SYNTAX AND SEMANTICS

Lexical Functional Grammar

34

Mary Dalrymple

ACADEMIC PRESS

A list of titles in this series appears at the end of this book.

PREFACE AND ACKNOWLEDGMENTS

This book is a tribute to the extraordinary accomplishments of Joan Bresnan and Ron Kaplan, my teachers, mentors, and friends. What is presented here is the theory they created together; it is lucky for all of us that they happened to end up in Pisa together back in 1977!

My first exposure to LFG was in a class taught by K.P. Mohanan at the University of Texas in 1981. Mohanan's teaching skills are legendary, and I'm grateful for having had such a good introduction to the theory.

My debt to colleagues and friends in writing this book is enormous. Tracy Holloway King assisted in every aspect of preparation of this book, from reading early and virtually unreadable drafts to providing sage advice and counsel on all aspects of the linguistic analyses presented here. I am also very grateful to the many linguists who provided helpful comments and criticism of early and late drafts of the book: Farrell Ackerman, David Ahn, Ash Asudeh, Martin van den Berg, Sascha Brawer, Joan Bresnan, Miriam Butt, Cleo Condoravdi, Dick Crouch, Cris Culy, Yehuda Falk, Brent Fitzgerald, Ken Forbus, Anette Frank, John Fry, Ron Kaplan, Shin-Sook Kim, Jonas Kuhn, John Lamping, Hiroshi Masuichi, Umarani Pappuswamy, Jonathan Reichenthal, Louisa Sadler, Ida Toivonen, Vijay Saraswat, and Annie Zaenen. Particular thanks go to colleagues who gave especially detailed and helpful comments, often on very short notice: worthy of special mention are Farrell Ackerman, Ash Asudeh, Martin van den Berg, Cleo Condoravdi, Chris Culy, Brent Fitzgerald, Yehuda Falk, Anette Frank, Ron Kaplan, Tracy Holloway King, Louisa Sadler, and Annie Zaenen. My sister Matty Dalrymple provided expert editing assistance, for which I am always grateful, and Jeanette Figueroa provided invaluable technical support.

I have also benefited from expert comments on particular chapters of the book; the range of topics covered in this book far exceeds anything I could have attempted unaided. Ron Kaplan provided assistance with Chapter 2 (Functional Structure), Chapter 5 (Describing Syntactic Structures), and Chapter 6 (Syntactic Relations and Syntactic Constraints); Tracy Holloway King assisted with Chapter 3 (Constituent Structure); Farrell Ackerman and Miriam Butt assisted with Chapter 8 (Argument Structure and Mapping Theory); Ash Asudeh, Martin van den Berg, Dick Crouch, and Tracy Holloway King assisted with Chapter 9 (Meaning and Semantic Composition); Cleo Condoravdi assisted with Chapter 10 (Modification); Martin van den Berg, Dick Crouch, John Lamping, Louisa Sadler, and Annie Zaenen assisted with Chapter 11 (Anaphora); Ash Asudeh, Cleo Condoravdi, Dick Crouch, and Tracy Holloway King assisted with Chapter 12 (Functional and Anaphoric Control); Chris Culy assisted with Chapter 13 (Coordination); and Ash Asudeh, Martin van den Berg, Cleo Condoravdi, Dick Crouch, Stanley Peters, Tracy Holloway King, and Annie Zaenen assisted with Chapter 14 (Long-Distance Dependencies). Besides help with particular chapters, I owe an enormous intellectual debt to colleagues whose clear thinking and unerring formal intuitions are evident on each page of this book: Ron Kaplan, John Lamping, John Maxwell, Fernando Pereira, and Vijay Saraswat.

Ken and David Kahn also deserve thanks for putting up with me as this book took shape, and for enriching my life beyond measure.

Two other books on LFG have recently appeared: Joan Bresnan's *Lexical-Functional Syntax* and Yehuda Falk's *Lexical-Functional Grammar: An Introduction to Parallel Constraint-Based Syntax.* These valuable resources are intended for use as textbooks and contain exercises and guidance for using the books as teaching material; Falk's book also contains a useful glossary of terms. This book contrasts with Bresnan's and Falk's in several ways: it is not intended primarily as a textbook but rather as a handbook and theoretical overview, and it includes semantic as well as syntactic analyses of the linguistic phenomena that are discussed. Each book fills a different need in the community; it is a happy confluence of factors that produced all of these LFG resources within a relatively brief period.

