Chapter 1

Introduction to LFG

Oleg Belyaev

Lomonosov Moscow State University and Institute of Linguistics of the Russian Academy of Sciences

This chapter provides a general summary of the architecture of LFG. It is mainly focused on describing the two main syntactic levels, c- and f-structure, and the projection architecture used in LFG in general. It also describes the notation for defining the range of possible c-structures and their corresponding f-structures. Core syntactic mechanisms such as structure sharing and X-bar theory are also briefly covered.

1 Introduction

In this chapter, I aim to summarize the main syntactic levels of LFG, constituent structure (c-structure) and functional structure (f-structure), while providing a general overview of the foundational features of this framework. In Section 2, I briefly describe the basic architecture of LFG and the overall role played by each of the syntactic levels. In Section 3, I describe the c-structure model used in standard LFG, its understanding of constituency, and the role of X′ theory. In Section 4, the notion of f-structure is introduced, together with notational conventions and a system of mapping c-structure to f-structure. In Section 5, I show how the basic system of c- and f-structure can be extended to include other levels of projection that comprise the architecture of LFG.

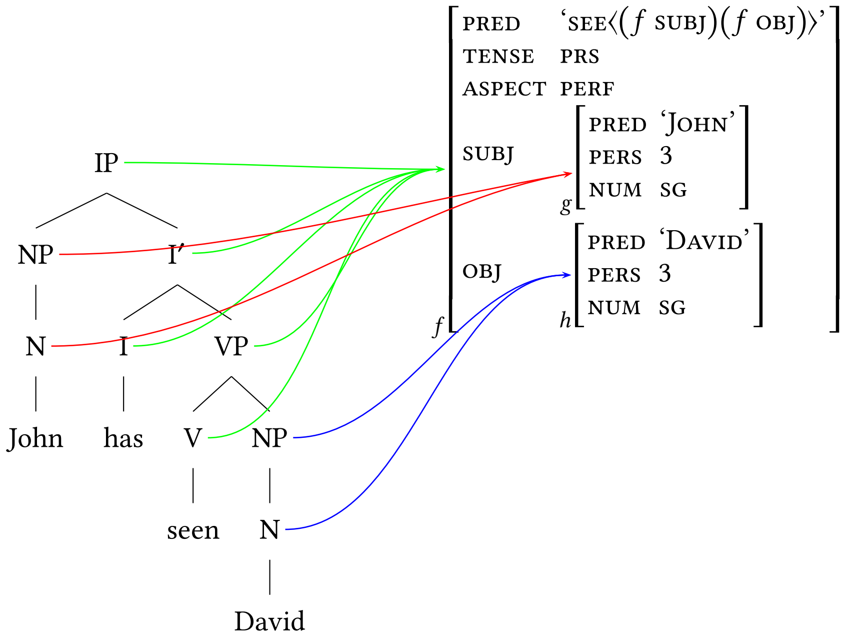
2 The basic architecture of LFG

At the core of LFG architecture as it was originally proposed in Kaplan & Bresnan (1982) is the split of syntax into two levels: constituent structure, or **c-structure**,

Oleg Belyaev. Introduction to LFG. To appear in: Dalrymple, Mary (ed.). Handbook of Lexical Functional Grammar. Berlin: Language Science Press. [preliminary page numbering]

and functional structure, or **f-structure**. The correspondence function $\phi(x)$ maps every c-structure node to an f-structure. As an example, consider the LFG analysis of the sentence *John has seen David* in (1), where the mapping function is represented by the arrows.

(1)



As seen in (1), the two parallel structures are substantially different: c-structure is a phrase structure tree that represents word order and hierarchical embedding, while f-structure is a feature-value structure that represents predicate-argument relations and the grammatical features of all the major parts of the sentence. Features appear as atomic values of f-structure attributes, while arguments and adjuncts appear as f-structures embedded as values of attributes such as SUBJ and OBJ in (1); which arguments can and, indeed, have to appear in the f-structure is specified in the value of the PRED attribute. While the mapping between the two structures follows certain constraints imposed both by the formal metalanguage and theoretical considerations (on which see Belyaev forthcoming(a) [this volume] and Andrews forthcoming [this volume], it is, in principle, language-specific: an LFG grammar consists of a set of rules and lexical entries that define the possible c-structures and their corresponding f-structures for a particular language.

This flexibility in the c- to f-structure correspondence ensures that each corresponds to a particular set of grammatical generalizations. Overall, f-structure is the main syntactic level that represents the predicates, their valencies and grammatical relations, as well as grammatical features such as number, case, aspect and gender. The majority of syntactic phenomena that have to do with feature assignment and feature checking are described using f-structure constraints; these include:

- feature government (case assignment, mood, constraints on the use of nonfinite forms, etc.);
- agreement;
- anaphoric constraints;
- wh-movement, topicalization and other long-distance dependencies.

All generalizations that have to do with argument relations and grammatical features have to be stated in terms of f-structure. For instance, a constraint that requires the verb to agree with Spec,IP or to assign accusative case to Comp,VP would be complex and somewhat unnatural to formulate (although not impossible). It is much more simple and natural in LFG for such rules to refer to grammatical functions such as SUBJ and OBJ instead. This implies that the role of constituent structure is more restricted than in other frameworks; for the most part, c-structure constraints only capture generalizations related to word order and various embedding possibilities.

