

chapter 16

Lexical-Functional Grammar

0. ALTERNATIVE THEORIES

The first fifteen chapters of this textbook are an introduction to syntactic theory from one particular perspective. That is the perspective of the Chomskyan Principles and Parameters (P&P) approach (and its descendant: Minimalism). While a large number of syntacticians (perhaps even a majority) operate using the assumptions and formalisms we've developed so far in this book, not everyone does. In this chapter and the next, we look at two other popular formalisms: *Lexical-Functional Grammar (LFG)* and *Head-Driven Phrase Structure Grammar (HPSG)*.

In many ways, these theories have the same basic goals and assumptions as P&P syntax. LFG and HPSG are considered to be generative grammars, just like P&P. Where all these theories differ is in the precise formulation of the rules and constraints. We will have something to say about choosing among formalisms at the end of chapter 17, but to a great degree it comes down to a matter of the range of phenomena one wants to account for and one's preferred means of formal expression. My inclusion of these approaches is not meant to imply that they are better than P&P, nor to imply that P&P is better than them. They should simply be viewed as alternatives.

As a beginning syntactician, you might wonder why you should bother looking at alternative approaches. After all, a significant part of the literature and research being done in syntax uses the assumptions and formalisms de-

veloped in the first fifteen chapters. Shouldn't we just pick one formalism and stick with it? To be honest, most researchers do just this; they do their work within only one formalism. For example, I do almost all of my research within the P&P approach. But this doesn't mean that I shouldn't be familiar with other formalisms too. An important body of work is conducted in these formalisms, and their results are often directly relevant to work being done in Chomskyan P&P syntax. Being able to interpret work done in these alternative approaches is a very useful skill (and unfortunately, one rarely taught to beginning syntacticians). The results found in other approaches to syntax have often affected the development of P&P theory. For example, the lexical approach to passives (whereby the passive morphology affects the thematic and case assigning properties of the underlying verbal morphology) replaced an earlier purely transformational approach to passives. This idea was borrowed from LFG. Similarly, the idea of feature checking is directly related to the notion of unification found in both LFG and HPSG. So I encourage you to keep an open mind and consider the advantages of being familiar with more than one formalism.

1. C-STRUCTURE

One major part of LFG is almost identical to the approach taken in the rest of the book. This is the idea that the words of a sentence are organized into constituents, which are represented by a tree, and generated by rules. In LFG, these trees are called the *c-structure*, and are roughly equivalent to the S-structure in P&P.¹ Many LFG theorists adopt X-bar theory, but it is not as strictly applied to all languages. For example, LFG posits a flat (VP-less) structure for many VSO languages (Kroeger 1993) and "free word order" or "non-configurational" languages like Warlpiri (Simpson 1991). This said, most LFG c-structures look just like the S-structure trees we have built elsewhere in this book. There is one major exception to this: since there is no movement, there are (for the most part) no traces (nor are there any other null elements).

¹ Because LFG doesn't use transformations, there is no D-structure. Phrase structure rules directly build the c-structure (= S-structure). Displaced items are dealt with in other ways. See below.

2. FUNCTIONS

As you might guess from its name, there are two driving forces in Lexical-Functional Grammar: the lexicon (which we explore in section 3 below) and functions. The notion of function is borrowed from math and computer science. A *function* is a rule that maps from one item to another.² There are really two kinds of functions in LFG, which can be a bit confusing. The first kind are called *grammatical functions* and are things like subject, object, etc. We called these grammatical relations in chapter 4. When a practitioner of LFG talks about functions, they are primarily talking about grammatical functions. The other kind of function refers to the principles that map between the different parts of the grammar, such as the mapping between the c-structure and the structure that represents the grammatical functions. This is called an *f-structure*. We are going to look at grammatical functions here, then turn to the mapping relations in section 4.

In P&P syntax, grammatical functions or relations are read off of the tree. That is, you know what the subject of a sentence is by looking for the NP that is in the specifier of TP. In LFG, grammatical functions are not defined by a tree; instead, they are primitive notions (meaning they can't be derived somehow). Every sentence has an f-structure that represents grammatical functions. In the f-structure, a particular NP will be identified as being the subject of the sentence, quite independent of the tree structure associated with the sentence. In the sentence *Diana loves phonology*, *Diana* is equated with the SUBJ grammatical function. This equation is usually represented in what is called an *Attribute Value Matrix (AVM)*; the item on the left is the attribute or function, the item on the right is the value attributed to that function:

- 1) [SUBJ [PRED³ 'Diana']]

Attributes can have various kinds of values, including other AVMs. For example, the value for the SUBJ function in sentence (2) is itself a matrix containing other functions:

- 2) The professor loves phonology.