Although much has had to be omitted in this work, my hope is that what has been collected here will be useful and that it will form a basis for future researchers to fill in the many gaps that remain.

LIST OF ABBREVIATIONS

1	first person
2	second person
3	third person
ABL	ablative case
ABS	absolutive case
ACC	accusative case
AUX	auxiliary verb
DAT	dative case
ERG	ergative case
FEM	feminine gender
FV	final vowel
GEN	genitive case
INF	infinitival
LOC	locative case
MASC	masculine gender
NEUT	neuter gender
NOM	nominative case
PART	partitive case
PL	plural
PRES	present tense
SG	singular

1

BACKGROUND AND THEORETICAL ASSUMPTIONS

Lexical Functional Grammar (LFG) is a nontransformational theory of linguistic structure which assumes that language is best described and modeled by parallel structures representing different facets of linguistic organization and information, related to one another by means of functional constraints.

The theory had its beginnings in the 1970s, at a time of some upheaval in the theory of generative grammar. Early transformational grammar proposed the existence of "kernel sentences" (Chomsky 1957), basic simple declarative clauses generated by a simple phrase structure grammar. More complex sentences were derived by various specific transformations: for example, passive sentences were derived from their active counterparts by means of a passive transformation, described in terms of properties of the phrase structures of the input and output sentences. The influence of the transformational view persists to the present day in the process-oriented terminology commonly used for various grammatical phenomena: passivization, dative shift, and so on.

In time, however, linguists began to be bothered by the lack of generality of the early transformational approach. It was not easy to see how the very specific transformations that had been proposed could capture crosslinguistic generalizations.

In particular, as discussed by Perlmutter and Postal (1983b), there seemed to be no way to give a uniform statement of transformational rules across languages with different phrase structural descriptions for obviously similar transformations such as Passive. Linguists began to see that the generalizations underlying many transformational rules depend not on phrase structure configuration, but on traditional abstract syntactic concepts such as subject, object, and complement. If rules could be stated in terms of these abstract concepts, a crosslinguistically uniform statement of generalizations about such rules would emerge.

At the same time, linguists noted that a large class of transformations were "structure preserving" (Emonds 1976, page 3):

> A transformational operation is structure-preserving if it moves, copies, or inserts a node C into some position where C can be otherwise generated by the grammar.

The existing transformational framework would not have led to the prediction that transformations would operate in this way. Since transformations were not constrained as to the output structure they produced, it was surprising that they would produce structures like those that the basic grammar could otherwise generate. This important finding had wide-reaching implications: the basic phrase structure of languages is invariant, and the application of particular transformations does not alter this basic phrase structure.

Why should so many transformations have been structure-preserving in this sense? Bresnan (1978) made the key observation: all structure-preserving transformations can be reformulated as lexical redundancy rules. According to this view, operations on the abstract syntactic argument structure of a lexical item produce a new syntactic argument structure, with a surface form that is realized in an expected way by a basic phrase structure grammar. This allowed an abstract and uniform crosslinguistic characterization of argument alternations like the active-passive relation, while also allowing for a theory of crosslinguistic similarities and differences in the phrasal expression of the different alternations.

With this, the need emerged for a theory allowing simultaneous expression of both the phrasal constituency of a sentence and its more abstract functional syntactic organization. The formal insights leading to the development of Lexical Functional Grammar arose originally from the work of Woods (1970), who explored methods for representing the surface constituent structure of a sentence together with more abstract syntactic information. Building on this work, Kaplan (1975a,b, 1976) realized that placing certain constraints on the representation of abstract syntactic structure and its relation to surface phrasal structure would lead to a simple, formally coherent and linguistically well-motivated grammatical architecture. Based on these formal underpinnings, the relation of the abstract functional syntactic structure of a sentence to its phrase structure could be fully explored. More information about the historical development of the theory can be found in Dalrymple et al. (1995a).