The correspondence architecture is not limited to syntax. Other projections that map c-structure nodes or f-structures to other structures (such as information structure, semantic structure, or prosody) have been proposed in the literature: see Section 5 for details.

3 C-structure

3.1 The notion of c-structure

C-structure (constituent structure) in LFG is a phrase structure tree. Possible trees are defined by a set of context-free statements ("phrase structure rules") of the type $A \rightarrow \alpha$, where A is a nonterminal symbol (representing some syntactic category), while α is a string of nonterminals or a single terminal. A simple set of rules that licenses the English sentence in (1) is given in (2).

(2) a. IP → NP I′ b. I′ → I VP c. VP → V NP d. NP → N

Such rules are well-established in modern linguistics since at least Chomsky (1957) and so hardly require further discussion. It should however be observed that, in LFG, these should not be understood as "rules" in the direct (procedural) sense, but rather a set of phrase structure principles that constrain hierarchical relations between mothers and daughters – crucially, not between levels further apart, such as granddaughters etc. Phrase structure grammars are one way of describing such principles that has proved most popular among LFG practitioners, but not the only way – possible alternatives are ID/LP rules (Falk 1983) and the specification language described in Potts (2002), which builds on the specification language in Blackburn & Gardent (1995).

The structures that are constrained in this way are not just strings,[1] but constituent structure trees whose nodes are individually mapped to f-structures, as shown in (1).

The syntax of phrase structure rules in LFG is somewhat more extensive than in many other frameworks, because the right-hand side α is allowed to be a regular expression and include such features as optionality (represented by parentheses around the symbol), disjunction (with the disjuncts in curly brackets, separated by either a vertical line (|) or a logical disjunction sign (∨): e.g. { NP | DP }), Kleene star (zero or more instances, NP*), Kleene plus (one or more instances, NP^+), and some other less frequently used expressions. Grammars where the right-hand side can include regular expressions are called extended context-free grammars or regular right part grammars and it is known (Woods 1970) that the set of languages they describe is the same as that of standard context-free grammar.

3.2 Main properties of c-structure

LFG is unique among all frameworks in the simplicity of its constituent structure representations. This is a deliberate design decision which is possible due to the parallel architecture approach of LFG. It has been widely accepted since Chomsky (1957) that context-free grammar is not by itself an adequate formalism for describing natural language; even if the majority of syntactic constructions can indeed be described by context-free grammar (Pullum & Gazdar 1982), the descriptions required would be cumbersome, artificial and theoretically unenlightening as a model of human linguistic competence. Therefore, most grammars

[1] In fact, in the original version of LFG architecture introduced in Kaplan (1989), c-structure is itself a projection from the string. In recent LFG work, this idea has been developed in more detail by distinguishing between the *s-string* (the string of syntactic units) and the *p-string* (the string of phonological units), see Dalrymple & Mycock (2011) and Bögel forthcoming [this volume] for more information.

which use constituent structure as the main level of syntactic representation introduce additional mechanisms such as transformations in order to increase their expressive power. But such additions are not required in LFG because all phenomena that require more powerful mechanisms are dealt with at f-structure and other levels. C-structure remains limited to modeling basic word order facts, hierarchical embedding, and recursion, the phenomena for which phrase structure always was and remains the most adequate formal representation.

The advantage of this simplicity is that constituent structure in LFG has a clear empirical basis and can be determined for individual languages based on classic tests not obscured by additional considerations. For example, since there is no syntactic displacement, constituents in LFG are continuous by definition – apparently "discontinuous" material may eventually converge in one f-structure, but will still be split into separate constituents at c-structure.

By contrast, some constituency diagnostics which are valid in other frameworks are not valid in LFG. For example, since c-command is a phrase structure-based relation in mainstream transformational grammar, the existence of binding asymmetries between subjects and objects implies a configurational structure where the subject c-commands the object or vice versa. Thus Speas (1990: 137) argues that, within standard GB assumptions, flat structure predicts the existence of subject reflexives bound by their objects; since few such languages, if any, are actually found, existence of a hierarchical structure with a VP and a subject c-commanding the direct object is part of Universal Grammar.

In LFG, such a conclusion is a *non sequitur* because constraints on anaphoric relations, and other related phenomena, are formulated chiefly in terms of f-structure; sometimes in terms of information structure, semantics, or even linear precedence; but almost never in terms of c-structure configuration. Reference to c-command is possible in principle,[2] but it is largely useless as a source of valid generalizations due to the core assumptions of LFG: the cross-linguistic variability of c-structure, the universality[3] of grammatical functions at f-structure, and variation in the syntax-semantics interface.

Constituent structure representations in LFG are therefore rather "shallow" in that their makeup is determined by a limited set of empirical diagnostics mostly

[2] As, for example, in the definition of extended heads in Bresnan et al. (2016: 136). Note that this is a concept that is used to describe regularities in the c- to f-structure mapping, not a constraint on f-structure relations themselves.

[3] "Universality" here refers to universal availability, as in a grammatical toolbox (cf. Jackendoff 2002), not in the sense of mapping the same semantic roles to the same grammatical functions in all languages, or even in a single language. See Belyaev forthcoming(b) [this volume] for more detail.

based on word order possibilities. These facts vary widely across languages, and so do c-structure rules and the resulting structures. While f-structures have a degree of universality (in the sense of sharing a single inventory of grammatical functions and broad similarity in the way analogous phenomena such as anaphora, coordination, agreement etc. are represented), c-structures are language-specific.

