Lexical Functional Grammar

Ida Toivonen
Carleton University

Mary Dalrymple
University of Oxford

February 8, 2025

Note: A shorter version of this chapter appears in the *International Encyclopedia of Language and Linguistics*, 2026, Elsevier.

Key Points/Objectives

Lexical Functional Grammar (LFG): LFG is a linguistic theory that represents language through a rich lexicon and separate but related structures that model syntax and other linguistic levels such as semantics, information structure and prosody.

Modularity: Different kinds of linguistic information are represented via separate but related structures which may be represented in different ways.

Syntax: LFG adopts two basic syntactic structures, constituent structure (c-structure), which models phrase structure, and functional structure (f-structure), which models grammatical functions and phenomena such as binding and agreement.

Cross-Linguistic Application: The LFG framework is designed to capture linguistic variation as well as universal tendencies.

Computational and Psychological Relevance: LFG strives to be computationally tractable and psychologically plausible.

1 Introduction

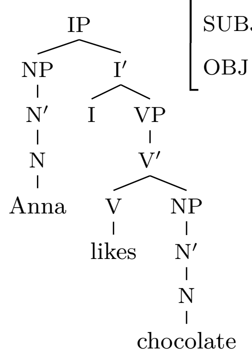
Lexical Functional Grammar (LFG) is a linguistic theory which studies the various facets of linguistic structure and the relations between them. Traditional LFG analyses focus on two syntactic structures. **Constituent structure** (c-structure) represents word order and phrasal groupings, and **functional structure** (f-structure) represents grammatical functions like subject and object. These structures have separate representations, but are related to each other in systematic ways. Recent LFG work includes investigations of prosodic structure, semantic structure, and other linguistic structures and their relation to c-structure and f-structure.

2 LFG's syntactic structures

Lexical Functional Grammar is a theory of the structure of language and how different aspects of linguistic structure are related. As the name implies, the theory is **lexical**: the lexicon is richly structured. Lexical relations rather than transformations or operations on phrase structure trees capture grammatical generalizations. LFG is also **functional**: grammatical functions like subject and object are primitives of the theory, and they are not defined in terms of phrase structure configuration or semantic roles.

LFG adopts two core syntactic levels: f(unctional)-structure and c(onstituent)-structure. F-structure is modeled with feature structures and c-structure is modeled with phrase structure trees. F-structure represents abstract grammatical functions like subject and object as well as abstract features like tense and case. C-structure models the overt expression of these relations, governed by language-particular constraints on word order and phrase structure. An example is given in (1).

(1) *Anna likes chocolate.*



$$\begin{bmatrix} \text{PRED} & \text{‘LIKE}\langle\text{SUBJ, OBJ}\rangle\text{’} \\ \text{TENSE} & \text{PRESENT} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{‘ANNA’}\end{bmatrix} \\ \text{OBJ} & \begin{bmatrix}\text{PRED} & \text{‘CHOCOLATE’}\end{bmatrix} \end{bmatrix}$$

This dual representation is intended to enable the modelling of cross-linguistic commonalities as well as differences. Languages vary greatly in word order and phrasal structure, and the theory of constituent structure allows for this variation within certain universally-defined parameters. In contrast, all languages draw from the same functional vocabulary. According to LFG's theory of functional structure, the abstract syntactic structure of every language is organized in terms of subject, object, and other grammatical functions, most of which are familiar from traditional grammatical work.

The relation between c-structure and f-structure is captured by mapping functions relating parts of one structure to parts of the other. For example, the [Spec, IP] in the c-structure tree is related to the subject f-structure by means of a function which relates nodes of the c-structure tree to parts of the f-structure for a sentence. Relations among c-structure, f-structure, and other linguistic levels (e.g., prosodic structure) have also been explored and defined in terms of functional mappings from subparts of one structure to the corresponding subparts of other structures.

The overall formal structure and basic linguistic assumptions of the theory have not changed since its development in the late 1970s by Joan Bresnan, a linguist trained at the Massachusetts Institute of Technology, and Ronald M. Kaplan, a psycholinguist and computational linguist trained at Harvard University. Bresnan (1982) is a collection of influential early papers in LFG; more recent works providing an introduction to LFG include Falk (2001); Kroeger (2004); Bresnan et al. (2016); Dalrymple et al. (2019); and Börjars et al. (2019). For more detailed information on individual topics, see the chapters in the *Handbook of Lexical Functional Grammar* (Dalrymple, 2023).

3 Constituent structure

Like many other linguistic theories, LFG represents word order and phrasal groupings by means of phrase structure trees, which in LFG are called constituent structures (c-structures). One example was given in (1), where the c-structure for *Anna likes chocolate* was given along with an f-structure. The LFG theory of c-structure is quite flexible. The goal is to model universal tendencies while simultaneously capturing linguistic variation.

Languages vary greatly in the basic phrasal expression of sentences. Basic word order can be verb-initial (Malagasy), verb-final (Japanese), or verb-medial (English). Word order correlates with grammatical function in some languages, such as English, in which the subject and other arguments appear in particular phrase structure positions. In other languages, word order is more free, and grammatical functions are identified by casemarking or agreement rather than phrasal configuration: in many languages, there is no specific phrasal position where the subject or object must always appear. Requirements for phrasal groupings also differ across languages. In English, for example, a noun and any adjectives that modify it must appear together and form a phrasal unit. In many other languages, including Latin, this is not necessary, and a noun can be separated from its modifying adjectives in the sentence.

3.1 Constituent structure representation

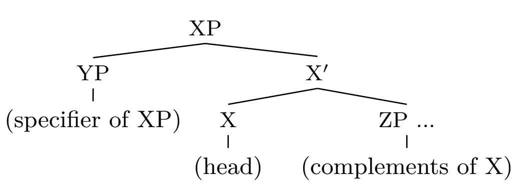
LFG's constituent structure represents word order, phrasal constituenthood, and syntactic categories. Syntactic categories divide into lexical and functional categories. Lexical categories such as N (noun) and V (verb) are familiar from traditional grammatical analysis. Most LFG analyses assume at least the lexical categories N (noun), A (adjective), V (verb), Adv (adverb), and P (preposition), though more or fewer categories may be relevant for a particular language. LFG also adopts a small set of **functional categories**, including I (for Inflection), C (for Complementizer), and D (for Determiner). Functional categories play an organizing role in the syntax, and are either associated with closed-class categories such as complementizers and determiners, or are filled with subtypes of particular lexical categories. The set of lexical and functional categories is universal in the sense that the categories are universally available to languages. However, not all languages make use of all available categories.

Constituent structure is organized according to **X-bar theory** (see **X-bar theory**), which assumes that phrases are internally headed and therefore **endocentric**: a phrase and its head have the same category, but a different bar level. For example, the basic lexical category N is the head of the single bar level category N′ ("N-bar"), which in turn is the head of the two-bar-level category N″ ("N-double-bar"). Similarly, the basic functional category I is the head of I′, which heads I″. Many LFG analyses assume that N″ and I″ are maximal phrases, and as maximal phrases, the categories N″ and I″ are usually written as NP (noun phrase) and IP (the category assigned to a sentence like *Anna likes cookies*). Nonprojecting categories are also assumed (Toivonen, 2003); these are lexical categories that are not heads of phrases, but appear on their own, adjoined to heads. For example, verbal particles (words corresponding to the particle *up* in a sentence like *I woke up the baby*) in some Germanic languages are nonprojecting words, typically prepositions, adjoined to the verb.

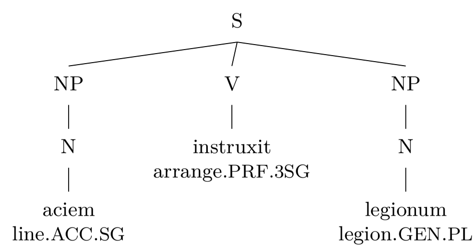
Not all phrases are endocentric. LFG assumes a single exocentric, non-headed category, the category S, which does not obey the constraints of X-bar theory. Not all languages make use of this phrase; it plays no role in the syntax of English, for example. In languages that make use of this phrase, it behaves as a maximal phrase, but it has no c-structure head, and it can dominate phrases of any category or bar level. It does not obey the generalizations which constrain the relation between endocentric c-structure categories and f-structure, and so is often motivated in languages with very free word order, in which morphological information rather than phrase structure position determines the functional role of a phrase. Figure 1 illustrates the difference between languages with flexible phrase structure and the exocentric category S (illustrated here with a Latin example) and highly configurational languages with endocentric structures (illustrated with the corresponding English sentence).