The name of the theory, "Lexical Functional Grammar," encodes two important dimensions along which LFG differs from other theories. First, the theory is *lexical* and not transformational: it states relations among different verbal diatheses in the lexicon rather than by means of syntactic transformations. In 1978, when the theory was first proposed, this was a fairly radical idea, but in the intervening years it has come to be much more widely accepted; it is a fundamental assumption of Categorial Grammar (Moortgat 1988; Morrill 1994; Steedman 1996) as well as of Head-Driven Phrase Structure Grammar (Pollard and Sag 1994), Construction Grammar (Kay 1998), and some transformationally oriented works (Grimshaw 1990).

Unlike some other theories of syntax, then, the lexicon is not merely a repository for exceptions, a place in which syntactically or semantically exceptional information is recorded. Since LFG is a lexical theory, regularities across classes of lexical items are part of the organization of a richly structured lexicon, and an articulated theory of complex lexical structure is assumed. Work on lexical issues has been an important focus of LFG from the beginning, and this research continues with work to be described in the following pages.

The second dimension that distinguishes Lexical Functional Grammar is that it is *functional* and not configurational: abstract grammatical functions like subject and object are not defined in terms of phrase structure configurations or of semantic or argument structure relations, but are primitives of the theory. LFG shares this view with Relational Grammar (Perlmutter and Postal 1983b) and Arc Pair Grammar (Johnson and Postal 1980), as well as with Construction Grammar (Kay 1998).

LFG assumes that functional syntactic concepts like subject and object are relevant for the analysis of every language: that the same notions of abstract grammatical functions are at play in the structure of all languages, no matter how dissimilar they seem on the surface. Of course, this does not imply that there are no syntactic differences among languages, or among sentences in different languages that have similar meanings; indeed, the study of abstract syntactic structure in different languages is and has always been a major focus of the theory. Just as the phrase structure of different languages obeys the same general principles (for example, in adherence to *X-bar theory*; see Chapter 3, Section 4.1), in the same way the abstract syntactic structure of languages obeys universal principles of functional organization and draws from a universally available set of possibilities, but may vary from language to language. In this sense, the functional structure of language is said to be "universal."

In recent LFG work, grammatical functions have been closely analyzed, and similarities have been found among them; natural classes of grammatical functions are found to behave alike, particularly in the theory of linking between semantic arguments and syntactic functions. To analyze these similarities, grammatical functions like subject and object are decomposed into more basic features such as ±RESTRICTED, as described in Chapter 8, Section 4.1. On this view, grammatical functions are no longer thought of as atomic. Even given these decompositions, however, the grammatical functions of LFG remain theoretical primitives, in that they are not derived or defined in terms of other linguistic notions such as agenthood or phrasal configuration.

This book concentrates primarily on the theory of LFG as it has developed since its inception in the late 1970s. Most of the book should be accessible to upper-level undergraduate or graduate students who have some background in syntax, though the semantic sections of the book will be easier to read for those who also have some background in logic and formal semantics. The book consists of five parts. In the first part, comprising Chapter 2 (Functional Structure), Chapter 3 (Constituent Structure), and Chapter 4 (Syntactic Correspondences), we will examine the two syntactic structures of LFG, the constituent structure and the functional structure. We will discuss the nature of the linguistic information they represent, the formal structures used to represent them, and the relation between the two structures.

The second part, comprising Chapter 5 (Describing Syntactic Structures) and Chapter 6 (Syntactic Relations and Syntactic Constraints), outlines the formal architecture of LFG and explains how to describe and constrain the constituent structure, the functional structure, and the relation between them. A clear understanding of the concepts described in Chapter 5 is essential for the discussion in the rest of the book. Chapter 6 is best thought of as a compendium of relatively more advanced formal tools and relations, and may be most profitably used as a reference in understanding the analyses presented in the rest of the book.