Still, even in c-structure there are certain basic theoretical constraints which are deemed to hold universally across languages. In mainstream LFG, these are ENDOCENTRICITY and LEXICAL INTEGRITY. The former is usually captured by a version of X-Bar Theory, which is generally the same as in GB (see Chomsky 1970; Jackendoff 1977) but less restrictive: no universal clause or NP structure, no universal mapping from X′-theoretic positions (specifier, complement) to grammatical functions are assumed; non-binary branching is allowed; various exceptions from endocentricity, most prominently the exocentric S node used in non-configurational languages are permitted. For more information on the version of X-Bar Theory used in LFG, see Belyaev forthcoming(a) [this volume] and Andrews forthcoming [this volume].

Lexical integrity is another principle that has been assumed in LFG since its inception. At its core, this principle states that words are constructed from different elements and according to different rules than syntactic phrases, and that the internal structure of words is invisible to rules of syntax (Bresnan & Mchombo 1995: 181). In formal terms, this is usually interpreted such that the leaves of c-structure trees must be morphologically complete words (Bresnan et al. 2016: 92). For more detail on lexical integrity as it is used in LFG, the challenges it faces and proposed modifications, see Belyaev forthcoming(a) [this volume].

4 F-structure

4.1 Defining equations

As mentioned above, at the most basic level f-structures in LFG are a type of attribute-value structure.[4] However, unlike most other frameworks which deal with this data type, the LFG formalism does not refer to f-structures as objects

[4] Carpenter (1992) is the standard reference on the mathematical properties of such feature structures. However, the structures described by Carpenter are *typed*, which is a crucial difference from LFG f-structures, which are untyped and defined using a functional notation.

that can be manipulated and to which various operations can be applied. In contrast, an f-structure is thought of as a *function* that maps attributes (attribute names) to their values.[5]

From this perspective, describing an f-structure consists in defining the value y for each argument x in the function's domain (i.e. the set of attribute names). In LFG, attribute-value pairs are usually described using the notation of function application probably inspired by the Lisp programming language, i.e. the more conventional $f(x) = y$ is expressed as $(f\ x) = y$. Thus, for the f-structure f in (1), the value of the attribute TENSE is defined by the equation $(f\ \text{TENSE}) = \text{PRS}$. By way of example, the full (minimal) set of equations that describes the f-structure of (1) is provided in (3).

(3) $(f\ \text{PRED}) = \text{`SEE}\langle(f\ \text{SUBJ})\ (f\ \text{OBJ})\rangle\text{'}$
$(f\ \text{TENSE}) = \text{PRS}$
$(f\ \text{ASPECT}) = \text{PERF}$
$(f\ \text{SUBJ}) = g$
$(f\ \text{OBJ}) = h$

$(g\ \text{PRED}) = \text{`JOHN'}$
$(g\ \text{PERS}) = 3$
$(g\ \text{NUM}) = \text{SG}$

$(g\ \text{PRED}) = \text{`DAVID'}$
$(g\ \text{PERS}) = 3$
$(g\ \text{NUM}) = \text{SG}$

Sets of equations as in (3) are called F-DESCRIPTIONS. A valid f-structure of a sentence is an f-structure that *minimally* satisfies this sentence's f-description. Thus, the f-structure displayed in (1) is the minimal f-structure that satisfies (3); were one to add the attribute-value pair [MOOD INDICATIVE], (3) would still be satisfied, but the structure would no longer be minimal.

Since an f-structure function application produces attribute values, and, as seen in (1) and (3), these values can also be f-structures, it is possible to use nested function applications. Thus, since $(f\ \text{SUBJ}) = g$, $((f\ \text{SUBJ})\ \text{PERS})$ is equivalent to $(g\ \text{PERS})$ and has the value 3. By convention, function application is left associative, thus the parentheses can be omitted and the equation written

[5] The term *f(unctional)-structure* can thus be understood in two ways: as a structure representing the "function" of words and phrases (as opposed to c-structure which represents "form") and, more formally, as a *function* proper. This set-theoretic understanding of f-structures is standard in the LFG literature, but f-structures can alternatively be modeled in terms of graph theory; an example of this approach is found in Kuhn (2003).

as (f SUBJ PERS) = 3. Early on, LFG has also adopted an extension of function application called functional uncertainty (Kaplan & Zaenen 1989b), which allows replacing the right-hand side of the function application (the "path" of attribute names) by a regular expression; thus, (f COMP* SUBJ) denotes the value of the attribute SUBJ of f or an f-structure embedded in any number of COMP attributes within f. For a formal definition of functional uncertainty and a more detailed discussion, see Belyaev forthcoming(a) [this volume].

While it is possible to describe individual f-structures using sets of equations as in (3), it is obvious that such a system cannot serve as a basis for any regular description of grammar, as it lacks a way of specifying the mapping from words or phrases to the f-structures that represent them. In LFG, this task is mediated through c-structure; f-descriptions for individual sentences are constructed on the basis of ANNOTATED C-STRUCTURE RULES, which are described in the next section.