Even in endocentric structures, LFG does not require phrase structure trees to be binary branching, and so there can be more than two daughters of any node in a c-structure tree. The nonhead daughter of a maximal phrase is called its **specifier**, and the nonhead sisters of a lexical category are its **complements**. This is shown schematically in (2):

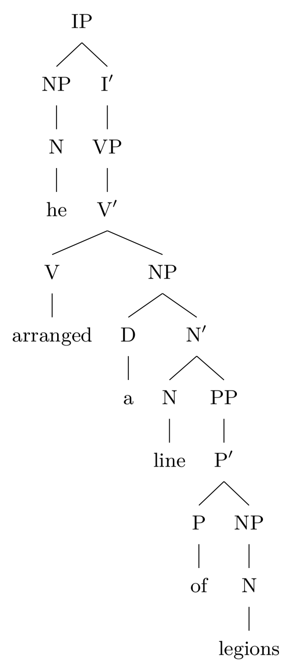
(2)



C (often the host of complementizers) takes an IP complement, and I (often the host of auxiliaries) takes a VP or an S complement. C, I, and V host words that provide information related to the verbal predicate of the clause: the verb itself, auxiliaries and complementizers. In other words, C, I, and V host the verb plus verbal information such as tense, mood, and aspect. Languages differ in exactly how these nodes are utilized. For example, in English, auxiliaries typically appear in I but in questions, they can appear in C. In French, all tensed verbs (not just the auxiliaries) appear in I. In the following, we will not assume DP, which will mean that the category of a phrase like *the boy* is NP.



$$\begin{bmatrix} \text{PRED} & \text{'ARRANGE}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \\ \text{PERS} & 3 \\ \text{NUM} & \text{SG} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{`LINE}\langle\text{OBL}_{\text{GEN}}\rangle\text{'} \\ \text{NUM} & \text{SG} \\ \text{OBL}_{\text{GEN}} & \begin{bmatrix} \text{PRED} & \text{`LEGION'} \\ \text{NUM} & \text{PL} \\ \text{CASE} & \text{GEN} \end{bmatrix} \\ \text{CASE} & \text{ACC} \end{bmatrix} \\ \text{TENSE} & \text{PAST} \\ \text{ASPECT} & \text{PERF} \end{bmatrix}$$



$$\begin{bmatrix} \text{PRED} & \text{`ARRANGE}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \\ \text{PERS} & 3 \\ \text{NUM} & \text{SG} \\ \text{CASE} & \text{NOM} \\ \text{GEND} & \text{MASC} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{`LINE}\langle\text{OBL}_{\text{GEN}}\text{'}\rangle \\ \text{NUM} & \text{SG} \\ \text{OBL}_{\text{GEN}} & \begin{bmatrix} \text{PRED} & \text{`LEGION'} \\ \text{NUM} & \text{PL} \end{bmatrix} \\ \text{DEF} & - \end{bmatrix} \\ \text{TENSE} & \text{PAST} \end{bmatrix}$$

Figure 1: C- and f-structures for two similar sentences in Latin and English (adapted from Dalrymple & Findlay, 2019, 143–144)

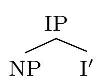
However, there is no general agreement of the status of such phrases in LFG: according to some analyses, *the boy* is a DP rather than an NP in at least some languages.

LFG assumes that all nodes are optional, and nodes that do not contain any lexical material are typically omitted. For example, in a language like French where the tensed verb is in I, the V node can be omitted from the representation since it does not contain any lexical material. The V does not contain an unpronounced copy or trace of the finite verb. In general, C-structure does not contain unpronounced material, such as unpronounced features, nor does it contain null pronominals in pro-drop languages like Italian or Japanese. C-structure also does not contain subparts of words (e.g, tense morphology). Rather, it reflects the structure and grouping of the full syntactic units – the words and phrases – in the sentence.

3.2 Phrase structure rules

LFG draws a strong distinction between the formal objects of the theory — constituent structure trees and functional structures — and the constraints or descriptions involving those objects. C-structure trees are constrained by phrase structure rules, which license local tree configurations. The phrase structure rule in (3a) licenses the c-structure in (3b):

(3) a. IP $\longrightarrow$ NP I′

b. 

The right-hand side of an LFG phrase structure rule is a **regular expression** (see **Formal Languages**), allowing for disjunction, optionality, and arbitrary repetition of a node or sequence of nodes. The V and NP daughters in the rule in (4) are optional, and the Kleene star (*) annotation on the PP indicates that a sequence of zero or more PP constituents may appear.

(4) V′ $\longrightarrow$ (V) (NP) PP*

4 Functional structure

Syntactic analyses in traditional grammatical descriptions are stated in terms of abstract syntactic functions like subject, object, and complement. In LFG, these functions are represented at f-structure. Grammatical features such as tense, case, person, and number are also represented at f-structure.

4.1 Grammatical functions and their representation

In a sentence such as *David devoured a sandwich*, *David* is the subject and *a sandwich* is the object. This information is represented by an **attribute-value structure**, the f-structure, in which the value of the SUBJ feature is the f-structure for the subject, and the value of the OBJ feature is the f-structure for the object.

(5) David devoured a sandwich.

$$\begin{bmatrix} \text{PRED} & \text{`DEVOUR}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`DAVID'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{SPEC} & \text{A} \\ \text{PRED} & \text{`SANDWICH'} \end{bmatrix} \end{bmatrix}$$

For clarity, many of the features and values in this f-structure have been omitted, a practice often followed in LFG presentations. The full f-structure would contain tense, aspect, person, number, and other functional features.

Every content word in a sentence contributes a value for the feature PRED. These values are called **semantic forms**. In the functional structure, semantic forms are surrounded by single quotes: the semantic form contributed by the word *David* is ‘DAVID’.

An important property of semantic forms is that they are uniquely instantiated for each instance of their use, reflecting the unique semantic contribution of each word within the sentence. This is occasionally indicated by associating a unique numerical identifier with each instance of a semantic form, as in (6):

(6) David devoured a sandwich.

$$\left[\begin{array}{ll} \text{PRED} & \text{‘DEVOUR}_{37}\langle\text{SUBJ,OBJ}\rangle\text{’} \\ \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{‘DAVID}_{42}\text{’}\end{array}\right] \\ \text{OBJ} & \left[\begin{array}{ll}\text{SPEC} & \text{A}\\ \text{PRED} & \text{‘SANDWICH}_{14}\text{’}\end{array}\right]\end{array}\right]$$

In (6), the particular occurrence of the semantic form for the word *David* as it is used in this sentence is represented as ‘DAVID_{42}’. Another use of *David* would be associated with a different unique identifier, perhaps ‘DAVID_{73}’. Representing semantic forms with explicit numerical identifiers clearly shows the unique contribution that content words make to the f-structure. However, the identifiers are often not displayed in order to avoid clutter.

A verb or other predicate generally requires a particular set of arguments: for example, the verb *devoured* requires a subject (SUBJ) and an object (OBJ). These arguments are said to be **governed** by the predicate; equivalently, the predicate is said to **subcategorize for** its arguments. The semantic form contributed by a verb or other predicate contains information about the arguments it governs. As shown above, the governed arguments appear in angled brackets: ‘DEVOUR⟨SUBJ,OBJ⟩’.

The LFG requirements of Completeness and Coherence ensure that all and only the grammatical functions governed by a predicate are found in the f-structure of a grammatically acceptable sentence. For example, the unacceptability of example (7) shows that the verb *devoured* cannot appear without an OBJ:

(7) *David devoured.

This sentence violates the principle of Completeness, according to which every grammatical function governed by a predicate must be filled. Here, the OBJ is not present, and the sentence is incomplete.

Furthermore, *devour* cannot appear with other functions besides the grammatical functions SUBJ and OBJ that it governs. Example (8) shows that it cannot appear with a sentential complement in addition to its object:

(8) *David devoured a sandwich that it was raining.

This sentence violates the principle of Coherence, according to which only the grammatical functions that are governed by a predicate can appear. Since the sentence contains a grammatical function that the verb *devour* does not govern, it is incoherent.

The grammatical functions that a predicate can govern are called **governable grammatical functions**. The inventory of universally-available governable grammatical functions is given in Table 1. Languages differ as to which of these functions are relevant, but in many languages, including English, all of these functions are used.

Table 1:	Governable grammatical functions
SUBJ:	subject
OBJ:	object
COMP:	sentential or closed (nonpredicative) infinitival complement
XCOMP:	an open (predicative) complement, often infinitival, whose SUBJ function is externally controlled
OBJ_θ:	a family of secondary OBJ functions associated with a particular, language-specific set of thematic roles
OBL_θ:	a family of thematically restricted oblique functions such as OBL_{GOAL} or $\text{OBL}_{\text{AGENT}}$

The subject and object functions are directly adopted from traditional linguistics, but the other functions are unique to LFG or have a LFG-specific definition. The X in XCOMP indicates that it is an open function whose SUBJ is supplied externally. Languages differ in which thematically restricted secondary objects they allow, and may not allow them at all. English only has $\text{OBJ}_{\text{THEME}}$. Further, OBL_θ functions are expressed as PPs in English, while other languages may express them as NPs with oblique case marking. Examples of the different grammatical functions are are provided in (9–14).

(9) **Sandra** left. (SUBJECT)

(10) Carlos ate **a cookie**. (OBJECT)

(11) They thought **that the bus was late**. (COMP)

(12) We tried **to leave**. (XCOMP)

(13) The grandparents gave Lisa **a present**. (OBJ_θ)

(14) The grandparents gave a present **to Lisa**. (OBL_{GOAL})

Not all phrases fill argument positions of a predicate. Modifying adjunct phrases are not required by a predicate and hence are not governable. In (15), the phrase *yesterday* bears the nongovernable grammatical function ADJ(unct):

(15) David devoured a sandwich yesterday.