² If you are unfamiliar with this notion you might want to consult a good “mathematics for linguists” textbook, such as Allwood, Andersson and Dahl (1977).

³ The term PRED here is a bit confusing, since ‘Diana’ is an argument of the clause. PRED here can be, very loosely, translated as the semantic head of the AVM.

$$3) \left[\begin{array}{c} \dots \\ \text{SUBJ} \end{array} \left[\begin{array}{cc} \text{PRED} & \text{'professor'} \\ \text{DEF} & + \\ \text{NUM} & \text{sng} \end{array} \right] \right]$$

In the embedded AVM, the function PRED tells you the lexical content of the subject NP, DEF tells you if it is definite or not, NUM tells you the number, etc. These are all properties of the subject. You'll notice that more than just grammatical functions are represented in these structures; various kinds of other features (like definiteness, etc.) are as well.

Once you combine all the AVMs for all the parts of a sentence you get the f-structure, containing all the sentence's featural and functional information. Where does all this featural information come from? Most of the information that is combined into the f-structure comes from the lexical entries of all the words in the sentence. The lexicon thus plays an important role in this theory.

3. THE LEXICON

The lexicon is where a lot of the work in LFG is done. All the information that ends up in an f-structure starts out in the lexical entries of the words that compose the sentence.

The rough equivalent to a theta grid in LFG is the *a-structure* or argument structure.⁴ As in P&P syntax, the a-structure is part of the lexical entry for the predicate. Simplifying somewhat, the lexical entry for the inflected verb *loves* is seen in (4). For the moment, ignore the arrows; I'll explain these below in section 4.

$$4) \text{ loves:} \quad V \quad \begin{array}{l} (\uparrow \text{PRED}) = \text{'love } <(\uparrow \text{SUBJ}), (\uparrow \text{OBJ})>' \\ (\uparrow \text{TENSE}) = \text{present} \\ (\uparrow \text{SUBJ NUM}) = \text{sng} \\ (\uparrow \text{SUBJ PERS}) = \text{3rd} \end{array}$$

This lexical entry says that *love* is a verb that means 'love', and takes two arguments, an obligatory subject and an obligatory object (as contained within the < > brackets). The parentheses here do *not* mean "optional" – as both arguments are, indeed, obligatory. It also tells us that *loves* is the present tense form. (The lexical entry for *loved* would have a different value for

⁴ This is a gross over-simplification, I'm ignoring a large number of mapping operations here. I'm also ignoring the complexities of inflectional morphology and how they interact with the featural structure. See Bresnan (2001) or Falk (2001) for extensive discussion.

form. (The lexical entry for *loved* would have a different value for TENSE.) Finally it tells us that the subject is third person singular. ((↑SUBJ NUM) means “my subject’s number is ...”).

All lexical items bring some information to the sentence. For example, we know that the determiner *the* is definite and such information is contained in its lexical entry.

5) *the*: D (↑DEF) = +

Functional and featural information comes with the lexical item when it is inserted into the c-structure (as we will see below).

4. F-STRUCTURE

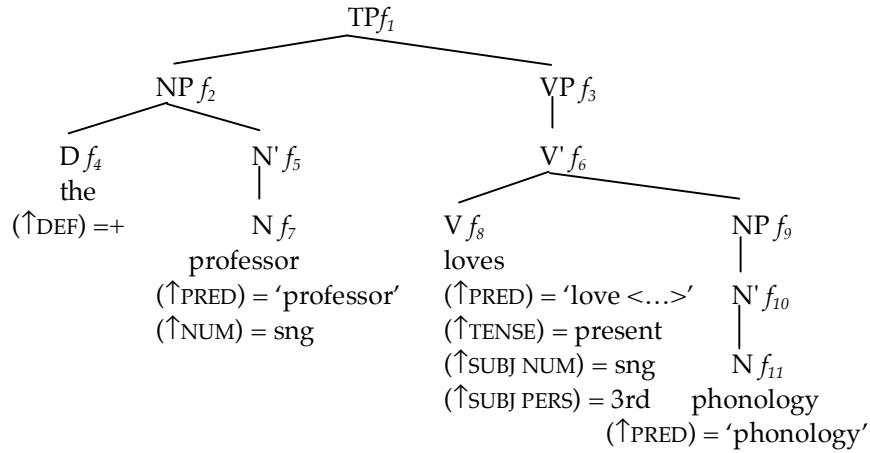
F-structures, as noted above, are the set of all the attribute value pairs for a sentence. Perhaps the easiest way to see this is to look at an example. We use again the sentence *the professor loves phonology*. An f-structure for this sentence is given in (6):