4.2 Annotated c-structure rules

The formal metalanguage introduced in Section 4.1 provides a way to describe abstract syntactic representations, but, used by itself, it does not allow describing actual grammars and making generalizations about linguistic notions. This is because f-structures should also be mapped to the building blocks of sentence structure – words and c-structure nodes – in a regular way. In other words, the correspondence function ϕ, introduced in Section 2, has to be defined. In LFG, this is done using ANNOTATED PHRASE STRUCTURE RULES. These rules contain additional statements, formulated in the functional description metalanguage, that specify the mapping from each node to the f-structure. In order to refer to the f-structure projections, the equations use the following additional notations:

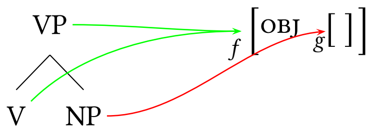
(4) the current c-structure node: $*$
the immediately dominating c-structure node: $\hat{*}$

Using this notation, we can formulate phrase structure rules like the following:

(5) $$\text{VP} \longrightarrow \underset{\phi(\hat{*}) = \phi(*)}{\text{V}} \quad \underset{(\phi(\hat{*})\ \text{OBJ}) = \phi(*)}{\text{NP}}$$

In (5), the annotation for V stands for "this node (V) maps to the same f-structure as the dominating node (VP)", while the annotation for NP stands for "this node (NP) maps to the OBJ attribute of the f-structure of the dominating node (VP)". The mapping that this rule defines is illustrated in (6). The nodes VP and V map to the same f-structure, labeled as f, while NP maps to the f-structure labeled as g – the direct object of the clause.

(6)



For convenience, $\phi(*)$ and $\phi(\hat{*})$ are usually replaced by the abbreviations ↓ (pronounced "down") and ↑ (pronounced "up"), respectively. These metavariables are assumed to be the only way to refer to material up or down the tree in phrase structure rules; direct reference to "low-level" variables such as $*$ is generally not used in LFG analyses. The conventional representation of the rule in (5) is given in (7).

(7) VP ⟶ V NP
↑=↓ (↑ OBJ)=↓

In the standard model of c-structure, lexical entries are nothing more than rules defining a preterminal node dominating a terminal node. However, they use a slightly different notation, where the word form is followed by its category and annotation, illustrated in (8).

(8) *John* N (↑ PRED)='JOHN'
(↑ PERS)=3
(↑ NUM)=SG

Since there is no further material down the tree, lexical entries typically only use the metavariable ↑ to provide information associated with the preterminal node. In some cases, ↓ is also used to draw subtle distinctions between information contributed by the word itself and the information contributed by the preterminal. For example, Zaenen & Kaplan (1995: 230) ingeniously map the verbal form to the PRED value, while other grammatical features are assumed to be contributed by the V node. In practice, this possibility is seldom used.

The projection function ϕ maps c-structure nodes to f-structures, but one may also define an inverse correspondence ϕ^{-1} to proceed in the opposite direction. This function provides the set of c-structure nodes that map to the f-structure given as its argument. Note that the inverse projection is not a function, as the f- to c-structure relation is one-to-many. Inverse projections are used in f-descriptions in order to use c-structure features in f-structure constraints. For example, to check that the subject's f-structure maps to an NP, one may use the equation $\text{NP} \in \text{CAT}\big((\uparrow\text{SUBJ})^{-1}\big)$. This is seldom needed, because, by design, most constraints on f-structure attributes can be described solely in terms of f-structure. However, sometimes the inverse projection is indispensable, e.g. when formulating the notion of f-precedence (see Kaplan & Zaenen 1989a; also see Belyaev forthcoming(a) [this volume]) describing linear order conditions on anaphora (Rákosi forthcoming [this volume]).

4.3 Well-formedness conditions

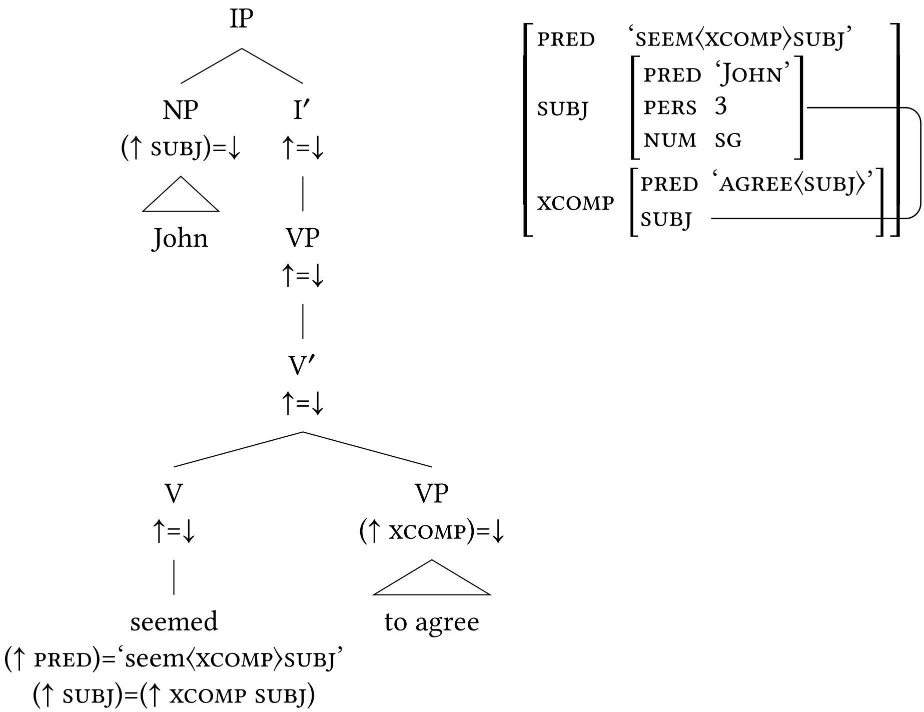
There are three conditions that any f-structure must satisfy in order to be treated as valid: Uniqueness (also known as Consistency), Completeness, and Coherence. Any f-structure that violates these conditions cannot be part of a valid analysis of any sentence. Uniqueness requires that each attribute have exactly one value – this actually follows from the notion of f-structure as a function, since a function, by definition, is a many-to-one or one-to-one mapping. Completeness requires that each argument listed in the PRED value of an f-structure (which is the locus of valency information) is present in the f-structure; Coherence, complementarily, requires that no extra arguments *not* listed in the PRED value are introduced. For more detail on how these conditions actually operate, see Belyaev forthcoming(a) [this volume].