There are two nongovernable grammatical functions. The function ADJ is the grammatical function of modifiers like *in the park*, *with a hammer*, and *yesterday*. The function XADJ is the grammatical function of open predicative adjuncts whose subject is externally controlled. The phrase filling the XADJ role is underlined in (16).

(16) a. Having opened the window, David took a deep breath.

b. David ate the celery naked.

c. David ate the celery raw.

In (16a) and (16b), the open adjunct XADJ is controlled by the matrix subject *David*. In (16c), the XADJ is controlled by the object: it is the celery that is raw.

Unlike governable grammatical functions, more than one adjunct function can appear in a sentence:

(17) David devoured a sandwich **[at noon]** **[yesterday]**.

Since the ADJ function can be multiply filled, its value is a **set** of f-structures:

(18) David devoured a sandwich at noon yesterday.

$$\left[\begin{array}{ll}
\text{PRED} & \text{`DEVOUR}\langle\text{SUBJ,OBJ}\rangle\text{'} \\
\text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`DAVID'}\end{array}\right] \\
\text{OBJ} & \left[\begin{array}{ll}\text{SPEC} & \text{A} \\ \text{PRED} & \text{`SANDWICH'}\end{array}\right] \\
\text{ADJ} & \left\{\begin{array}{l}
\left[\begin{array}{ll}\text{PRED} & \text{`YESTERDAY'}\end{array}\right] \\
\left[\begin{array}{ll}\text{PRED} & \text{`AT}\langle\text{OBJ}\rangle\text{'} \\ \text{OBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`NOON'}\end{array}\right]\end{array}\right]
\end{array}\right\}
\end{array}\right]$$

The same is true of XADJ: more than one XADJ phrase can appear in a single sentence:

(19) **Having opened the window**, David ate the celery **naked**.

Hence, the value of the XADJ feature is also a set of f-structures.

The f-structures that have been presented so far have included only a subset of their functional features. In fact, it is common in LFG literature to display only those features that are relevant to the analysis under discussion, since a full representation would often be too unwieldy. A full f-structure for these sentences would contain at least the features and values listed in Table 2, and probably other language-specific features and values as well. Table 2 contains some simplifications; for example, the PRES.IMPERFECT value for aspect is an abbreviation for an f-structure representing a complex description of sentential aspect.

Table 2: Functional structure features

	Feature	Value
Person:	PERS	1, 2, 3
Gender:	GEND	MASC, FEM, ...
Number:	NUM	SG, DUAL, PL, ...
Case:	CASE	NOM, ACC, ...
Surface form:	FORM	Surface word form
Verb form:	VFORM	PASTPART, PRESPART,...
Complementizer form:	COMPFORM	THAT, WHETHER,...
Tense:	TENSE	PRES, PAST,...
Aspect:	ASPECT	PRES.IMPERFECT,...
Pronoun type:	PRONTYPE	REL, WH, PERS,...

The values given in Table 2 are the ones that are most often assumed, but some authors have argued for a different representation of the values of some features. For example, Dalrymple & Kaplan (2000) argue for a set-based representation of the PERS and GEND features to allow for an account of feature resolution in coordination, and of the CASE feature to allow for case indeterminacy.

4.2 Functional descriptions

As with c-structures, we draw a sharp distinction between f-structures and their descriptions. The set of f-structure constraints associated with the analysis of some sentence is called a **functional description** or **f-description**.

To refer to the value of a feature in some f-structure, we use an expression like the following:

(20) Reference to the value of the TENSE feature in f:

$(f \text{ TENSE})$

This expression refers to the value of the TENSE feature in the f-structure f. If we want to specify the value of that feature, we use an expression like:

(21) $(f \text{ TENSE}) = \text{PAST}$

This **defining equation** specifies that the feature TENSE in the f-structure f has the value PAST.

We can also specify that a feature has a particular f-structure as its value. The expression in (22) specifies that the value of the SUBJ feature in f is the f-structure g:

(22) $(f \text{ SUBJ}) = g$

Some features take as their value a **set** of functional structures. For example, since any number of adjuncts can appear in a sentence, the value of the feature ADJ is a set. We can specify that an f-structure h is a member of the ADJ set with the following constraint, using the set-membership symbol $\in$:

(23) $h \in (f \text{ ADJ})$

The constraints discussed so far are called **defining constraints**, since they define the required properties of a functional structure. An abbreviated f-description for a sentence like *David sneezed* is:

(24) $(f\ \mathrm{PRED}) = \text{‘SNEEZE}\langle\mathrm{SUBJ}\rangle\text{’}$
$(f\ \mathrm{TENSE}) = \mathrm{PAST}$
$(f\ \mathrm{SUBJ}) = g$
$(g\ \mathrm{PRED}) = \text{‘DAVID’}$

This f-description holds of the following f-structure, where the f-structures are annotated with the names used in the f-description (24):

(25) David sneezed.

$$f\!:\left[\begin{array}{ll} \mathrm{PRED} & \text{‘SNEEZE}\langle\mathrm{SUBJ}\rangle\text{’} \\ \mathrm{TENSE} & \mathrm{PAST} \\ \mathrm{SUBJ} & g\!:\left[\begin{array}{ll} \mathrm{PRED} & \text{‘DAVID’} \end{array}\right] \end{array}\right]$$

The f-description in (24) also holds of the f-structure in (26), which contains the attributes and values that are mentioned in in (24) as well as some additional information:

(26)
$$f\!:\left[\begin{array}{ll} \mathrm{PRED} & \text{‘SNEEZE}\langle\mathrm{SUBJ}\rangle\text{’} \\ \mathrm{TENSE} & \mathrm{PAST} \\ \mathrm{SUBJ} & g\!:\left[\begin{array}{ll} \mathrm{PRED} & \text{‘DAVID’} \\ \mathrm{PERS} & 3 \end{array}\right] \\ \mathrm{ADJ} & \left\{\begin{array}{l} \left[\begin{array}{ll} \mathrm{PRED} & \text{‘YESTERDAY’} \end{array}\right] \\ \left[\begin{array}{ll} \mathrm{PRED} & \text{‘AT}\langle\mathrm{OBJ}\rangle\text{’} \\ \mathrm{OBJ} & \left[\begin{array}{ll} \mathrm{PRED} & \text{‘NOON’} \end{array}\right] \end{array}\right] \end{array}\right\} \end{array}\right]$$

However, we require the f-structure solution for a particular f-description to be the **minimal solution** to the f-description: no additional attributes or values that are not mentioned in the f-description are included. Thus, the correct solution to the f-description in (24) is the f-structure in (25), not the larger one in (26). Formally, the solution to an f-description is the most general f-structure that satisfies the f-description, which subsumes all other (larger) f-structures that satisfy the f-description.

Besides the defining constraints described above, LFG also allows elements of the f-description to check the properties of the minimal solution to the defining equations. The expression in (27) is a **constraining equation**, distinguished from a defining equation by the c subscript on the equals sign in the expression:

(27) Constraining equation:

$(f\ \mathrm{SUBJ\ NUM}) =_c \mathrm{SG}$

When this expression appears, the f-structure f that is the minimal solution to the defining equations must contain the feature SUBJ whose value has a feature NUM with value SG. The constraining equation in (27) does not hold of the f-structure in (25), since in that f-structure, the value of the NUM feature has been left unspecified, and the SUBJ of f does not have a NUM feature with value SG.

In contrast, the functional description in (28a) for the sentence *David sneezes* has a well-formed solution, the f-structure in (28b):

(28) a. $(f\ \mathrm{PRED}) = \text{‘SNEEZE}\langle\mathrm{SUBJ}\rangle\text{’}$
$(f\ \mathrm{TENSE}) = \mathrm{PRES}$
$(f\ \mathrm{SUBJ}) = g$
$(g\ \mathrm{PRED}) = \text{‘DAVID’}$
$(g\ \mathrm{NUM}) = \mathrm{SG}$
$(f\ \mathrm{SUBJ\ NUM}) =_c \mathrm{SG}$

b.
$$f\!:\left[\begin{array}{ll} \mathrm{PRED} & \text{‘SNEEZE}\langle\mathrm{SUBJ}\rangle\text{’} \\ \mathrm{TENSE} & \mathrm{PRES} \\ \mathrm{SUBJ} & g\!:\left[\begin{array}{ll} \mathrm{PRED} & \text{‘DAVID’} \\ \mathrm{NUM} & \mathrm{SG} \end{array}\right] \end{array}\right]$$

Here, the value SG for the NUM feature for g is specified in the second-to-last line of the functional description. Thus, the f-structure in (28b) satisfies the defining constraints given in the first five lines of (28a). Moreover, it satisfies the constraining equation given in the last line of (28a).

We can also place other requirements on the minimal solution to the defining equations in some f-description. The expression in (29a) requires f not to have the value PRESENT for the feature TENSE, which can happen if f has no TENSE feature, or if f has a TENSE feature with some value other than PRESENT. When it appears in a functional description, the expression in (29b) is an **existential constraint**, requiring f to contain the feature TENSE, but not requiring any particular value for this feature. We can also use a **negative existential constraint** to require an f-structure not to contain a feature, as in (29c), which requires f not to contain the feature TENSE with any value whatsoever.