The third part of the book, comprising Chapter 7 (Beyond Syntax: Nonsyntactic Structures), Chapter 8 (Argument Structure and Mapping Theory), and Chapter 9 (Meaning and Semantic Composition), explores the relation of nonsyntactic structures to the functional structure and constituent structure. Chapter 7 introduces the *projection architecture*, a theory of the relations between different aspects of linguistic structure. Chapter 8 discusses the content and representation of *argument structure*, its relation to syntax, and its role in determining the syntactic functions of the arguments of a predicate. Chapter 9 introduces the LFG view of the syntax-semantics interface and semantic representation, according to which the meaning of an utterance is determined via logical deduction from a set of premises associated with the syntactic subparts of the utterance. We will use this theory in the analyses presented in the following chapters.

The fourth part of the book illustrates the concepts of the theory more explicitly by presenting a series of sketches of the syntax and semantics of a range of representative linguistic phenomena. The syntactic aspects of the analyses are presented separately from the semantic aspects, so readers who are not interested in formal semantic analysis should still be able to profit from the syntactic discussion in these chapters. Chapter 10 (Modification) discusses the syntax and semantics of modifiers, particularly concentrating on modification of nouns by adjectives. Chapter 11 (Anaphora) presents a theory of the syntax and semantics of anaphoric binding, including both intrasentential and intersentential anaphora. Chapter 12 (Functional and Anaphoric Control) discusses constructions involving *control*, where the referent of an argument (often the subject) of a subordinate clause is constrained by lexical or constructional factors. Chapter 13 (Coordination) presents an analysis of aspects of the syntax and semantics of coordination, and Chapter 14 (Long-Distance Dependencies) discusses long-distance dependencies in topicalization, relative clause formation, and question formation.

The fifth part of the book, Chapter 15 (Related Research Threads and New Directions), discusses new developments in the theory of LFG, including computational and algorithmic research in parsing and generation, LFG-based theories of language acquisition, and Optimality Theory-based work.

The book concludes with an appendix containing the rules of *linear logic*, to be introduced in Chapter 9, and three indexes: an index of cited authors, a language index, and a subject index. The language index contains information about the linguistic family to which the language belongs as well as a rough characterization of where the language is spoken.

2

FUNCTIONAL STRUCTURE

LFG assumes two different ways of representing syntactic structure, the *constituent structure* or *c-structure* and the *functional structure* or *f-structure*. These two structures constitute two subsystems of the overall system of linguistic structures. Functional structure is the abstract functional syntactic organization of the sentence, familiar from traditional grammatical descriptions, representing syntactic predicate-argument structure and functional relations like subject and object. Constituent structure is the overt, more concrete level of linear and hierarchical organization of words into phrases.

Section 1 of this chapter presents motivation for the categories and information appearing in functional structure and outlines some common characteristics of functional structure categories. Section 2 shows that *syntactic subcategorization requirements*, a characterization of the array of syntactic arguments required by a predicate, are best stated in functional terms. The formal representation of functional structure and constraints on f-structure representations are discussed in Section 3. Finally, Section 4 contrasts the LFG view with other theoretical approaches to the definition and treatment of functional structure.

1. FUNCTIONAL INFORMATION AND FUNCTIONAL STRUCTURE

Abstract grammatical relations have been studied for thousands of years. Apollonius Dyscolus, a grammarian in Alexandria in the second century A.D., gave a syntactic description of Greek that characterized the relations of nouns to verbs and other words in the sentence, providing an early characterization of transitivity and "foreshadow[ing] the distinction of subject and object" (Robins 1967). The role of the subject and object and the relation of syntactic predication were fully developed in the Middle Ages by the modistae, or speculative grammarians (Robins 1967; Covington 1984).

More recent work also depends on assuming an underlying abstract regularity operating crosslinguistically. Modern work on grammatical relations and syntactic dependencies was pioneered by Tesnière (1959) and continues in the work of Hudson (1984), Mel'čuk (1988), and others working within the dependency-based tradition. Typological studies are also frequently driven by reference to grammatical relations: for instance, Greenberg (1966) states his word order universals by reference to subject and object. Thus, LFG aligns itself with approaches in traditional, nontransformational grammatical work, in which these abstract relations were assumed.