4.4 Structure sharing and "movement"

Unlike transformation-based grammatical approaches, LFG has no special formal mechanism such as movement or Internal Merge to handle dependencies between different structural positions. The closest equivalent to such a mechanism is STRUCTURE SHARING, which consists in one f-structure being the value of two or more distinct attributes. The possibility of structure sharing follows from the general makeup of the formalism: If f-structures are functions and features are their arguments, it is expected that these structures are reentrant: a function can return the same value for different arguments. Since reentrancy is obviously required for the simplest cases such as reentrant atomic values, structure sharing is only a natural consequence of this property.

A classic example of the use of structure sharing to describe a movement-like process is the LFG analysis of raising. Raising verbs such as English *seem* are analyzed as having a non-thematic subject that is shared with the subject of the complement clause:

(9)



This correctly predicts that the raised subject appears as the argument of the matrix clause while being subcategorized for and assigned a semantic role in the complement clause. For more detail on control and raising, see Vincent forthcoming [this volume].

It is important to note that while structure sharing is, in formal terms, the closest counterpart to movement in LFG, this does not mean that all phenomena that are treated via movement in transformational frameworks should involve structure sharing in LFG. This is because movement is normally the *only* mechanism for “non-canonical” or “displaced” positioning of material in transformational frameworks, while LFG draws a crucial distinction between c- and f-structure. Two sentences may differ in the c-structure while having the same f-structure – this is called SCRAMBLING and this is the most widespread mechanism of syntactic “displacement” in non-configurational languages or languages that allow mapping to the same grammatical function in different positions. For example, Arka (2003) proposes the following rule for S in Balinese:

(10) $\text{S} \longrightarrow \left\{ \begin{array}{c} \text{VP} \\ \uparrow=\downarrow \end{array} \middle| \begin{array}{c} \text{NP} \\ (\uparrow \text{GF})=\downarrow \end{array} \right\}^{*}$

This allows any number of NPs to alternate with any number of VPs in any order; each NP may be freely assigned to any grammatical function. Therefore, sentences with the same predicate and the same set of NP arguments will have identical f-structures, with the only difference being found at c-structure. But no c-structure configuration will be considered as "basic" in any formal sense of the term.[6]

5 Additional levels of projection

C-structure and f-structure were originally thought of as the only levels of grammar in LFG: c-structure as a kind of "form" representation, and f-structure as a "functional" representation, in some sense reflecting semantics and having a degree of universality compared to c-structure. It quickly became clear, however, that these two levels are not enough to represent the full complexity of grammatical phenomena. First, semantics should be separate from f-structure to handle phenomena that are not represented in syntax, such as quantifier scope. Second, f-structure in its standard form is a collection of information of different types: purely morphological and morphosyntactic atomic features; grammatical functions; valency information (PRED features); and semantic information (if features such as ANIM are used to describe effects of animacy on grammatical marking). Third, f-structure simply cannot handle some phenomena, like prosody, which require a different kind of structure whose constituents are not equivalent to either c-structure constituents or f-structures.

A possible way to overcome these difficulties would be to extend the role of the existing c- and f-structure, which would mirror similar developments in transformational grammar, with its central role of constituent structure and the proliferation of functional projections (see Sells forthcoming [this volume]). However, the architecture of LFG permits a more elegant solution. While the original system does only consist of c- and f-structure, there is nothing intrinsic about this binarity: the two are connected by a projection function ϕ that maps nodes to f-structure. It is possible to define other functions that would connect c- or f-structures to various other structures; thus, where $\phi(*)$ (abbreviated $\downarrow$) stands for

[6]Of course, even in non-configurational languages, certain word orders are often viewed as less marked compared to others. This is probably due to differences in information structure, which in modern LFG literature is usually treated as a separate level that may interact with other levels such as c-structure, f-structure, and prosody (Zaenen forthcoming [this volume]; see also Dalrymple & Nikolaeva 2011). Crucially, an information structure difference between sentences does not automatically entail any difference at either c- or f-structure.

the f-structure of the annotated node, $\mu(*)$ would be the morphosyntactic structure (m-structure) of this c-structure node, and $\sigma(\phi(*))$ (abbreviated $\uparrow_\sigma$) would be the semantic structure (s-structure) that the f-structure that corresponds to this node maps to (if s-structure is viewed as projected from f-structure). The simultaneous description of two or more grammatical structures by the same rule or lexical entry is called CODESCRIPTION, which is the main principle governing the interaction of levels in LFG.