(29) a. Negative equation: $(f\ \text{TENSE}) \neq \text{PRESENT}$

b. Existential constraint: $(f\ \text{TENSE})$

c. Negative existential constraint: $\neg(f\ \text{TENSE})$

Functional descriptions can also be stated in terms of the Boolean operations of conjunction, disjunction, and negation. In the f-descriptions given above, we implicitly assumed that the constraints in the f-description are interpreted conjunctively: if an f-description contains more than one requirement, each requirement must hold. LFG also allows disjunctions and negations of sets of requirements. For example, a verb like *sneeze* contributes the following f-description:

(30) *sneeze*

$$
\begin{array}{l}
(f\ \text{PRED}) = \text{`SNEEZE}\langle\text{SUBJ}\rangle\text{'} \\
\{(f\ \text{VFORM}) = \text{BASE}\ | \\
\ \ (f\ \text{TENSE}) = \text{PRES} \\
\ \ \neg\{(f\ \text{SUBJ PERS}) = 3 \\
\ \ \ \ \ (f\ \text{SUBJ NUM}) = \text{SG}\}\ \}
\end{array}
$$

Disjunction is indicated by curly brackets, with the alternatives separated by a vertical bar |. Negation for a set of requirements is represented by prefixing ¬, and the scope of negation is indicated by curly brackets. This lexical entry allows two possibilities. The first is for the base form of the verb, in which the value of the VFORM feature is BASE. For the second possibility, the value of the feature TENSE is PRES for present tense, and a third-person singular subject is disallowed by negating the possibility for the PERS feature to have value 3 when the NUM feature has value SG.

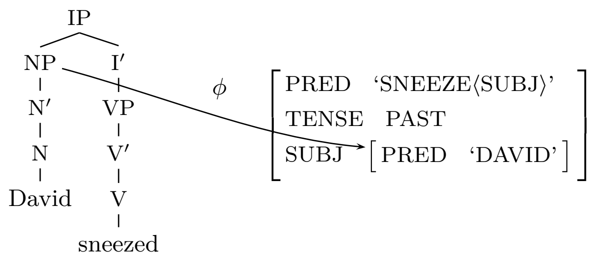
5 The Constituent Structure-Functional Structure Relation

There are clear crosslinguistic regularities relating constituent structure positions to grammatical functions. In particular, phrases and their heads are required to correspond to the same f-structure, and specifier and complement positions are associated with particular grammatical functions. Such generalizations constrain the relation between c-structure positions and the f-structure positions they are associated with.

5.1 Structural correspondences

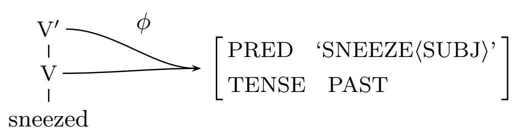
To express these generalizations formally, relating nodes in the c-structure tree and the f-structures they correspond to, we can define a function called ϕ (phi) relating nodes of the c-structure tree to parts of the f-structure for a sentence. In (31), the ϕ function from the NP node to the f-structure it corresponds to is represented by an arrow and labeled ϕ:

(31) *David sneezed.*



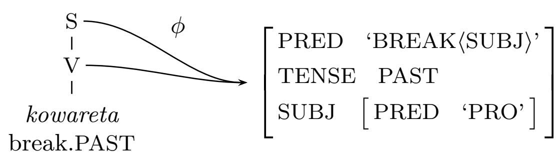
Each node of the c-structure tree corresponds to some part of the f-structure. As shown in (32), more than one c-structure node can correspond to the same f-structure (the ϕ function is **many-to-one**):

(32)



Further, there can be f-structures that have no corresponding c-structure node (the ϕ function is **into**). Example (33) shows the c-structure and f-structure for a sentence of Japanese, a so-called pro-drop language in which the verb optionally specifies functional information about its subject. When there is no overt subject phrase in the sentence, the information specified by the verb supplies the SUBJ value for the sentence. In (33), since there is no overt subject, all of the information about the subject comes from specifications on the verb, and there is no c-structure node corresponding to the SUBJ f-structure:

(33) *koware-ta*
break-PAST
‘[It/Something] broke.’

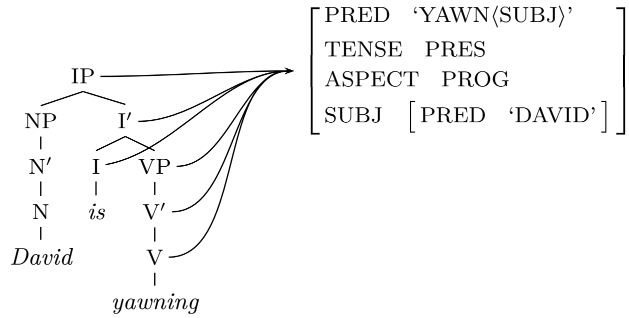


Japanese is a *radical* or *discourse* pro-drop language, which is different from regular, morphological pro-drop languages such as, for example, Italian. Morphological pro-drop languages contribute information (such as, e.g., PERSON and NUMBER) about the omitted argument through bound morphology on the verb. The functional descriptions that specify the f-structure information are associated with the relevant bound morphology. See Toivonen (2023) for discussion of pro-drop in LFG.

5.2 Constituent Structure-Functional Structure Correspondences

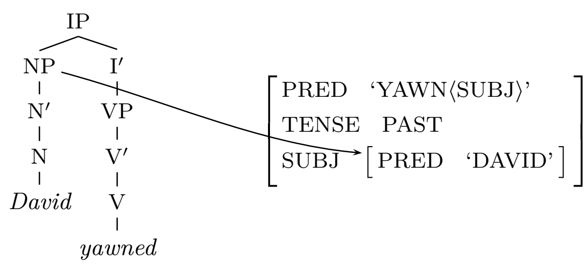
The ϕ function is important in stating universally valid relations between c-structure positions and the functional roles associated with them. For example, a phrase and its head always correspond to the same f-structure. Furthermore, the complement of a functional category is an **f-structure co-head**: the functional head and its complement correspond to the same f-structure. This is shown in (34), where the functional category IP, its heads I′ and I, and its complement VP map to the same f-structure.

(34) *David is yawning.*



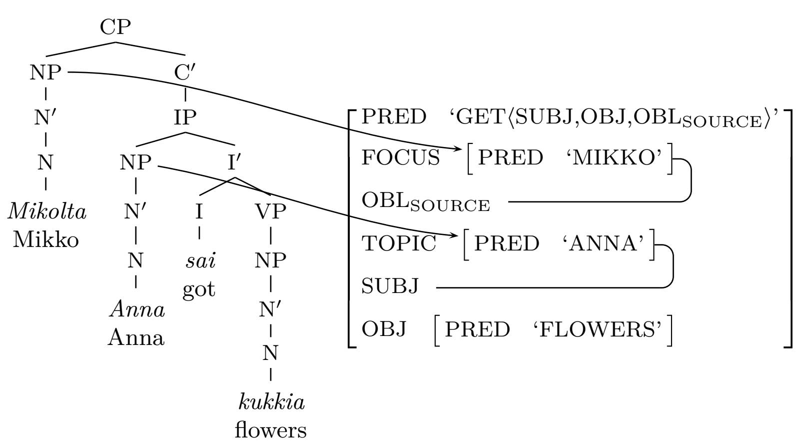
The specifier position of the functional categories IP and CP is filled by a phrase bearing a **grammaticized discourse function**: SUBJ, TOPIC, or FOCUS. Within these limits, languages can differ as the particular grammaticized discourse function allowed in each of these positions. In English, as we have seen, the specifier position of IP is filled by the SUBJ:

(35) *David yawned.*



In Finnish, the specifier of IP is associated with the TOPIC function, and the specifier of CP is associated with FOCUS:

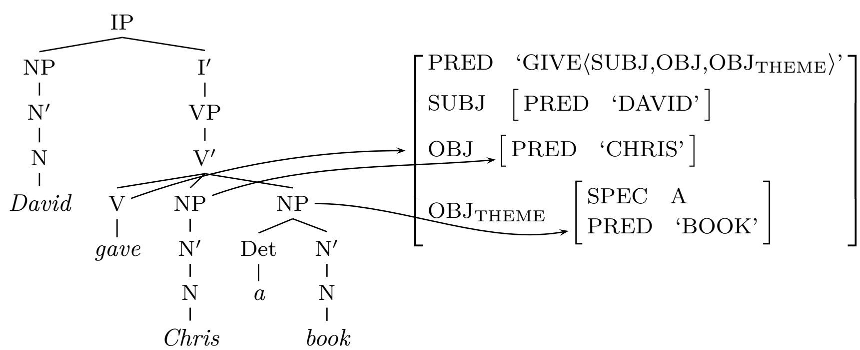
(36) *Mikolta Anna sai kukkia.*
Mikko Anna got flowers
‘From Mikko, Anna got flowers.’