1.1. Distinctions among Grammatical Functions

It is abundantly clear that there are differences in the behavior of phrases depending on their grammatical function. For example, in languages exhibiting "superiority" effects, there is an asymmetry between subjects and nonsubjects in *multiple wh-questions*, questions with more than one wh-phrase. It is not possible for the object phrase in a wh-question to appear in initial position in the sentence if the subject is also a wh-phrase like *what* or *who* (Chomsky 1981, Chapter 4):

(1) a. *Who saw what?*

b. **What did who see?*

Not all languages exhibit these effects: for example, King (1995, page 56) shows that superiority effects do not hold in Russian. Nevertheless, many languages do exhibit an asymmetry between subjects and nonsubjects in constructions like (1).

In fact, however, the subject-nonsubject distinction is only one aspect of a rich set of distinctions among grammatical functions. Keenan and Comrie (1977) propose a more fine-grained analysis of abstract grammatical structure, the *Keenan-Comrie hierarchy* for relative clause formation. The Keenan-Comrie hierarchy gives a ranking on grammatical functions that constrains relative clause formation by restricting the grammatical function of the argument in the relative clause that is interpreted as coreferent with the modified noun. The border between any two adjacent grammatical functions in the hierarchy can represent a distinction between acceptable and unacceptable relative clauses in a language, and different languages can set the border at different places on the hierarchy:[1]

(2) Keenan-Comrie Hierarchy:

SUBJ > DO > IO > OBL > GEN > OCOMP

Keenan and Comrie state that "the positions on the Accessibility Hierarchy are to be understood as specifying a set of possible grammatical distinctions that a language may make." In some languages, the hierarchy distinguishes subjects from all other grammatical functions: only the subject of a relative clause can be relativized, or interpreted as coreferent with the noun modified by the relative clause. Other languages allow relativization of subjects and objects in contrast to other grammatical functions. This more fine-grained hierarchical structure refines the subject/nonsubject distinction and allows more functional distinctions to emerge.

Keenan and Comrie speculate that their hierarchy can be extended to other processes besides relative clause formation, and indeed Comrie (1975) applies the hierarchy in an analysis of grammatical functions in causative constructions. In fact, the Keenan-Comrie hierarchy closely mirrors the "relational hierarchy" of Relational Grammar, as given by Bell (1983), upon which much work in Relational Grammar is based:

(3) Relational Hierarchy of Relational Grammar:

1 (SUBJ) > 2 (OBJ) > 3 (indirect object)

The Obliqueness Hierarchy of Head-Driven Phrase Structure Grammar (Pollard and Sag 1994) also reflects a hierarchy of grammatical functions like this one. As demonstrated by a large body of work in Relational Grammar, HPSG, LFG, and other theories, the distinctions inherent in these hierarchies are relevant across languages with widely differing constituent structure representations, languages that encode grammatical functions by morphological as well as configurational means. There is a clear and well-defined similarity across languages at this abstract level.

LFG assumes a universally available inventory of grammatical functions:

(4) Lexical Functional Grammar:

SUBJect, OBJect, OBJ$_\theta$, COMP, XCOMP, OBLique$_\theta$, ADJunct, XADJunct

The labels OBJ$_\theta$ and OBL$_\theta$ represent families of relations indexed by semantic roles, with the θ subscript representing the semantic role associated with the ar-

[1] The nomenclature that Keenan and Comrie use is slightly different from that used in this book: in their terminology, DO is the direct object, which we call OBJ; IO is the indirect object; OBL is an oblique noun phrase; GEN is a genitive/possessor of an argument; and OCOMP is an object of comparison.

gument. For instance, $\text{OBJ}_{\text{THEME}}$ is the member of the group of thematically restricted OBJ_{θ} functions that bears the semantic role THEME, and $\text{OBL}_{\text{SOURCE}}$ and OBL_{GOAL} are members of the OBL_{θ} group of grammatical functions filling the SOURCE and GOAL semantic roles.

Grammatical functions can be cross-classified in several different ways. The *governable grammatical functions* SUBJ, OBJ, OBJ_{θ}, COMP, XCOMP, and OBL_{θ} can be *subcategorized*, or required, by a predicate; these contrast with modifying adjuncts ADJ and XADJ, which are not subcategorizable.