This modularity has been successfully used to model a number of grammatical levels, such that LFG, as it is currently practiced, is no longer centered around the interaction between c- and f-structure, although these still play a major role as the main syntactic representations. It is also crucial that LFG, by design, still retains a degree of "syntactocentricity" in that all additional projections are defined with reference to c-structure nodes. This is different from the notion of a truly parallel architecture advocated e.g. in Culicover & Jackendoff (2005), where each level of representation (specifically, in their model, syntax and semantics) is conceived of as a separate "combinatorially autonomous" system that is linked to other levels via a system of correspondence constraints. In LFG, only c-structure is combinatorial in this sense,[7] with possible trees defined directly through phrase structure rules; the content of other projections is not autonomously generated, but defined through phrase structure annotations that connect the elements of these projections to c-structure nodes. Thus, while c-structure is not as central as constituent structure in other frameworks, it acts as a "hub" that connects all the different levels of sentence structure together.[8]

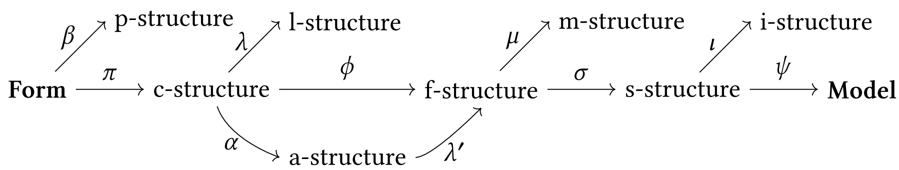
There is currently no agreed-upon set of representational levels. Some, like s-structure or prosodic structure, are almost universally adopted and consistently interpreted in terms of projection. Others, like information structure (i-structure),

[7] C-structure rules are somewhat less central in approaches like Halvorsen (1983) and Andrews (2008), which use description by analysis, rather than the standard codescription approach, to describe the syntax-semantics interface: In these approaches, meaning is constructed on the basis of f-structure, without direct reference to c-structure. Even here, however, semantics is not a separate combinatorial system but is constructed on the basis of another structure which, in turn, is projected from c-structure; this still seems rather different from Culicover and Jackendoff's vision of parallel architecture.

[8] This flavour of syntactocentricity is far less radical than in mainstream generative grammar and may in fact be unavoidable in a (broadly) lexicalist framework, inasmuch as words are viewed as the "building blocks" of sentences. In fact, I am not aware of a fully developed and formalized implementation of any truly parallel architecture. There is no way around the fact that phonetic form is the only part of language that is directly available for perception; thus the part of grammar that is tasked with combining such "surface" elements into complete utterances – i.e. syntax in the narrow sense – will always have a special role.

are assumed by most authors, but specific interpretations vary: for example, i-structure is projected from c-structure in King (1997); Butt & King (1997), but from s-structure in the more recent proposal of Dalrymple & Nikolaeva (2011). Finally, some levels are specific to particular approaches and are not universally adopted, e.g. morphosyntactic structure (m-structure), viewed as projected from c-structure (Butt et al. 2004; Butt, Fortmann, et al. 1996) or f-structure (Sadler & Nordlinger 2004); or argument structure (a-structure), which is used in some approaches to argument mapping (Butt et al. 1997) but is viewed as redundant in some more recent proposals such as (Asudeh & Giorgolo 2012; Asudeh et al. 2014; Findlay 2016). One version of how the correspondence architecture might look is provided in (11).[9]

(11)



To date, additional levels and projections that have been discussed and described in the LFG literature include the following (references to some of the proposals are given in parentheses; most have separate chapters in the handbook, which describe proposed representations in detail):

- argument structure (a-structure) (Butt et al. 1997), see Findlay & Kibort forthcoming [this volume];
- semantic structure (s-structure) (Dalrymple 1999), see Asudeh forthcoming [this volume];
- information structure (i-structure) (King 1997; Butt & King 1997; Dalrymple & Nikolaeva 2011), see Zaenen forthcoming [this volume];
- prosodic structure (p-structure) (Dalrymple & Mycock 2011; Bögel 2012), see Bögel forthcoming [this volume];
- morphological / morphosyntactic structure (m-structure), see (Butt et al. 2004; Sadler & Nordlinger 2004), Asudeh et al. forthcoming [this volume];

[9] The argument structure projection functions α and λ' are from the proposal in Butt et al. (1997). In this approach, which is not universally accepted in the literature, the projection function ϕ is the composition $\alpha \circ \lambda'$. I use the label λ' to distinguish this from the projection function λ that maps c-structure to l-structure, specifying category labels (Lowe & Lovestrand 2020).

- grammatical marking structure (g-structure) (Falk 2006);
- l-structure, a level that represents complex categories of c-structure nodes in the approach of Lowe & Lovestrand (2020): see Belyaev forthcoming(a) [this volume].