When an f-structure contains a FOCUS or TOPIC function, the **Extended Coherence Condition** requires it to be integrated into the f-structure by either anaphorically or functionally binding another f-structure in the sentence. In a sentence like *Bill, I like him*, the f-structure for *Bill* anaphorically binds the f-structure *him*: the two phrases *Bill* and *him* are syntactically independent, and each phrase has its own f-structure, but the anaphoric relation between the two satisfies the Extended Coherence Condition. In (36), the FOCUS also bears the $\text{OBL}_{\text{SOURCE}}$ function, and the TOPIC is also the SUBJ: these relations involve functional binding, because the same f-structure fills both functions.

The complements of a lexical category bear non-discourse grammatical functions: that is, any grammatical function other than SUBJ, FOCUS, or TOPIC. In (37), the complements of V are associated with the grammatical functions OBJ and $\text{OBJ}_{\text{THEME}}$:

(37) *David gave Chris a book.*



5.3 Constraining the Constituent Structure-Functional Structure Relation

In describing the relation between c-structure and f-structure, we use the following symbols for the f-structure corresponding to the current node in a phrase structure rule and the f-structure of its mother node:

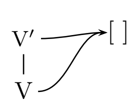
(38) the f-structure of the immediately dominating node: $\uparrow$
the f-structure of the current c-structure node: $\downarrow$

We can use these symbols to annotate the V′ phrase structure rule with f-structure correspondence constraints:

(39) $$\text{V}' \longrightarrow \begin{array}{c} \text{V} \\ \uparrow = \downarrow \\ \text{mother's f-structure} = \text{self's f-structure} \end{array}$$

This annotated rule licenses the configuration in (40). In the c-structure, the V′ node dominates the V node, as the phrase structure rules require. The V′ and V nodes correspond to the same f-structure, as the annotations on the V node require.

(40)

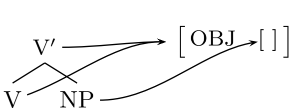


In the rule shown in (41), the $\uparrow=\downarrow$ annotation on the V node specifies that the V and the V′ node correspond to the same f-structure. The annotation on the NP node requires the f-structure $\downarrow$ corresponding to the NP to be the value of the OBJ value in the f-structure $\uparrow$ for the mother node.

(41) $$\text{V}' \longrightarrow \begin{array}{c} \text{V} \\ \uparrow = \downarrow \end{array} \quad \begin{array}{c} \text{NP} \\ (\uparrow \ \text{OBJ}) = \downarrow \end{array}$$

The rule in (41) licenses the following configuration:

(42)



We can use the same formal vocabulary in specifications of lexical entries. The lexical entry for the verb *sneezed* is shown in (43). It specifies that the c-structure category of *sneezed* is V, and also specifies constraints on the f-structure $\uparrow$ of the preterminal V node that dominates the terminal node *sneezed*:

(43) *sneezed* V ($\uparrow$ PRED) = ‘SNEEZE⟨SUBJ⟩’
($\uparrow$ TENSE) = PAST

This lexical entry licenses the c-structure/f-structure configuration in (44).

(44)

$$\begin{array}{c} \text{V} \\ | \\ \text{sneezed} \end{array} \longrightarrow \begin{bmatrix} \text{PRED} & \text{‘SNEEZE⟨SUBJ⟩’} \\ \text{TENSE} & \text{PAST} \end{bmatrix}$$

In the examples above, expressions such as ($\uparrow$ OBJ) refer to a particular f-structure, the value of the attribute OBJ in the f-structure $\uparrow$. It is also possible to refer to f-structures that are embedded within another f-structure at an arbitrary depth, by the use of **functional uncertainty**, where a regular expression (defined above, in the discussion of phrase structure rules) is used to characterize a path through an f-structure. The rule shown in (45) states that the NP daughter of IP bears the FOCUS role (($\uparrow$ FOCUS) = $\downarrow$) and also bears some grammatical function GF which may be embedded within any number of COMPs (($\uparrow$ COMP* GF) = $\downarrow$). Such a rule would play a role in the analysis of examples like (36), where the same phrase fills both the FOCUS and the SUBJ role.

(45)

$$\text{IP} \longrightarrow \begin{array}{c} \text{NP} \\ (\uparrow \text{ FOCUS}) = \downarrow \\ (\uparrow \text{ COMP* GF}) = \downarrow \end{array} \quad \begin{array}{c} \text{I}' \\ \uparrow = \downarrow \\ \end{array}$$

6 Syntax and semantics

Several research strands in LFG have explored the relation of constituent and functional structure to meaning. Section 6.1 provides an overview on LFG research on argument structure, and Section 6.2 presents the “glue” approach to the interface between syntax and semantics.

6.1 Mapping theory and argument linking

Mapping theory explores correlations between the semantic roles of the arguments of a predicate and their syntactic functions. If a language assigns the syntactic function SUBJ to the agent argument of an active verb like *kick*, for example, it invariably assigns SUBJ to the agent argument of semantically similar verbs like *hit*. The locus for this information is argument structure (a-structure): a syntactic structure with strong ties to the lexicon.

Early formulations of the rules of mapping theory proposed rules relating specific thematic roles to specific grammatical functions: for example, that the thematic role of AGENT is always realized as SUBJ. Later work proposed more general rules relating thematic roles to classes of grammatical functions, rather than specific functions. It is most often assumed that grammatical functions are cross-classified with the features $\pm$R and $\pm$O. Several versions of mapping theory have been proposed (Bresnan & Kanerva, 1989; Bresnan & Zaenen, 1990; Bresnan et al., 2016; Kibort, 2014); in the following, we sketch the theory of Bresnan & Zaenen (1990). Findlay et al. (2023) provide a thorough overview of different LFG approaches to a-structure.

The feature $\pm$R distinguishes **restricted** (+R) **grammatical functions from unrestricted** ($-$R) functions. The grammatical functions SUBJ and OBJ are classified as unrestricted, meaning that they can be filled by an argument bearing any thematic role. These contrast with restricted grammatical functions like obliques or thematically restricted objects, which must be filled by arguments with particular thematic roles: for example, the $\text{OBL}_{\text{SOURCE}}$ function must be filled by an argument bearing the thematic role SOURCE, and the thematically restricted object function $\text{OBJ}_{\text{THEME}}$ is filled by a THEME argument.

The feature $\pm$O distinguishes **objective** (+O) grammatical functions from **nonobjective** ($-$O) functions. The unrestricted OBJ function and the restricted OBJ_θ functions are objective, while the SUBJ and the OBL_θ functions are nonobjective.

These features cross-classify the grammatical functions as in (46):

(46)

	−R	+R
−O	SUBJ	OBL_θ
+O	OBJ	OBJ_θ

These features are used to state rules of **intrinsic classification** of particular thematic roles. Such rules constrain the relation between thematic roles and the classes of grammatical functions that these features delineate. For example, arguments bearing the AGENT role are classified as intrinsically nonobjective (−O), either SUBJ or $\mathrm{OBL}_{\mathrm{AGENT}}$. Arguments bearing the THEME role are disjunctively classified, either as intrinsically unrestricted (−R), bearing the SUBJ or OBJ function, or as intrinsically objective (+O), filling the OBJ or $\mathrm{OBJ}_{\mathrm{THEME}}$ function.

Besides these intrinsic classifications, **default mapping rules** classify the arguments of a predicate according to their relative position on the thematic hierarchy:

(47) Thematic hierarchy (Bresnan & Kanerva, 1989):

AGENT > BENEFACTIVE > RECIPIENT/EXPERIENCER
> INSTRUMENT > THEME/PATIENT > LOCATIVE

One of the default mapping rules requires the argument of a predicate that is highest on the thematic hierarchy to be classified as unrestricted (−R). For example, if a verb requires an AGENT argument and a PATIENT argument, the AGENT argument thematically outranks the PATIENT argument, and thus the AGENT argument is classified as unrestricted.

For a predicate with an AGENT and a PATIENT argument, like *kick*, this has the following result (Bresnan & Kanerva, 1989):

(48)

	kick ⟨	AGENT	PATIENT	⟩
		|	|	
intrinsic:		[−O]	[−R]	
defaults:		[−R]		
		SUBJ	SUBJ/OBJ	
Final classification:		SUBJ	OBJ	

The AGENT argument is classified as intrinsically nonobjective. The default rules add the unrestricted classification to the thematically highest argument, the AGENT. Since the AGENT is classified as [−O,−R], it is the SUBJ. The unrestricted classification of the PATIENT argument allows it to bear either the SUBJ or the OBJ role, but since the AGENT is assigned the SUBJ role, the PATIENT must be realized as OBJ. Thus, the argument classification rules, together with well-formedness conditions like the Subject Condition requiring each verbal predicate to have a subject, constrain the mapping between argument roles and grammatical functions.

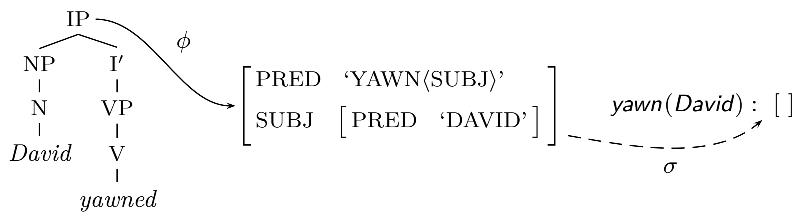
6.2 Glue: The syntax-semantics interface

The dominant theory of the syntax-semantics interface in LFG is the so-called **glue** approach (Dalrymple, 1999, 2001), a theory of how syntax guides the process of semantic composition. LFG assumes that the syntactic level that is primarily involved in semantic composition is the functional structure. The glue approach assumes that each part of the f-structure corresponds to a semantic resource associated with a meaning, and that the meaning of an f-structure is obtained by assembling the meanings of its parts according to a set of instructions specifying how the semantic resources can combine. These assembly instructions are provided as a set of logical premises in the “glue language” of **linear logic**, and the derivation of a meaning for a sentence corresponds to a logical deduction.