6)
$$\left[\begin{array}{ll} \text{PRED} & \text{'love <SUBJ, OBJ>'} \\ \text{TENSE} & \text{present} \\ \text{SUBJ} & \left[\begin{array}{ll} \text{DEF} & + \\ \text{NUM} & \text{sng} \\ \text{PRED} & \text{'professor'} \end{array} \right] \\ \text{OBJ} & \left[\text{PRED} \quad \text{'phonology'} \right] \end{array} \right]$$

The topmost PRED function tells you what the predicate of the sentence is. It also contains information about the a-structure of the sentence. The TENSE feature tells you about the tense of the sentence. The SUBJ and OBJ functions have submatrices (containing information on their internal structure) as values.

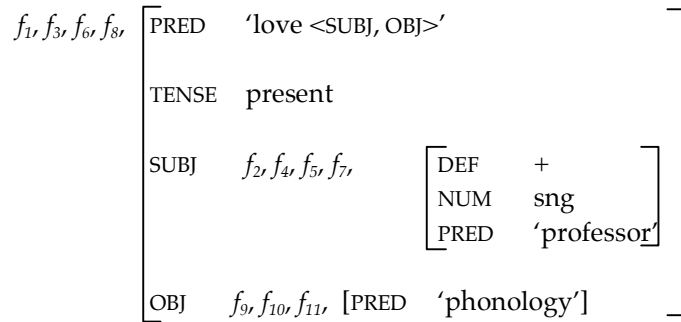
C-structures must be related to f-structures somehow. This is accomplished with the use of variables. Consider the simple c-structure in (7). Each lexical item is followed by the information it contributes by virtue of its lexical entry. Each node in the tree is marked with a *variable* (f_1, f_2, f_3, \dots , etc.). These will be used in mapping to the f-structure. Again, ignore the arrows for the moment.

7)



Each of these variables corresponds to a pair of matrix brackets in the f-structure. There is *no* one-to-one correspondence here. Multiple nodes in the tree can correspond to the same (sub)AVM:

8)



This means that the information contained in nodes f_2, f_4, f_5, f_7 contribute the SUBJ features to the sentence. f_9, f_{10}, f_{11} , provide the OBJ info, etc.

The mapping formally happens through what is called an *f-description*. The f-description is set out with *functional equations*. These equations tell us, for example, that the subject of the sentence f_1 corresponds to the constituent f_2 . This is written as:

$$9) (f_1 \text{ SUBJ}) = f_2$$

The fact that the subject NP (f_2) is definite is encoded in D node (f_4):

$$10) f_2 = f_4$$

When a node is a head, or simply passes information up the tree (e.g., V' or V), then a simple equivalence relation is stated:

11) $f_3 = f_6$

These functional equations control how information is mapped through the tree and between the tree and the f-structure. Each piece of information in the f-structure comes from (various places in) the tree as controlled by the functional equations in the f-description.

