be onto. This is illustrated by (14), which shows the c-structure and f-structure that might be appropriate for a sentence containing a gerund with a missing subject.

(14)




Phrasally-based theories typically postulate an empty node on the tree side in order to represent the fact that there is a dummy understood subject, because subjects (and predicate-argument relations) are represented in those theories by particular node configurations. In LFG, given that the notion of subject is defined in the range of the correspondence, we need not postulate empty nodes in the tree. Instead, the f-structure's description, derived from the tree relations of the gerund c-structure, can have an equation that specifies directly that the subject's predicate is an anaphoric pronoun, with no node in the tree that it corresponds to. This account of so-called null anaphors has interesting linguistic and mathematical properties, discussed below and in Kaplan and Zaenen (1989a).

In sum, the LFG formalism presented by Kaplan and Bresnan (1982) is based on the architectural notions of structure, structural description, and structural correspondence. Within this framework, particular notational conventions were chosen to suppress unnecessary detail and make it more convenient to express certain common patterns of description. Thus, the allowable c-structures for a sentence were specified by the rewriting rules of a context-free grammar (augmented by a Kleene-closure operator for repetitive expansions) rather than by what seemed to be a less perspicuous listing of dominance, precedence, and labeling relations. The description of an appropriate f-structure was derived from functional annotations attached to the c-structure rules. For interpreting these functional annotations, Kaplan and Bresnan defined a special instantiation procedure that relied implicitly on the c-structure to f-structure correspondence $\phi$. To see that dependence more explicitly, consider the annotated rewriting rule in (15):

(15) $$\mathrm{S} \longrightarrow \underset{(\phi(\mathrm{M}(n))\ \textsc{subj}) = \phi(n)}{\mathrm{NP}} \quad \underset{\phi(\mathrm{M}(n)) = \phi(n)}{\mathrm{VP}}$$

The context-free expansion is matched against nodes in a candidate c-structure to verify that the local $[_{\mathrm{S}}$ NP VP$]$ configuration is acceptable. The symbol $n$ in a constraint annotated to a category stands for the node that matches that particular category in the candidate tree. The annotations use that symbol, the mother function M, and the structural correspondence $\phi$ to express general propositions about the f-structures that correspond to the nodes that satisfy this rule. Thus, (15) specifies that the f-structure corresponding to the NP's mother applies to SUBJ to give the f-structure corresponding to the NP, and that the f-structure corresponding to the mother of the VP, namely the S node, is also the f-structure corresponding to the VP. The conjunction of these constraints across the whole c-structure, with actual nodes substituted for the generic $n$, is the desired f-structure description. Kaplan and Bresnan simplified to a more convenient notation. The symbol $\uparrow$ abbreviates the complex term $\phi(\mathrm{M}(n))$, the composition of the structural correspondence with the mother function, and $\downarrow$ stands for $\phi(n)$, the f-structure corresponding to the matching node. This reduces the annotation on the NP to the familiar form in (16):

(16) $(\uparrow \textsc{subj}) = \downarrow$

This can be read as 'the matching NP node's mother's f-structure's subject is the matching node's f-structure'. This method of generating range descriptions by analyzing and matching the properties of domain structures is what we call *description by analysis*. Halvorsen (1983) applied this technique to derive descriptions of semantic structures from an analysis of the f-structures they were assumed to correspond to.

LFG's store of basic underlying concepts is thus quite limited, yet it supports a notational system in which a variety of complex linguistic phenomena have been easy to characterize. Perhaps because of its simple architectural base, this system has remained remarkably stable in the years since it was introduced, particularly when compared to other formal frameworks that have undergone extensive revision over the same period of time. In continuing to explore the implications of this architecture, we have found some useful consequences that had previously gone unnoticed and have also seen the value of certain extensions and revisions. The remainder of this paper gives a brief survey of these more recent developments.

## 3 Extensions and variations

The tripartite division of structures, descriptions, and correspondences suggest three ways in which the theory might be modified. One way, of course, is to add to the catalog of structure-types that are used for linguistic representations. LFG currently acknowledges two syntactic structure-types beyond the string, and there may be grammatical phenomena that are best represented in terms of other native relations. Kaplan and Bresnan (1982) introduced one extension to the f-structure domain beyond the simple attribute-value properties that have been discussed here. They allowed the values of f-structure attributes to be sets of f-structures as well as individual f-structures, symbols, and semantic forms. Sets were used to represent grammatical relations such as adjuncts that can be independently realized in several positions in a clause and thus seemed to be immune to the functional uniqueness condition. The description language also was augmented with the membership operator $\in$, so that constraints on set elements could be stated.