The deduction is performed on the basis of logical premises contributed by the words in the sentence (and possibly by syntactic constructions). Linear logic, a resource-based logic, is used to state requirements on how the meanings of the parts of a sentence can be combined to form the meaning of the sentence as a whole. Linear logic is different from classical logic in that it does not admit rules that allow for premises to be discarded or used more than once in a deduction. Premises in a linear logic deduction are, then, resources that must be accounted for in the course of a deduction; this nicely models the semantic contribution of the words in a sentence, which must contribute exactly once to the meaning of the sentence, and may not be ignored or used more than once. A sentence like *David knocked twice* cannot mean simply *David knocked*: the meaning of *twice* cannot be ignored. It also cannot mean the same thing as *David knocked twice twice*; the meaning of a word in a sentence cannot be used multiple times in forming the meaning of the sentence.

The syntactic structures for the sentence *David yawned*, together with the desired semantic result, are displayed in (49):

(49) *David yawned.*



The semantic structure for the sentence is related to its f-structure by the correspondence function σ, represented as a dotted line. This result is obtained on the basis of the following lexical information, associated with the verb *yawned*:

(50) $\lambda X.\mathit{yawn}(X) : (\uparrow \text{ SUBJ})_\sigma \multimap {\uparrow_\sigma}$

This formula is called a **meaning constructor**. It pairs the meaning for *yawned*, the one-place predicate $\lambda X.\mathit{yawn}(X)$, with the linear logic formula $(\uparrow \text{ SUBJ})_\sigma \multimap {\uparrow_\sigma}$. In this formula, the connective $\multimap$ is the *linear implication* symbol of linear logic. This symbol expresses a meaning similar to *if...then...*; in this case, stating that *if* a semantic resource $(\uparrow \text{ SUBJ})_\sigma$ representing the meaning of the subject is available, *then* a semantic resource $\uparrow_\sigma$ representing the meaning of the sentence can be produced. Unlike the implication operator of classical logic, the linear implication operator $\multimap$ carries with it a requirement for *consumption* and *production* of semantic resources: the formula $(\uparrow \text{ SUBJ})_\sigma \multimap {\uparrow_\sigma}$ indicates that if a semantic resource $(\uparrow \text{ SUBJ})_\sigma$ is found, it is *consumed* and the semantic resource $\uparrow_\sigma$ is *produced*.

We also assume that a name like *David* contributes a semantic resource, its semantic structure. In an example like *David yawned*, this resource is consumed by the verb *yawned*, which requires a resource for its SUBJ to produce a resource for the sentence. This accords with the intuition that the verb in a sentence must obtain a meaning for its arguments in order for a meaning for the sentence to be available.

The f-structure for the sentence *David yawned*, together with the instantiated meaning constructors contributed by *David* and *yawned*, is given in (51):

(51)
$$y:\left[\begin{array}{ll}\text{PRED} & \text{‘YAWN}\langle\text{SUBJ}\rangle\text{’}\\ \text{SUBJ} & d:\left[\text{PRED} \quad \text{‘DAVID’}\right]\end{array}\right]$$

$$\begin{array}{ll}[\textbf{David}] & \mathit{David} : d_\sigma\\ [\textbf{yawn}] & \lambda X.\mathit{yawn}(X) : d_\sigma \multimap y_\sigma\end{array}$$

The left-hand side of the meaning constructor labeled **[David]** is the proper noun meaning *David*, and the left-hand side of the meaning constructor labeled **[yawn]** is the meaning of the intransitive verb *yawned*, the one-place predicate $\lambda X.\mathit{yawn}(X)$.

We must also provide rules for how the right-hand (glue) side of each of the meaning constructors in (51) relates to the left-hand (meaning) side in a meaning deduction. For simple, nonimplicational meaning constructors like **[David]** in (51), the meaning on the left-hand side is the meaning of the semantic structure on the right-hand side. For meaning constructors which contain the linear implication operator $\multimap$, like **[yawn]**, *modus ponens* on the glue side corresponds to function application on the meaning side:

(52)
$$\frac{X : f_\sigma \qquad P : f_\sigma \multimap g_\sigma}{P(X) : g_\sigma}$$

With these correspondences between linear logic formulas and meanings, we perform the following series of reasoning steps:

(53)

$\mathsf{David} : d_\sigma$	The meaning David is associated with the SUBJ semantic structure d_σ.
$\lambda X.\mathsf{yawn}(X) : d_\sigma \multimap y_\sigma$	On the glue side, if we find a semantic resource for the SUBJ d_σ, we consume that resource and produce a semantic resource for the full sentence y_σ. On the meaning side, we apply the function $\lambda X.\mathsf{yawn}(X)$ to the meaning associated with d_σ.
$\mathsf{yawn}(\mathsf{David}) : y_\sigma$	We have produced a semantic structure for the full sentence y_σ, associated with the meaning $\mathsf{yawn}(\mathsf{David})$.

By using the function application rule and the meaning constructors for *David* and *yawned*, we deduce the meaning $\mathsf{yawn}(\mathsf{David})$ for the sentence *David yawned*, as desired.

Glue analyses of a variety of linguistic phenomena have been explored; see Asudeh (2022, 2023) for overviews, and Asudeh (2022) for a comprehensive inventory of work conducted in the glue framework.

7 The General Architecture

The c-structure and f-structure model of syntax and the notion of a rich lexicon have remained stable since the inception of LFG. Other areas of language have also been modeled, often with the goal of exploring how those areas interact with syntax, but there are still debates about the formal and theoretical nature of non-syntactic structures. Glue Semantics (Section 6.2) has become the standard for modelling compositionality and syntax-semantics mapping, but a variety of formalisms have been used to yield the model-theoretic meaning. For example, Asudeh & Toivonen (2012) adopt event semantics, and Belyaev & Haug (2014) make use of Partial Compositional DRT. A-structure is also fairly standardly adopted, but the details of the structure and mapping rules are debated. Some proposals in fact develop a model where a-structure is eliminated, and information about argument roles is instead captured at s-structure, modeled with event semantics (Asudeh & Giorgolo, 2012; Findlay, 2020).

Many recent LFG projects reach beyond syntax and semantics. LFG models language with parallel levels of grammatical structures. The different levels (projections) of language are independent from each other, but the mappings between them are explicit and important. The LFG architecture does not dictate how different levels should be formalized and there is no underlying assumption that each level should be modeled using the same formalism or theoretical principles. Instead, each level is described with whatever tools are judged to be most suitable for the relevant phenomena. It is not always easy to determine what tools are most suitable. As mentioned above, several formal approaches to semantics have been explored. There have also been multiple proposals for how to incorporate morphology, phonology (including prosody) and discourse information. We will briefly summarize some of that literature.

It is perhaps surprising that there is no standard approach to morphology within LFG: morphology is taken very seriously in the framework generally and also in many specific analyses. Inflectional morphology is, for example considered as important as c-structural configurations for identifying grammatical functions. Although the syntactic information contributed by word-internal morphology figures prominently in many syntactic and semantic LFG analyses, authors often stay agnostic on how words are formed. There are however, specific proposals that have drawn on different morphological approaches. Finite State Morphology (Beesley & Karttunen, 2003) is standardly used in computational implementations of LFG use (Butt et al., 1999). Sadler & Nordlinger (2004); Dalrymple (2015); Thomas (2021) and others have incorporated Paradigm Function Morphology

(Stump, 2001) into LFG. Melchin et al. (2020) and others have developed a novel framework, Lexical-Realizational Functional Grammar ($\mathrm{L_R FG}$), which combines Distributed Morphology and LFG. $\mathrm{L_R FG}$ requires some modifications of LFG's c-structure. Asudeh & Siddiqi (2023) provides an overview of morphology in LFG.

Work on phonology in LFG has mainly focused on prosody, since prosody often influences interpretation. For example, prosody can disambiguate otherwise identical strings. Phonology is analyzed at phonological structure (p-structure). A number of different proposals for p-structure have been developed, and they differ crucially in assumptions about how p-structure maps to syntax and semantics, as discussed in Bögel (2023). Prominent proposals include Dalrymple & Mycock (2011) and Bögel (2015). As Bögel (2023) points out, evaluating the merits of competing theories of p-structure crucially involves taking into account the entire grammatical architecture. She also notes that a comparison of the proposals necessarily involves a careful consideration of language processing more generally, as some proposals assume a certain directionality (either production or comprehension), whereas others do not.