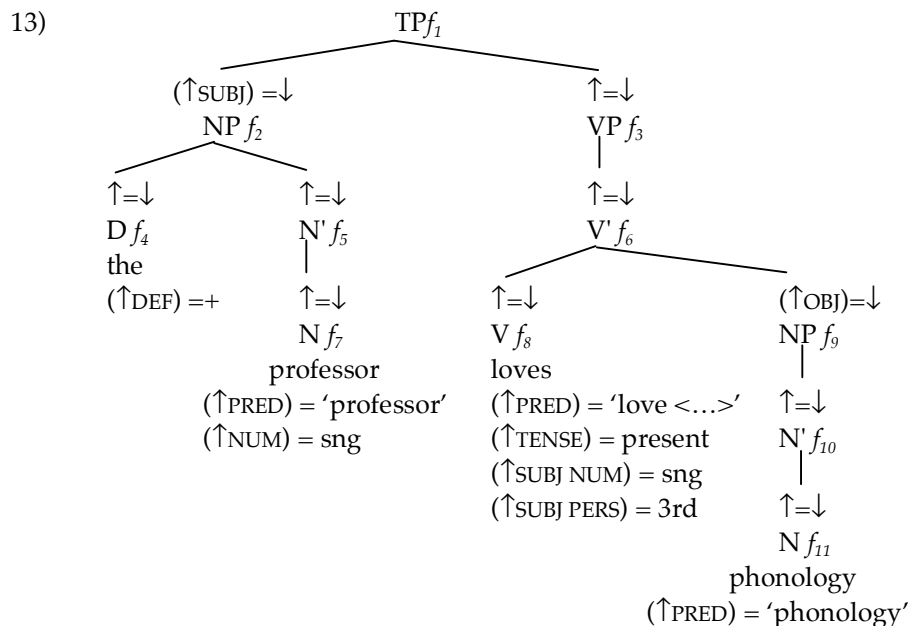
In order to make this clearer, as a notational device, c-structures are often annotated with their functional equations. There is a useful device that is used to clarify these annotations. These are *metavariables*. Metavariables are variables that stand for other variables. There are two metavariables:

- 12) a) \downarrow means “this node”
 b) \uparrow means “my mother” (immediately dominating node)

So the equation $\uparrow = \downarrow$ means “all of the features I have also belong to my mother” – in other words, a head. The notation $(\uparrow_{\text{SUBJ}}) = \downarrow$ means “I represent the subject function of my mother”.

We also saw these arrows in lexical entries. They mean the same thing here. $(\uparrow_{\text{PRED}}) = \text{'love'}$ means “the terminal node that I fill has the predicate value of ‘love’.”

Here is the c-structure in (7) repeated below as (13) with the functional equations annotated using metavariables:



Unification is the idea that features and functions coming from disparate places in the tree must be compatible with one another. Take, for

example, the fact that the verb specifies that its subject must be third person singular (as marked by the suffix -s). If the subject has number or person features, then those features must match the verb's subject features in the f-structure. This is forced by the fact that the f-structure (in mathematical terms) solves for, or is the solution to, the set of functional equations known as the functional description. If the features didn't match, then the f-structure wouldn't unify. Notice that this is a very similar notion to the idea of feature checking, discussed in chapter 12. Feature checking also ensures compatibility of features coming from different parts of the sentence. Minimalism simply uses movement, rather than functional equations, to guarantee this. Both systems have their advantages; LFG's system has a certain mathematical precision and elegance that P&P movement and feature checking do not. By contrast, P&P/Minimalism is able to derive word order differences between languages from feature checking and movement. Minimalism thus provides a slightly more explanatory theory of word order than LFG, which uses language-specific phrase structure rules. It is not at all clear which approach is preferable.

There are a number of constraints on f-structures. Three of these conspire together to result in the LFG equivalent of the theta criterion:

- 14) a) **Uniqueness:** In a given f-structure, a particular attribute may have at most one value.
- b) **Completeness:** An f-structure must contain all the governable grammatical functions that its predicate governs.
- c) **Coherence:** All the governable grammatical functions in an f-structure must be governed by a local predicate.

(14a) is also the constraint that forces unification. All f-structures must meet these constraints.

4.1 Why F-structures?

We now have a fairly detailed sketch of the basics of LFG. Before turning to some implementations of the model dealing with some of the empirical issues we've looked at elsewhere in this book, it is worth considering why the LFG model uses f-structures. The answer is fairly straightforward: Information about a particular grammatical function may come from more than one place in the tree and, more importantly, the sources of information do not have to be constituents. Falk (2001) gives the example of the pair of following sentences:

- 15) a) The deer are dancing.
- b) The deer is dancing.