6 Conclusion

In this chapter, I have described the main architectural notions of LFG – the c- and f-structures. LFG can be viewed as incorporating the best features of constituent-structure-based (at c-structure) and dependency-based (at f-structure) frameworks, while avoiding their main drawbacks. Frameworks that use phrase structure as the only syntactic representation require additional mechanisms such as transformations, multiple dominance or separate linearization to properly capture word order variation and feature constraints; LFG manages to keep c-structure relatively simple due to the fact that all feature interactions are captured at f-structure, without referring to constituent structure positions. At the same time, the fact that f-structure does not directly refer to individual words or phrase structure nodes allows adequately capturing word order variation while keeping predicate-argument representations fairly uniform across languages. I have also described how the core architecture may be extended to other projections beyond f-structure. Each of these modules captures a separate part of grammar (prosody, semantic structure, information structure, etc.) and has its own internal makeup. The modules are linked together using annotations of c-structure rules in the same way as f-structure is projected from c-structure. Hence, grammars in LFG are factorized into several distinct components, each of which is responsible for its own range of phenomena and largely operates according to its own principles, with c-structure serving as a “hub” tying all the components together.

Acknowledgements

I am grateful to Mary Dalrymple for her insightful comments on this chapter and painstaking editorial work. I would also like to thank Tina Bögel for some comments on an earlier draft of the chapter. Finally, I am grateful to two anonymous reviewers for their attentive reading of the chapter and useful comments. This research has been supported by the Interdisciplinary Scientific and Educational School of Moscow University “Preservation of the World Cultural and Historical Heritage”.

References

Andrews, Avery D. 2008. The role of PRED in LFG+Glue. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '08 conference*, 46–67. Stanford, CA: CSLI Publications.

Andrews, Avery D. Forthcoming. Clause structure and configurationality. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Arka, I Wayan. 2003. *Balinese morphosyntax: A lexical-functional approach*. Canberra: Pacific Linguistics.

Asudeh, Ash. Forthcoming. Glue Semantics. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Asudeh, Ash & Gianluca Giorgolo. 2012. Flexible composition for optional and derived arguments. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '12 conference*, 64–84. Stanford, CA: CSLI Publications.

Asudeh, Ash, Gianluca Giorgolo & Ida Toivonen. 2014. Meaning and valency. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '14 conference*, 68–88. Stanford, CA: CSLI Publications.

Asudeh, Ash, Paul B. Melchin & Daniel Siddiqi. Forthcoming. Morphology. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Belyaev, Oleg. Forthcoming(a). Core concepts of LFG. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Belyaev, Oleg. Forthcoming(b). Grammatical functions: their classification and properties. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Blackburn, Patrick & Claire Gardent. 1995. A specification language for Lexical-Functional Grammars. In *Proceedings of the 7th conference of the European chapter of the ACL (EACL 1995)*, 39–44. Association for Computational Linguistics. DOI: 10.3115/976973.976980.

Bögel, Tina. 2012. The p-diagram — a syllable-based approach to p-structure. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '12 conference*, 99–117. Stanford, CA: CSLI Publications.

Bögel, Tina. Forthcoming. Prosody and its interfaces. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Bresnan, Joan, Ash Asudeh, Ida Toivonen & Stephen Wechsler. 2016. *Lexical-Functional Syntax*. 2nd edn. (Blackwell Textbooks in Linguistics 16). Malden, MA: Wiley-Blackwell.

Bresnan, Joan & Sam A. Mchombo. 1995. The lexical integrity principle: Evidence from Bantu. *Natural Language & Linguistic Theory* 13(2). 181–254. DOI: 10.1007/bf00992782.

Butt, Miriam, Mary Dalrymple & Anette Frank. 1997. An architecture for linking theory in LFG. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '97 conference*, 1–16. Stanford, CA: CSLI Publications.

Butt, Miriam, Christian Fortmann & Christian Rohrer. 1996. Syntactic analyses for parallel grammars. In *COLING '96: Proceedings of the 16th Conference on Computational Linguistics*, 182–187. Copenhagen: International Committee on Computational Linguistics. DOI: 10.3115/992628.992662.

Butt, Miriam & Tracy Holloway King. 1997. Null elements in discourse structure. Unpublished manuscript intended for publication in the Proceedings of the NULLS Seminar. http://ling.uni-konstanz.de/pages/home/butt/main/papers/nulls97.pdf.

Butt, Miriam, María-Eugenia Niño & Frédérique Segond. 2004. Multilingual processing of auxiliaries within LFG. In Louisa Sadler & Andrew Spencer (eds.), *Projecting morphology*. Stanford, CA: CSLI Publications. Previously published as Butt, Niño & Segond (1996).

Butt, Miriam, María-Eugenia Niño & Frederique Segond. 1996. Multilingual processing of auxiliaries in LFG. In D. Gibbon (ed.), *Natural language processing and speech technology: Results of the 3rd KONVENS conference*, 111–122. Berlin: Mouton de Gruyter.

Carpenter, Robert L. 1992. *The logic of typed feature structures*. Cambridge, UK: Cambridge University Press. DOI: 10.1017/cbo9780511530098.

Chomsky, Noam. 1957. *Syntactic structures*. The Hague: Mouton. DOI: 10.1515/9783110218329.