A more recent example of how the properties of formal structures might usefully be extended can be seen in Bresnan and Kanerva's (1989) proposals for a natural-class organization of grammatical functions. They observe that many lexical redundancy rules can be eliminated in favor of general instantiation principles if lexical entries are marked with underspecified grammatical function labels (for example, a neutral objective function that subsumes (and can be instantiated as either) OBJ or OBJ2). In previous work, function labels were unanalyzable atomic symbols bearing no relation to one another. On this new suggestion, the functions are partially ordered in a subsumption lattice, and new principles of interpretation are required.

Beyond these relatively minor adjustments to the structural domain, there have been no proposals for substantially different ways of organizing linguistic information. By far the most interesting innovations have concerned the c-structure and f-structure description languages and the variety of attribute-value structures that can be related by structural correspondences.

**Extending the description language.** C-structures were described originally by context-free rewriting rules whose right-hand sides could contain the Kleene-closure operator and thus could denote arbitrary regular languages. The regular sets are closed not only under union and (Kleene) concatenation but also under intersection and complementation. Thus, the generative capacity of the c-structure component is unchanged if intersection and complementation are allowed as operators in c-structure rules. These operators permit many new ways of factoring c-structure generalizations, including but not limited to the ID/LP format that Pullum (1982) proposed for GPSG. Immediate dominance and linear precedence constraints can both be transformed into regular predicates using concatenation and complementation, and the combined effect of these constraints in a given rule can be obtained simply by intersecting that regular-set collection. For example, the unordered ID rule

(17) $\mathrm{S} \rightarrow [\mathrm{NP}, \mathrm{VP}]$

can be translated to the equivalent but less revealing form

(18) $\mathrm{S} \rightarrow [\mathrm{VP}^* \ \mathrm{NP} \ \mathrm{VP}^*] \cap [\mathrm{NP}^* \ \mathrm{VP} \ \mathrm{NP}^*]$

This intersection will admit an S node if its string of daughter nodes satisfies two conditions: it must contain one NP with some unknown number of VP's around it, and it must also contain one VP surrounded by some unknown number of NP's. The only strings that simultaneously satisfy both conditions are those that contain exactly one NP and one VP appearing in either order, and this is precisely the requirement intended by the ID rule (17). As detailed by Kaplan and Zaenen (1989a), this translation goes through even with repetition factors attached to the categories and does not require a complex multi-set construction for its mathematical interpretation as Gazdar et al. (1985) proposed. Similarly, linear-precedence restrictions can also be translated to simple, intersectable regular predicates. The condition that NP's must come before VP's, for example, is satisfied by strings in the regular set

$$\overline{\Sigma * \ \mathrm{VP} \ \Sigma * \ \mathrm{NP} \ \Sigma *}$$

where $\Sigma$ denotes the set of all categories and the over-bar indicates complementation with respect to $\Sigma^*$.

Thus, compact notation for immediate domination and linear precedence, as well as for other regular predicates described by Kaplan and Maxwell (1993), can be freely introduced without changing the power of the context free system. Some caution is required, however, for regular predicates defined over categories annotated with functional schemata.

Although the system of combined c-structure/f-structure constraints is closed under intersection (since the f-structure description language is closed under conjunction), it is not known whether it is closed under complementation of arbitrary regular expressions. The complement of a single annotated category can be translated to standard notation, however, by applying de Morgan's laws and using negated f-structure constraints. This more limited form of complementation is sufficient for the ID/LP specifications and for a number of other useful predicates.

Extensions to the c-structure description language provide one way of characterizing the kinds of ordering variations that appear across languages. The LFG architecture naturally provides for another way of expressing ordering dependencies, by defining an order-like relation (called f-precedence) on f-structures and including a precedence operator in the f-structure description language. The formal and empirical properties of f-precedence relation are explored at some length by Kaplan and Zaenen (1989a); here we give only a brief summary of their discussion. We first note that precedence is not a native relation on f-structure: f-structures are not distinguished by the order in which attributes and values appear. However, the native precedence relation in the c-structure (c-precedence to distinguish it from f-precedence) naturally induces a relation on f-structure by virtue of the c-structure to f-structure correspondence $\phi$. For two f-structures, $f_1$ and $f_2$, we say that $f_1$ f-precedes $f_2$ if and only if all nodes that $\phi$ maps into $f_1$ c-precede all nodes that $\phi$ maps into $f_2$. This can be formalized in terms of the inverse mapping $\phi^{-1}$:

(19) $f_1 <_f f_2$ iff
for all $n_1 \in \phi^{-1}(f_1)$ and for all $n_2 \in \phi^{-1}(f_2)$,
$n_1 <_c n_2$