The form of the subject noun is identical in both of these sentences. However, in terms of meaning, it is plural in (15a) and singular in (15b). Let's assume that the lexical entry for the form 'deer' lacks any specification for number:

16) *deer* N (\uparrow PRED) = 'deer'

The number associated with the subject function in the sentences in (15) comes from the auxiliary:

17) a) *are* T (\uparrow TENSE) = present
 (\uparrow SUBJ NUM) = pl
 b) *is* T (\uparrow TENSE) = present
 (\uparrow SUBJ NUM) = sg

While the number comes from the auxiliary, it is only really a property of the subject. The NUM feature still gets mapped to the SUBJ function, because of the functional annotation (\uparrow SUBJ NUM) = pl. Similar facts are seen in "free word order" or "non-configurational" languages, like the aboriginal Warlpiri language spoken in Australia. Contra the principle of modification we discussed in chapter 3, in Warlpiri, words that modify one another do not have to appear as constituents on the surface. Since information can come from various parts of the tree, this points towards a system where functional information is not read directly off the tree; instead, an f-structure-like level is motivated.⁵

You can now try General Problem Set 1

5. ASSORTED PHENOMENA

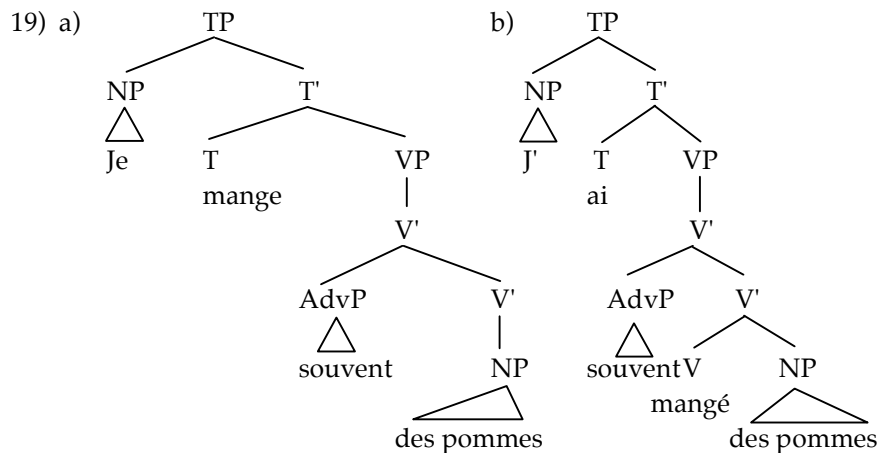
Having now quickly laid out the fundamentals of LFG (c-structure, f-structure, the lexicon, etc.), we turn to how various phenomena discussed in other parts of this book are treated in LFG.

⁵ It is worth briefly mentioning how P&P deals with these same facts. At D-structure, these units do form constituents. A transformation, known as *scrambling* (see the problem sets in chapter 11), then reorders the elements so they don't surface as constituents. Both approaches achieve essentially the same results with different underlying assumptions.

5.1 Head Mobility

In chapter 9, we analyzed alternations between the position of a tensed verb and the position of auxiliaries in languages like French, Vata, or Irish as involving head-to-head movement. In French, the main verb alternates in its position relative to adverbs, depending upon the presence or absence of an auxiliary. When an auxiliary is present, the verb stays in its base position. When there is no auxiliary, the verb moves to T. Although LFG has no movement, it has a related account of these phenomena. LFG simply posits that tensed verbs and untensed participial forms belong to different categories. This is called *head mobility*. Tensed verbs are of category T, whereas untensed forms are of category V:

- 18) a) *mange* T (\uparrow PRED) = 'eat <(\uparrow SUBJ), (\uparrow OBJ)>'
 (\uparrow TENSE) = present
 b) *mangé* V (\uparrow PRED) = 'eat <(\uparrow SUBJ), (\uparrow OBJ)>'
 c) *ai* T (\uparrow TENSE) = present⁶



You'll notice that in (19a) the VP has no head V. This is allowed in LFG, but not in P&P theory (as it violates the endocentric properties of X-bar theory).

In English, both participles and tensed verbs are of the category V; only auxiliaries are of category T:

- 20) a) *eat* V (\uparrow PRED) = 'eat <(\uparrow SUBJ), (\uparrow OBJ)>'
 (\uparrow TENSE) = present
 b) *eaten* V (\uparrow PRED) = 'eat <(\uparrow SUBJ), (\uparrow OBJ)>'
 c) *have* T (\uparrow TENSE) = present


⁶ The sentence *J'ai souvent mangé des pommes* also bears what are called aspect features. We leave these aside here.