Chomsky, Noam. 1970. Remarks on nominalization. In Roderick A. Jacobs & Peter S. Rosenbaum (eds.), *Readings in English transformational grammar*, 184–221. Waltham, MA: Ginn.

Culicover, Peter W. & Ray Jackendoff. 2005. *Simpler Syntax*. Oxford: Oxford University Press. DOI: 10.1093/acprof:oso/9780199271092.001.0001.

Dalrymple, Mary (ed.). 1999. *Semantics and syntax in Lexical Functional Grammar: The resource logic approach* (Language, Speech, and Communication). Cambridge, MA: The MIT Press. DOI: 10.7551/mitpress/6169.001.0001.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell III & Annie Zaenen (eds.). 1995. *Formal issues in Lexical-Functional Grammar*. Stanford, CA: CSLI Publications.

Dalrymple, Mary & Louise Mycock. 2011. The prosody-semantics interface. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '11 conference*, 173–193. Stanford, CA: CSLI Publications.

Dalrymple, Mary & Irina Nikolaeva. 2011. *Objects and information structure* (Cambridge Studies in Linguistics). Cambridge, UK: Cambridge University Press.

Falk, Yehuda N. 1983. Constituency, word order, and phrase structure rules. *Linguistic Analysis* 11. 331–360.

Falk, Yehuda N. 2006. *Subjects and universal grammar: An explanatory theory*. Cambridge, UK: Cambridge University Press. DOI: 10.1017/cbo9780511486265.

Findlay, Jamie Y. 2016. Mapping theory without argument structure. *Journal of Language Modelling* 4(2). 293–338. DOI: 10.15398/jlm.v4i2.171.

Findlay, Jamie Y. & Anna Kibort. Forthcoming. Argument structure and mapping theory. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

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

Jackendoff, Ray. 1977. *$\bar{X}$ syntax: A study of phrase structure* (Linguistic Inquiry Monographs 2). Cambridge, MA: The MIT Press.

Jackendoff, Ray. 2002. *Foundations of language: Brain, meaning, grammar, evolution*. Oxford: Oxford University Press.

Kaplan, Ronald M. 1989. The formal architecture of Lexical-Functional Grammar. *Journal of Information Science and Engineering* 5. 305–322. Revised version published in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 7–27).

Kaplan, Ronald M. & Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan (ed.), *The mental representation of grammatical relations*, 173–281. Cambridge, MA: The MIT Press. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 29–130).

Kaplan, Ronald M. & Annie Zaenen. 1989a. Functional precedence and constituent structure. In Chu-Ren Huang & Keh-Jiann Chen (eds.), *Proceedings of ROCLING II*, 19–40. Taipei.

Kaplan, Ronald M. & Annie Zaenen. 1989b. Long-distance dependencies, constituent structure, and functional uncertainty. In Mark Baltin & Anthony Kroch (eds.), *Alternative conceptions of phrase structure*, 17–42. Chicago: University of Chicago Press. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 137–165).

King, Tracy Holloway. 1997. Focus domains and information structure. In Miriam Butt & Tracy Holloway King (eds.), *Proceedings of the LFG '97 conference*, 2–13. Stanford, CA: CSLI Publications.

Kuhn, Jonas. 2003. *Optimality-Theoretic Syntax – A declarative approach*. Stanford, CA: CSLI Publications.

Lowe, John J. & Joseph Lovestrand. 2020. Minimal phrase structure: A new formalized theory of phrase structure. *Journal of Language Modelling* 8(1). 1–51. DOI: 10.15398/jlm.v8i1.247.

Potts, Christopher. 2002. *Comparative economy conditions in natural language syntax*. North American Summer School in Logic, Language, and Information 1, workshop on model-theoretic syntax, Stanford University.

Pullum, Geoffrey K. & Gerald Gazdar. 1982. Natural languages and context-free languages. *Linguistics and Philosophy*. 471–504. DOI: 10.1007/bf00360802.

Rákosi, György. Forthcoming. Anaphora. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Sadler, Louisa & Rachel Nordlinger. 2004. Relating morphology to syntax. In Louisa Sadler & Andrew Spencer (eds.), *Projecting morphology*, 159–186. Stanford, CA: CSLI Publications.

Sells, Peter. Forthcoming. LFG and Minimalism. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Speas, Margaret. 1990. *Phrase structure in natural language*. Dordrecht: Kluwer Academic Publishers. DOI: 10.1007/978-94-009-2045-3.

Vincent, Nigel. Forthcoming. Raising and control. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Woods, William A. 1970. Transition network grammars for natural language analysis. *Communications of the ACM* 13(10). 591–606. DOI: 10.1145/355598.362773.

Zaenen, Annie. Forthcoming. Information structure. In Mary Dalrymple (ed.), *Handbook of Lexical Functional Grammar*. Berlin: Language Science Press.

Zaenen, Annie & Ronald M. Kaplan. 1995. Formal devices for linguistic generalizations: West Germanic word order in LFG. In Jennifer S. Cole, Georgia M. Green & Jerry L. Morgan (eds.), *Linguistics and computation*, 3–27. Stanford, CA: CSLI Publications. Reprinted in Dalrymple, Kaplan, Maxwell & Zaenen (1995: 215–240).