Discourse-based information (e.g., information about what is new or old) has been incorporated into many LFG analyses. One strand of research combines Optimality Theory and LFG to handle the discourse effects on syntax (Choi, 1996; Morimoto, 2000). Early LFG work included this information in the f-structure through the grammaticalized discourse functions TOPIC and FOCUS, which were identified with a grammatical function (Bresnan & Mchombo, 1987), but in current LFG, it is common to adopt a separate i(nformation)-structure projection (King, 1997; Dalrymple & Nikolaeva, 2011; Bögel, 2021). Zaenen (2023) provides a comprehensive overview of work relating to information structure in LFG.

The detailed proposals about individual modules and the mappings between them have inspired research into the nature of the entire architecture (Kaplan 1987; Asudeh 2006; Bögel 2023; and others). Butt et al. (2024a) point out that the LFG architecture is unique in that it allows for holistic, computationally tractable analyses that connect all parts of grammar (prosody, syntax, semantics, pragmatics). Butt et al. illustrate this observation with an analysis of question formation in Urdu, which they implement computationally.

Many LFG researchers routinely implement their analyses computationally. An explicit goal of LFG from the beginning has been to create linguistic analyses that are computationally tractable. This ideal is still very much an active part of the conversation. Wedekind & Kaplan (2020) is a recent paper that discusses the computational tractability of LFG grammars. The ParGram project is a collaboration between groups of researchers writing and implementing grammars for a number of languages, including English, German, Norwegian, Hindi-Urdu and Japanese.

Alongside computational tractability, psychological plausibility was an original motivation for LFG and is still informing research today. Several current projects are directly concerned with how language processing and use fit into the general architecture. Bresnan (2021) develops a proposal that attempts to capture the correlation between usage probabilities and contractions of auxiliaries in English. Broadly put, contractions are more likely to occur in predictable contexts. Bresnan assumes that the lexicon consists of dynamic exemplars, which include detailed memories of usage (Pierrehumbert, 2001). In another project that explores the interaction between usage and linguistic form, Bögel et al. (2024) explore the effects of semantic priming and lexical frequency on acoustic salience. Bögel et al.'s formal analysis differs from Bresnan's, but the phenomena they try to model are similar, and both proposals take advantage of LFG's parallel architecture.

In sum, although syntax and semantics are central in the LFG framework, a complete understanding of many linguistic phenomena crucially depends on an understanding of the interaction of a number of linguistic modules. A growing number of research projects focus on phonology (mainly prosody), morphology, and information structure. Some work further explores how interactions between grammar and language processing can be modeled in LFG. Although there is still disagreement regarding the formal nature of individual projections and the general architecture, different proposals consistently adhere to the view that the grammar consists of a number of relatively independent projections. Each projection is subject to different theoretical principles. The formalism used to model each projection depends on the type of data the projection is concerned with. Moreover, proposals about the nature of different projections and the mapping between them must be clear and explicit enough for analyses to be implemented computationally. The LFG framework thus offers the necessary tools to develop, evaluate and compare proposals about all aspects of grammar and language.

8 Summary and Outlook

Lexical Functional Grammar is a robust linguistic framework that systematically captures the relationship between syntax and other levels of linguistic representation. By distinguishing between constituent and functional structures, LFG accommodates the syntactic diversity of language while maintaining universal grammatical principles. It provides a flexible, non-transformational approach to syntactic representation, ensuring that grammatical functions remain central to linguistic analysis. The theory has expanded beyond its original scope to include phonology, morphology, semantics, argument structure, and information structure, reinforcing its role as a multidimensional linguistic model. LFG continues to be a valuable tool in theoretical, descriptive and applied linguistics. Future research will continue to refine its view of the interfaces between grammatical modules.

Bibliography

Alsina, A. (1993). *Predicate Composition: A Theory of Syntactic Function Alternations.* Ph.D. thesis, Stanford University.

Andrews, A. D., III & Manning, C. D. (1999). *Complex Predicates and Information Spreading in LFG.* Stanford: CSLI Publications.

Asudeh, A. (2004). *Resumption as Resource Management.* Ph.D. thesis, Stanford University.

Asudeh, A. (2006). 'Direct compositionality and the architecture of lfg.' In Butt, M., Dalrymple, M. & King, T. H. (eds.) 'Intelligent Linguistic Architectures: Variations on Themes by Ronald M. Kaplan,' Stanfod: CSLI, 363–387.

Asudeh, A. (2022). 'Glue semantics.' *Annual Review of Linguistics 8*, 321–341. doi: 10.1146/annurev-linguistics-032521-053835.

Asudeh, A. (2023). 'Glue semantics.' In Dalrymple, M. (ed.) 'The handbook of Lexical Functional Grammar,' Language Science Press, Empirically Oriented Theoretical Morphology and Syntax, chapter 15, 651–697. doi:10.5281/zenodo.10185964.

Asudeh, A. & Giorgolo, G. (2012). 'Flexible composition for optional and derived arguments.' In Butt, M. & King, T. H. (eds.) 'Proceedings of the LFG12 Conference,' Stanford: CSLI Publications, 64–84. URL `https://web.stanford.edu/group/cslipublications/cslipublications/LFG/17/papers/lfg12asudehgiorgolo.pdf`.

Asudeh, A. & Siddiqi, D. (2023). 'Morphology in LFG.' In Dalrymple, M. (ed.) 'The handbook of Lexical Functional Grammar,' Language Science Press, Empirically Oriented Theoretical Morphology and Syntax, chapter 19, 855–901. doi:10.5281/zenodo.10185976.

Asudeh, A. & Toivonen, I. (2012). 'Copy raising and perception.' *Natural Language and Linguistic Theory 30*(2), 321–380.

Beesley, K. & Karttunen, L. (2003). *Finite State Morphology.* CSLI Publications.

Belyaev, O. & Haug, D. (2014). 'Pronominal coreference in ossetic correlatives and the syntax-semantics interface.' In Butt & King (2014). URL `https://web.stanford.edu/group/cslipublications/cslipublications/LFG/19/papers/lfg14belyaevhaug.pdf`.

Bod, R. & Kaplan, R. M. (1998). 'A probabilistic corpus-driven model for Lexical-Functional analysis.' In 'COLING/ACL98: Joint Meeting of the 36th Annual Meeting of the ACL and the 17th International Conference on Computational Linguistics,' Montréal: International Committee on Computational Linguistics, 145–151.

Bögel, T. (2015). *The syntax-prosody interface in Lexical Functional Grammar*. Ph.D. thesis, University of Konstanz.

Bögel, T. (2023). 'Prosody and its interfaces.' In Dalrymple, M. (ed.) 'The handbook of Lexical Functional Grammar,' Language Science Press, Empirically Oriented Theoretical Morphology and Syntax, chapter 17, 779–821. doi:10.5281/zenodo.10185970.

Bögel, T., Freiseis, M., Hill, R., Wambach, D. & Zhao, T. (2024). 'Language redundancy and acoustic salience: an account in lfg.' In Butt et al. (2024b), 89–115. URL `https://lfg-proceedings.org/lfg/index.php/main/article/view/60/43`.

Bresnan, J. (1978). 'A realistic transformational grammar.' In Halle, M., Bresnan, J. & Miller, G. A. (eds.) 'Linguistic Theory and Psychological Reality,' Cambridge, MA: The MIT Press, 1–59.

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

Bresnan, J. (2001). *Lexical-Functional Syntax*. Oxford: Blackwell.

Bresnan, J. (2021). 'Formal grammar, usage probabilities, and auxiliary contraction.' *Language 97*(1), 108–150.

Bresnan, J., Asudeh, A., Toivonen, I. & Wechsler, S. (2016). *Lexical-Functional Syntax, Second edition*. Hoboken, NJ: Wiley-Blackwell.

Bresnan, J. & Kanerva, J. M. (1989). 'Locative inversion in Chicheŵa: A case study of factorization in grammar.' *Linguistic Inquiry 20*(1), 1–50. Reprinted in Stowell & Wehrli (1992).

Bresnan, J. & Mchombo, S. A. (1987). 'Topic, pronoun, and agreement in Chicheŵa.' *Language 63*(4), 741–782.

Bresnan, J. & Zaenen, A. (1990). 'Deep unaccusativity in LFG.' In Dziwirek, K., Farrell, P. & Mejías-Bikandi, E. (eds.) 'Grammatical Relations. A Cross-Theoretical Perspective,' Stanford: CSLI Publications, 45–57.

Butt, M. (1995). *The Structure of Complex Predicates in Urdu*. Dissertations in Linguistics. Stanford: CSLI Publications. Revised version of 1993 Stanford University dissertation.

Butt, M., Bögel, T., Zymla, M.-M. & Mumtaz, B. (2024a). 'Alternative questions in urdu: from the speech signal to semantics.' In Butt et al. (2024b), 140–164. URL `https://lfg-proceedings.org/lfg/index.php/main/article/view/65/50`.

Butt, M., Findlay, J. & Toivonen, I. (eds.) (2024b). *On-Line Proceedings of the LFG24 Conference*. Konstanz: PubliKon.

Butt, M. & King, T. H. (eds.) (2014). *On-Line Proceedings of the LFG14 Conference*. Stanford: CSLI Publications. URL `https://web.stanford.edu/group/cslipublications/cslipublications/LFG/19/papers/lfg14belyaevhaug.pdf`.