5.2 Passives

21) a) *kiss* V (\uparrow PRED) = ‘kiss <(\uparrow SUBJ), (\uparrow OBJ)>
 +en \Downarrow \Downarrow
 b) *kissed*_{pass} V (\uparrow PRED) = ‘kiss < \emptyset (\uparrow SUBJ)>

5.3 Raising and Control

22)

```
graph TD
    TP --> NP[NP]
    TP --> T_prime[T']
    NP --> Jean["  
Jean"]
    T_prime --> T[T]
    T --> is[is]
    T_prime --> AP[AP]
    AP --> A_prime[A']
    A_prime --> A[A]
    A --> likely["likely/reluctant"]
    A_prime --> VP_prime[VP']
    VP_prime --> to[to]
    VP_prime --> VP[VP]
    VP --> V_prime[V']
    V_prime --> V[V]
    V --> dance[dance]
```

The differences between control and raising constructions are expressed in lexical entries and f-structures, not in the phrase structure (c-structure) as in P&P.

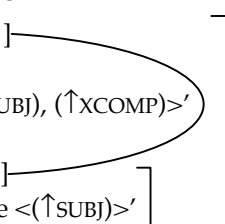
Let's start with control constructions.⁷ LFG uses a special grammatical function to represent non-finite complements. Since embedded predicates in these constructions don't have their subject expressed overtly, they are considered *open functions* (that is, they are missing an argument). The grammatical function used to represent open functions is XCOMP (or XADJU in the case of adjuncts). Predicates like *is likely* and *is reluctant* select for XCOMPS in their lexical entries. The lexical entry for *reluctant* is given in (17):

- 23) *reluctant* A (\uparrow PRED) = 'reluctant <(\uparrow SUBJ), (\uparrow XCOMP)>'

Reluctant takes a thematic subject and an open function as its arguments. When we combine this with a predicate such as *dance*, we get an f-structure like the following for the sentence *Jean is reluctant to dance*:

- 24) * $\left[\begin{array}{ll} \text{SUBJ} & [\text{PRED 'Jean'}] \\ \text{PRED} & \text{'reluctant <(\uparrow SUBJ), (\uparrow XCOMP)>} \\ \text{TENSE} & \text{present} \\ \text{XCOMP} & \left[\begin{array}{ll} \text{SUBJ} & \text{?????} \\ \text{PRED} & \text{'dance <(\uparrow SUBJ)>} \end{array} \right] \end{array} \right]$

This f-structure is ill-formed, because it violates the principle of completeness: The subject function of the XCOMP is not filled. This is resolved using the LFG equivalent of control: *functional control*. Functional control is indicated with a curved line linking the two functions.

- 25) $\left[\begin{array}{ll} \text{SUBJ} & [\text{PRED 'Jean'}] \\ \text{PRED} & \text{'reluctant <(\uparrow SUBJ), (\uparrow XCOMP)>} \\ \text{TENSE} & \text{present} \\ \text{XCOMP} & \left[\begin{array}{ll} \text{SUBJ} & [] \\ \text{PRED} & \text{'dance <(\uparrow SUBJ)>} \end{array} \right] \end{array} \right]$
- 

This indicates that the subject of the main clause is also the subject of the non-finite complement. Functional control is licensed by the lexical entry of

⁷ I am simplifying the situation here. The range of phenomena P&P theory groups as "control" constructions divides into two groups in the LFG analysis: functional control and anaphoric control, which have different properties. We won't distinguish these here. See Falk (2001) or Bresnan (2001) for explanations of the difference.

26) *reluctant* A (\uparrow PRED) = ‘*reluctant* <(\uparrow SUBJ), (\uparrow XCOMP)>’
 (\uparrow SUBJ) = (\uparrow XCOMP SUBJ)

In control constructions, the controller must *f-command* the controllee. One node f-commands another if it is less deeply embedded in the f-structure (inside fewer square brackets).