This relation has some peculiar and unexpected properties because of the fact that $\phi$ may be neither one-to-one nor onto. A null anaphor is not the image of any node, and therefore it vacuously both f-precedes and is f-preceded by every other element in the f-structure. Mathematically, this implies that f-precedence is neither transitive nor anti-symmetric—it is not really an ordering relation at all. But these characteristics appear to be just what is needed to given a systematic account of certain constraints on anaphoric relations (Bresnan 1984; Kameyama 1988; Kaplan and Zaenen 1989a). Kaplan and Zaenen also point out one other interesting property of f-precedence: it can be used to impose ordering restrictions on nodes that are not sisters in the c-structure tree and may in fact be quite removed from each other. This can happen when the correspondence $\phi$ maps these nodes to locally related units of f-structure.

Functional precedence illustrates the interplay of description and correspondence mechanisms in expressing interesting linguistic constraints. Native relations in a domain structure map into induced relations on the range; these relations are typically degraded in some way, for the same reason that the range structures are degraded images of the domain structures they correspond to. The structural correspondence collapses some distinctions and in some cases introduces new ones, as it picks out and represents a subset of the domain's information dependencies. The definition of functional precedence given in (19) is an example of what we call *description through inversion*.

Functional uncertainty is another example of new expressive power obtained by extending the description language without changing the collection of underlying formal objects. The original LFG theory provided a mechanism of *constituent control* to characterize the constraints on long-distance dependencies (Kaplan and Bresnan 1982). Constituent control was essentially a translation into LFG terms of traditional phrasal approaches to long-distance dependencies, and carried forward the claim that the various constraints on those constructions were best formulated in terms of phrase and category configurations. Kaplan and Bresnan (1982) had briefly considered a functional approach to these phenomena, but rejected it since it seemed to require grammatical specifications of infinite size. Kaplan and Zaenen (1989b) proposed functional uncertainty as a new descriptive technique for avoiding the problem of infinite specification, reexamined the constituent control account of island constraints in light of this new technique, and concluded that functional restrictions offered a clearer and more accurate characterization of long-distance dependencies and island constraints. Kaplan and Zaenen simply extended the LFG notation for expressing function application so that the attribute position could be realized as a regular set. Thus, in addition to ordinary equations such as $(\uparrow \textsc{subj})=\downarrow$, it is possible to write in the grammar equations such as $(\uparrow \textsc{comp}^{*}\ \textsc{subj}|\textsc{obj})=\downarrow$. This equation expresses the uncertainty about what the within-clause functional role of an extraposed topic might be: it might be identified as either the subject or object of a clause embedded inside any number of complements. According to Kaplan and Zaenen, this constraint is satisfied by an f-structure if there is some string in the regular language $\textsc{comp}^{*}\ \textsc{subj}|\textsc{obj}$ such that the equation resulting from substituting that string for the regular expression is satisfied by that f-structure. In effect, the uncertainty expression provides a finite specification for what would otherwise be an infinite disjunction. Under this proposal, the constraints on when a long-distance dependency is permitted are embodied in restrictions on the regular expressions that appear in uncertainty equations, and are quite independent of categorial config-

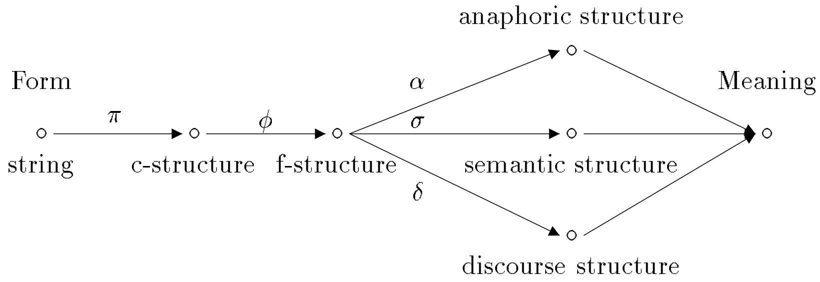



FIGURE 1 Decomposition of $\Gamma$

urations. Kaplan and Zaenen give a number of arguments in support of this functional approach, pointing out, for example, that subcategorized functions but not adjuncts can be extracted in Icelandic, even though these appear in identical phrase-structure positions.