Butt, M., Niño, M. & Segond, F. (1999). *A Grammar Writer's Cookbook*. Stanford, CA: CSLI Publications.

Butt, M. & Toivonen, I. (eds.) (2020). *On-Line Proceedings of the LFG20 Conference*. Stanford: CSLI Publications.

Bögel, T. (2021). 'Function words at the interface: A two-tier approach.' *Languages 6*(4). doi:10.3390/languages6040197.

Börjars, K., Nordlinger, R. & Sadler, L. (2019). *Lexical-Functional Grammar: An Introduction*. Cambridge: Cambridge University Press.

Choi, H.-W. (1996). *Optimizing Structure in Context: Scrambling and Information Structure*. Ph.D. thesis, Stanford University.

Dalrymple, M. (1993). *The Syntax of Anaphoric Binding*. Number 36 in CSLI Lecture Notes. Stanford: CSLI Publications.

Dalrymple, M. (ed.) (1999). *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*. Cambridge, MA: The MIT Press.

Dalrymple, M. (2001). *Lexical Functional Grammar*, volume 34 of *Syntax and Semantics*. New York: Academic Press.

Dalrymple, M. (2015). 'Morphology in the lfg architecture.' In Butt, M. & King, T. H. (eds.) 'Proceedings of the LFG15 Conference,' Stanford: CSLI Publications, 64–83. URL `https://web.stanford.edu/group/cslipublications/cslipublications/LFG/20/papers/lfg15dalrymple.pdf`.

Dalrymple, M. (ed.) (2023). *Handbook of Lexical Functional Grammar*. Empirically Oriented Theoretical Morphology and Syntax. Berlin: Language Science Press. doi: 10.5281/zenodo.10037797.

Dalrymple, M. & Findlay, J. Y. (2019). 'Lexical functional grammar.' In Kertész, A., Moravcsik, E. & Rákosi, C. (eds.) 'Current Approaches to Syntax: A Comparative Handbook,' De Gruyter Mouton.

Dalrymple, M. & Kaplan, R. M. (2000). 'Feature indeterminacy and feature resolution.' *Language 76*(4), 759–798.

Dalrymple, M., Kaplan, R. M., Maxwell, J. T., III & Zaenen, A. (eds.) (1995). *Formal Issues in Lexical-Functional Grammar*. Stanford: CSLI Publications.

Dalrymple, M., Lowe, J. J. & Mycock, L. (2019). *The Oxford Reference Guide to Lexical Functional Grammar*. Oxford: Oxford University Press.

Dalrymple, M. & Mycock, L. (2011). 'The prosody-semantics interface.' In Butt, M. & King, T. H. (eds.) 'The Proceedings of the LFG11 Conference,' Stanford: CSLI, 173–193. URL `http://cslipublications.stanford.edu/LFG/16/abstracts/lfg11abs-dalrymplemycock.html`.

Dalrymple, M. & Nikolaeva, I. (2011). *Objects and information structure*. Cambridge Studies in Linguistics. Cambridge, UK: Cambridge University Press. doi: 10.1017/cbo9780511993473.

Falk, Y. N. (2001). *Lexical-Functional Grammar: An Introduction to Parallel Constraint-Based Syntax*. Stanford: CSLI Publications.

Findlay, J. (2020). 'Mapping theory and the anatomy of a verbal lexical entry.' In Butt & Toivonen (2020), 127–147. URL `https://web.stanford.edu/group/cslipublications/cslipublications/LFG/LFG-2020/lfg2020-findlay.pdf`.

Findlay, J. Y., Taylor, R. & Kibort, A. (2023). 'Argument structure and mapping theory.' In Dalrymple, M. (ed.) 'The handbook of Lexical Functional Grammar,' Language Science Press, Empirically Oriented Theoretical Morphology and Syntax, chapter 16, 699–778. doi:10.5281/zenodo.10185966.

Kaplan, R. M. (1987). 'Three seductions of computational psycholinguistics.' In Whitelock, P., Wood, M. M., Somers, H. L., Johnson, R. & Bennett, P. (eds.) 'Linguistic Theory and Computer Applications,' Academic Press: London, 149–181. Reprinted in Dalrymple et al. (1995, 339–367).

Kibort, A. (2014). 'Mapping out a construction inventory with (lexical) mapping theory.' In Butt & King (2014), 262–282. URL https://web.stanford.edu/group/cslipublications/cslipublications/LFG/19/papers/lfg14belyaevhaug.pdf.

King, T. H. (1995). *Configuring Topic and Focus in Russian*. Dissertations in Linguistics. Stanford: CSLI Publications. Revised version of 1993 Stanford University dissertation.

King, T. H. (1997). 'Focus domains and information structure.' In Butt, M. & King, T. H. (eds.) 'The Proceedings of the LFG97 Conference,' Stanford: CSLI, 2–13. URL http://csli-publications.stanford.edu/LFG/LFG2-1997/lfg97king.pd.

Kroeger, P. (1993). *Phrase Structure and Grammatical Relations in Tagalog*. Dissertations in Linguistics. Stanford: CSLI Publications. Revised version of 1991 Stanford University dissertation.

Kroeger, P. (2004). *Analyzing Syntax: A Lexical-Functional Approach*. Cambridge, UK: Cambridge University Press.

Levin, L. S. (1986). *Operations on Lexical Forms: Unaccusative Rules in Germanic Languages*. Ph.D. thesis, Massachusetts Institute of Technology. Reprinted by Garland Press, New York, 1988.

Levin, L. S., Rappaport, M. & Zaenen, A. (eds.) (1983). *Papers in Lexical-Functional Grammar*. Bloomington: Indiana University Linguistics Club.

Manning, C. D. (1996). *Ergativity: Argument Structure and Grammatical Relations*. Dissertations in Linguistics. Stanford: CSLI Publications. Revised version of 1994 Stanford University dissertation.

Melchin, P. B., Asudeh, A. & Siddiqi, D. (2020). 'Ojibwe agreement in lexical-realizational functional grammar.' In Butt & Toivonen (2020), 268–288. URL https://csli-publications.stanford.edu/LFG/LFG-2020/lfg2020-mas.pdf.

Mohanan, T. (1994). *Argument Structure in Hindi*. Dissertations in Linguistics. Stanford: CSLI Publications. Reprinted version of 1990 Stanford University dissertation.

Morimoto, Y. (2000). *Discourse Configurationality in Bantu Morphosyntax*. Ph.D. thesis, Stanford University.

Nordlinger, R. (1998). *Constructive Case: Evidence From Australian Languages*. Dissertations in Linguistics. Stanford: CSLI Publications. Revised version of 1997 Stanford University dissertation.

Pierrehumbert, J. B. (2001). 'Exemplar dynamics: Word frequency, lenition and contrast.' In Bybee, J. & Hopper, P. J. (eds.) 'Frequency and the emergence of linguistic structure,' Amsterdam: John Benjamins, 137–157.

Sadler, L. & Nordlinger, R. (2004). 'Relating Morphology to Syntax.' In Louisa Sadler and Andrew Spencer (ed.) 'Projecting Morphology,' Stanford, CA: CSLI Publications, 159–186.

Simpson, J. (1991). *Warlpiri Morphology and Syntax: A Lexicalist Approach*. Dordrecht: Kluwer.

Stowell, T. & Wehrli, E. (eds.) (1992). *Syntax and the Lexicon*, volume 26 of *Syntax and Semantics*. San Diego: Academic Press.

Stump, G. (2001). *Inflectional morphology: A theory of paradigm structure*. Cambridge, UK: Cambridge University Press.

Thomas, A. (2021). *Divergence and mismatches in Portuguese morphology and syntax*. Ph.D. thesis, University of Oxford.

Toivonen, I. (2003). *Non-Projecting Words: A Case Study of Swedish Verbal Particles*. Dordrecht: Kluwer.

Toivonen, I. (2023). 'Pronoun incorporation.' In Dalrymple, M. (ed.) 'The Handbook of Lexical Functional Grammar,' Language Science Press, chapter 13, 563–601.

Wedekind, J. & Kaplan, R. M. (2020). 'Tractable Lexical-Functional Grammar.' *Computational Linguistics 46*(3), 515–569.

Zaenen, A. (2023). 'Information structure.' In Dalrymple, M. (ed.) 'The handbook of Lexical Functional Grammar,' Language Science Press, Empirically Oriented Theoretical Morphology and Syntax, chapter 18, 823–253. doi:10.5281/zenodo.10185972.

Keywords

Constituent structure, c-structure, functional structure, f-structure, open/closed grammatical function, annotated phrase structure rule, completeness, coherence, mapping theory, Glue semantics, linguistic architecture.

Biography

Ida Toivonen is Professor of Linguistics and Cognitive Science at Carleton University. Between 2002 and 2005 she was a member of the Department of Linguistics at the University of Canterbury in New Zealand. She received a BA in Linguistics and French from Brandeis University in 1995, and a PhD in Linguistics from Stanford University in 2001.

Mary Dalrymple is Emerita Professor of Syntax at the University of Oxford. She received a PhD in Linguistics from Stanford University in 1990, and was a member of the research staff at Xerox PARC in Palo Alto, California and a consulting faculty member in the Department of Linguistics and the Symbolic Systems Program at Stanford University until 2002.