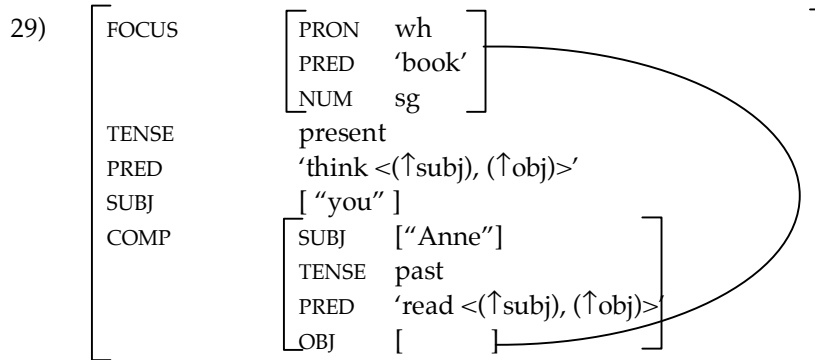
27) *likely* A (\uparrow_{PRED}) = ‘likely <(\uparrow_{XCOMP})> (\uparrow_{SUBJ})’
 (\uparrow_{SUBJ}) = ($\uparrow_{\text{XCOMP SUBJ}}$)

28) SUBJ [PRED 'Jean']
 PRED 'likely <(↑XCOMP)>, (↑SUBJ)'
 TENSE present
 XCOMP [SUBJ []
 PRED 'dance <(↑SUBJ)>']

The difference between the raising and the control sentence simply reduces to a matter of argument structure.

5.4 *Wh-movement: Long Distance Dependencies*

The sharing of feature structures (as expressed by the curved line) is also used to define long distance dependencies such the relationship between a *wh*-phrase and the gap (or trace) it is associated with. There is a special grammatical function: FOCUS, which is found in *wh*-constructions. I'll abstract away from the details, but in English, this function is associated with the specifier of CP. The element taking the FOCUS function must share features with some argument (this is forced on the sentence by the constraint of coherence). The following is the f-structure for the sentence *Which novel do you think Anne read?* (COMP is the function assigned tensed embedded clauses.)



The FOCUS shares the features of the OBJ function of the complement clause, indicating that they are identical.

There is much more to *wh*-dependencies in LFG than this simple picture. For example, LFG has an account of island conditions based on f-structures. There is also an explicit theory of licensing *wh*-dependencies. For more on this see the readings in the further reading section at the end of the chapter.

You can now try Challenge Problem Sets 1–3

6. CONCLUSION

This concludes our whirlwind tour of Lexical-Functional Grammar. A single chapter does not give me nearly enough space to cover the range and depth of this theory, and as such doesn't give LFG the opportunity to really shine. I've glossed over many details and arguments here, but I hope this chapter gives you enough information to pursue further reading in this alternative framework. Remember that the results derived in frameworks like LFG can give us insights about the way human language works. If we're clever

enough, we should be able to incorporate these insights into the P&P or Minimalist framework, too. Similarly, a student of LFG should be able to use the insights of Minimalism or P&P to inform their theorizing. Remember that syntactic theories are only hypotheses about the way syntax is organized. We really don't know which approach, if any of these, is right. So by looking at other theoretical approaches we question our basic assumptions, and consider new alternatives.

IDEAS, RULES, AND CONSTRAINTS INTRODUCED IN THIS CHAPTER

- i) **C-structure:** Constituent structure. The tree in LFG. Roughly equivalent to S-structure in P&P.
- ii) **Grammatical Function:** Same thing as a grammatical relation. Common grammatical functions: SUBJ = subject; OBJ = object; PRED = predicate; XCOMP = open complement (non-finite clause); COMP = closed complement (finite embedded clause); FOCUS = the function associated with *wh*-phrases
- iii) **F-structure:** The level of representation where grammatical functions are unified.
- iv) **Attribute Value Matrix (AVM):** A matrix that has an attribute (or function) on the left and its value on the right. The set of all AVMs for a sentence form the sentence's f-structure.
- v) **A-structure:** Argument structure. The LFG equivalent of the theta grid.
- vi) **Variables:** LFG uses variables (f_1, f_2, f_3, \dots , etc.) for each node on the c-structure which are used in the mapping between c-structure and f-structure.
- vii) **Functional Equation:** An equation that maps one variable to another (e.g., $(f_1 \text{ SUBJ}) = f_2$ says that f_2 maps to f_1 's SUBJ function).
- viii) **F-description:** The set of all functional equations. Defines the mapping between c-structure and f-structure.
- ix) **Annotated C-structure:** A c-structure annotated with the functional equations which map it to the f-structure.
- x) **Metavariable:** A variable over variables. \uparrow = my mother's variable, \downarrow = my variable.