**Extending the configuration of correspondences.** The LFG architecture was developed with only two syntactic structures set in correspondence, but the correspondence idea provides a general way of correlating many different kinds of linguistic information through modular specifications. Representations of anaphoric dependencies, discourse functions, and semantic predicate-argument and quantifier relations can all be connected in mutually constraining ways by establishing an appropriate set of structures and correspondences. One hypothetical configuration for mapping between the external form of an utterance and internal representations of its meaning (e.g., the claims that it makes about the world, speaker, discourse, etc.) is shown in Figure 1. Starting out with the word string, we assume a structural correspondence $\pi$ that maps to the phrases of the constituent structure, which is then mapped by $\phi$ to the functional structure in the usual LFG way. We might postulate a further correspondence $\sigma$ from f-structure to units of a semantic structure that explicitly marks predicate-argument relationships, quantifier scope ambiguities, and so forth—dependencies and properties that do not enter into syntactic generalizations but are important in characterizing the utterance's meaning. We might also include another correspondence $\alpha$ defined on f-structures that maps them onto anaphoric structures: two f-structure units map onto the same element of anaphoric structure just in case they are coreferential. The figure also shows a mapping $\delta$ from f-structure to a level of discourse structure to give a separate formal account of discourse notions such as topic and focus. The anaphoric and discourse structures, like the semantic structure, also contribute to meaning representations. By fitting these other systems of linguistic information into the same conceptual framework of description and correspondence, we can make use of already existing mathematical and computational techniques.

We note, however, that this arrangement suggests a new technique for generating abstract structure descriptions. In this diagram, the f-structure is both the range of $\phi$ and the domain of $\sigma$ (and also $\alpha$ and $\delta$). Thus the composition of $\sigma$ and $\phi$ is implicitly a function that maps from the c-structure directly to the semantic structure, and this can also be regarded as a structural correspondence. This enables somewhat surprising descriptive possibilities. Since $\sigma$ only maps between f-structure and semantic structure, it might seem that the semantic structure may only contain information that is derivable from attributes and values present in the f-structure. This would be expected if the correspondence $\sigma$ were an interpretation function operating on the f-structure to produce the semantic structure. The semantic structure, for example, could not reflect category and precedence properties in the c-structure that do not have correlated features in the f-structure. But $\sigma$, as an element-wise correspondence, does not interpret the f-structure at all. It is merely a device for encoding descriptions of the semantic structure in terms of f-structure relations. And since the f-structure is described in terms of $\phi$ and c-structure properties, the composition $\sigma(\phi(n))$ can be used to assert properties of semantic structure also in terms of c-structure relations, even though there is no direct correspondence. Descriptions generated by the context-free grammar can use designators such as $\sigma\uparrow$ $[=\sigma(\phi(\mathrm{M}(n)))]$ along with $\uparrow$ to characterize f-structure and semantic structure simultaneously.

In general, a compositional arrangement of correspondences permits the *codescription* of separate levels of representation, yet another descriptive technique that has been applied to a number of problems. Halvorsen and Kaplan (1988) explore various uses of codescription in defining the syntax/semantics interface. Kaplan and Maxwell (1988b) exploit a codescription configuration in their account of constituent coordination in LFG. To deal with coordinate reduction, they interpreted function application on f-structure set-values as picking out a value from the mathematical generalization of the set elements. This properly distributes grammatical functions and predicates over the reduced clauses, but there is no place in the resulting f-structure to preserve the identity of the conjunction (*and* or *or*) which is required in the semantic structure to properly characterize the meaning. A codescriptive equation establishes the proper conjunction in the semantic structure even though there is no trace of it in the f-structure. As a final application, Kaplan et al. (1989) suggest using codescription as a means for relating source and target functional and semantic structures in a machine translation system.

## 4 Conclusion

The formal architecture of Lexical-Functional Grammar provides the theory with a simple conceptual foundation. These underlying principles have become better understood as the theory has been applied to a wide range of grammatical phenomena, but the principles themselves have remained essentially unchanged since their inception. The recent work surveyed in this paper has identified and explored a number of variations that this architecture allows, in an effort to find more natural and formally coherent ways of discovering and expressing linguistic generalizations. Promising new descriptive devices are being introduced and new correspondence configurations are being investigated. The success of these mechanisms in easily extending to new areas of grammatical representation indicates, perhaps, that this architecture mirrors and formalizes some fundamental aspects of human communication systems.

## References

Bresnan, Joan. 1982a. Control and Complementation. In *The Mental Representation of Grammatical Relations*, ed. Joan Bresnan. 282–390. Cambridge, MA: The MIT Press.

Bresnan, Joan (ed.). 1982b. *The Mental Representation of Grammatical Relations*. Cambridge, MA: The MIT Press.

Bresnan, Joan. 1982c. The Passive in Lexical Theory. In *The Mental Representation of Grammatical Relations*, ed. Joan Bresnan. 3–86. Cambridge, MA: The MIT Press.