- xi) $\hat{f}=\downarrow$: “All the functional information I contain, my mother also contains.”
- xii) $(\hat{f}_{SUB})=\downarrow, (\hat{f}_{OBJ})=\downarrow$: “I am the subject of the node that dominates me” or “I am the object of the node that dominates me.”
- xiii) **Unification**: All the features and functions associated with the f-structure must be compatible. (Similar to feature checking in P&P.)
- xiv) **Uniqueness**: In a given f-structure, a particular attribute may have at most one value.
- xv) **Completeness**: An f-structure must contain all the governable grammatical functions that its predicate governs.
- xvi) **Coherence**: All the governable grammatical functions in an f-structure must be governed by a local predicate.
- xvii) **Head Mobility**: The idea that lexical items can take different categories depending upon their features. E.g., a tensed verb in French is of category T, whereas an untensed one is a V. This derives head-to-head movement effects.
- xviii) **Lexical Rule of Passives**: Passives in LFG are entirely lexical. There is no syntactic movement:
- xix) **Open function** (XCOMP): A function with a missing argument (e.g., a non-finite clause).
- xx) **Functional Control**: The LFG equivalent of control, indicated with a curved line linking two AVMs in an f-structure.
- xxi) **Raising vs. Control**: In LFG raising vs. control reduces to a lexical difference. The SUBJ function in raising constructions isn’t thematic, but is in control constructions.

FURTHER READING

- Bresnan, Joan (2001) *Lexical-Functional Syntax*. Oxford: Blackwell.
- Dalrymple, Mary, Ronald Kaplan, John Maxwell and Annie Zaenen (eds.) (1995) *Formal Issues in Lexical-Functional Grammar*. Stanford: CSLI Publications.
- Falk, Yehuda N. (2001) *Lexical-Functional Grammar: An Introduction to Parallel Constraint-Based Syntax*. Stanford: CSLI Publications.

Kaplan, Ronald (1995) The formal architecture of Lexical-Functional Grammar. In Mary Dalrymple, et al. (eds.), *Formal Issues in Lexical-Functional Grammar*. Stanford: CSLI Publications. pp. 7–27.

Kaplan, Ronald and Joan Bresnan (1982) Lexical-Functional Grammar: A formal system for grammatical representation. In Joan Bresnan (ed.), *The Mental Representation of Grammatical Relations*. Cambridge: MIT Press. pp. 173–281.

A lot of material on LFG can be found on the web:

<http://www-lfg.stanford.edu/lfg>

<http://www.essex.ac.uk/linguistics/LFG/>

Further reading on other theories:

Edmondson, Jerold and Donald A. Burquest (1998) *A Survey of Linguistic Theories* (3rd ed.). Dallas: Summer Institute of Linguistics.

Sells, Peter (1985) *Lectures on Contemporary Syntactic Theories*. Stanford: CSLI Publications.

GENERAL PROBLEM SETS

1. ENGLISH

[Application of Skills; Intermediate]

Draw the annotated c-structures and f-structures for the following sentences. (It may also be helpful to write out lexical entries for each of the words detailing the information each word contributes):

- a) Susie loves the rain.
- b) Joan thinks that Norvin is likely to write a paper.
- c) What have you read?

CHALLENGE PROBLEM SETS

CHALLENGE PROBLEM SET 1: ICELANDIC (AGAIN)

[Critical Thinking; Challenge]

Go back to the questions on quirky case in Icelandic in previous chapters and review the data. These data caused problems for us with our case driven theory of movement and our theory of PRO. Do these same problems arise in LFG? Why or why not?

CHALLENGE PROBLEM SET 2: TRANSFORMATIONS OR NOT?

[Critical Thinking; Challenge]

Construct the design for an experiment that would distinguish between a transformational approach, and a non-transformational approach like LFG.

CHALLENGE PROBLEM SET 3: WANNA-CONTRACTION*[Critical and Creative Thinking; Challenge]*

How might LFG account for *wanna*-contraction (see chapter 11) if it doesn't have movement or traces?