Bresnan, Joan, and Ronald M. Kaplan. 1982. Introduction: Grammars as Mental Representations of Language. In *The Mental Representation of Grammatical Relations*. xvii–lii. Cambridge, MA: The MIT Press.

Bresnan, Joan, Ronald M. Kaplan, Stanley Peters, and Annie Zaenen. 1982. Cross-serial Dependencies in Dutch. *Linguistic Inquiry* 13:613–635.

Bresnan, Joan. 1984. Bound Anaphora on Functional Structures. Presented at the Tenth Annual Meeting of the Berkeley Linguistics Society.

Bresnan, Joan, and Jonni M. Kanerva. 1989. Locative Inversion in Chicheŵa: A Case Study of Factorization in Grammar. *Linguistic Inquiry* 20(1):1–50. Also in E. Wehrli and T. Stowell, eds., Syntax and Semantics 26: Syntax and the Lexicon. New York: Academic Press.

Gazdar, Gerald, Ewan Klein, Geoffrey K. Pullum, and Ivan A. Sag. 1985. *Generalized Phrase Structure Grammar*. Cambridge, MA: Harvard University Press.

Halvorsen, Per-Kristian. 1983. Semantics for Lexical-Functional Grammar. *Linguistic Inquiry* 14(4):567–615.

Halvorsen, Per-Kristian, and Ronald M. Kaplan. 1988. Projections and Semantic Description in Lexical-Functional Grammar. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, 1116–1122.

Tokyo, Japan. Institute for New Generation Systems. Reprinted in Part IV of this volume.

Kameyama, Megumi. 1988. Functional Precedence Conditions on Overt and Zero Pronominals. Unpublished MS, MCC, Austin, Texas.

Kaplan, Ronald M. 1975. On Process Models for Sentence Comprehension. In *Explorations in cognition*, ed. Donald A. Norman and David E. Rumelhart. San Francisco: W. H. Freeman.

Kaplan, Ronald M., and Joan Bresnan. 1982. Lexical-Functional Grammar: A Formal System for Grammatical Representation. In *The Mental Representation of Grammatical Relations*, ed. Joan Bresnan. 173–281. Cambridge, MA: The MIT Press. Reprinted in Part I of this volume.

Kaplan, Ronald M., and John T. Maxwell. 1988a. An Algorithm for Functional Uncertainty. In *Proceedings of COLING-88*, 297–302. Budapest. Reprinted in Part II of this volume.

Kaplan, Ronald M., and John T. Maxwell. 1988b. Constituent Coordination in Lexical-Functional Grammar. In *Proceedings of COLING-88*, 303–305. Budapest. Reprinted in Part II of this volume.

Kaplan, Ronald M., Klaus Netter, Jürgen Wedekind, and Annie Zaenen. 1989. Translation by Structural Correspondences. In *Proceedings of the Fourth Meeting of the European ACL*, 272–281. University of Manchester, April. European Chapter of the Association for Computational Linguistics. Reprinted in Part IV of this volume.

Kaplan, Ronald M., and Annie Zaenen. 1989a. Functional Precedence and Constituent Structure. In *Proceedings of ROCLING II*, ed. Chu-Ren Huang and Keh-Jiann Chen, 19–40. Taipei, Republic of China.

Kaplan, Ronald M., and Annie Zaenen. 1989b. Long-distance Dependencies, Constituent Structure, and Functional Uncertainty. In *Alternative Conceptions of Phrase Structure*, ed. Mark Baltin and Anthony Kroch. Chicago University Press. Reprinted in Part II of this volume.

Kaplan, Ronald M., and John T. Maxwell. 1993. LFG Grammar Writer's Workbench. Unpublished technical report. Xerox Palo Alto Research Center.

Kasper, Robert T., and William C. Rounds. 1986. A Logical Semantics for Feature Structures. In *Proceedings of the Twenty-Fourth Annual Meeting of the ACL*. New York. Association for Computational Linguistics.

Kay, Martin. 1979. Functional Grammar. In *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistic Society*, ed. Christine Chiarello and others, 142–158. The University of California at Berkeley. Berkeley Linguistics Society.

Kay, Martin. 1984. Functional Unification Grammar: A Formalism for Machine Translation. In *Proceedings of COLING-84*, 75–78. Stanford, CA.

Pullum, Geoffrey K. 1982. Free Word Order and Phrase Structure Rules. In *Proceedings of the Twelfth Annual Meeting of the North Eastern Linguistic Society*, ed. James Pustejovsky and Peter Sells, 209–220. University of Massachusetts at Amherst.

Woods, William A. 1970. Transition Network Grammars for Natural Language Analysis. *Communications of the ACM* 13(10):591–606.