The governable grammatical functions form several natural groups. First, one can distinguish the *core arguments* or *terms* (SUBJ, OBJ, and the family of thematically restricted objects OBJ_{θ}) from the family of *nonterm* or *oblique* functions OBL_{θ}. Crosslinguistically, term functions behave differently from nonterms in constructions involving anaphoric binding (Chapter 11) and control (Chapter 12); we will discuss other differences between terms and nonterms in Section 1.3 of this chapter.

Second, SUBJ and the primary object function OBJ are the *semantically unrestricted* functions, while OBL_{θ} and the secondary object function OBJ_{θ} are restricted to particular thematic or semantic roles, as the θ in their name indicates. Arguments with no semantic content, like the subject *it* of a sentence like *It rained*, can fill the semantically unrestricted functions, while this is impossible for the semantically restricted functions. We will discuss this distinction in Section 1.4 of this chapter.

Finally, *open* grammatical functions (XCOMP and XADJ), whose subject is controlled by an argument external to the function, are distinguished from *closed* functions. These will be discussed in Section 1.7 of this chapter.

Some linguists have considered inputs and outputs of relation-changing rules like passive to be good tests for grammatical functionhood: for example, an argument is classified as an object in an active sentence if it appears as a subject in the corresponding passive sentence, under the assumption that the passive rule turns an object into a passive subject. However, as we will discuss in Chapter 8, grammatical function alternations like passive are best viewed not in terms of transformational rules, or even in terms of lexical rules manipulating grammatical function assignment, but as alternative means of linking grammatical functions to semantic arguments. Therefore, appeal to these processes as viable diagnostics of grammatical functions requires a thorough understanding of the theory of argument linking, and these diagnostics must be used with care.

In the following, we present the inventory of grammatical functions assumed in LFG theory and discuss a variety of grammatical phenomena that make reference to these functions. Some of these phenomena are sensitive to a grammatical hierarchy, while others can refer either to specific grammatical functions or to the members of a larger class of functions. Thus, the same test (for example, relativizability) might distinguish subjects from all other grammatical functions in one language, but might pick out both subjects and objects in another language. A number of tests are also specific to particular languages or to particular types of languages: for example, switch-reference constructions, constructions in which a verb is inflected according to whether its subject is coreferential with the subject of another verb, do not constitute a test for subjecthood in a language in which switch-reference plays no grammatical role. In a theory like LFG, grammatical functions are theoretical primitives, not defined in phrasal or semantic terms; therefore, we do not define grammatical functions in terms of a particular, invariant set of syntactic behaviors. Instead, grammatical phenomena can be seen to cluster and distribute according to the grammatical organization provided by functional roles.

1.2. Governable Grammatical Functions and Modifiers

A major division in grammatical functions distinguishes arguments of a predicate from modifiers. The arguments are the *governable grammatical functions* of LFG; they are subcategorized for, or *governed*, by the predicate. Modifiers modify the phrase with which they appear, but they are not governed by the predicate.

(5) Governable grammatical functions:

$$\underbrace{\text{SUBJ OBJ XCOMP COMP OBJ}_\theta\ \text{OBL}_\theta}_{\text{GOVERNABLE GRAMMATICAL FUNCTIONS}} \qquad \underbrace{\text{ADJ XADJ}}_{\text{MODIFIERS}}$$

Linguists have proposed a number of identifying criteria for governable grammatical functions. Dowty (1982) proposes two tests to distinguish between governable grammatical functions and modifiers: what he calls the *entailment test*, namely that using a predicate entails the existence of all of its arguments, but not its modifiers; and what he calls the *subcategorization test*, namely that it is possible to omit modifiers but not arguments when a predicate is used. These tests do capture some intuitively correct properties of the distinction between governable grammatical functions and modifiers; however, neither test is completely successful in distinguishing between them.

Dowty's first test, the entailment test, fails for some phrases that seem uncontroversially to be modifiers. In particular, since the use of many predicates entails that some event occurred at some place at some time, the test implies that temporal modifiers are arguments of those predicates. For instance, the use of the verb *yawned* in a sentence like *David yawned* entails that there was some past time at which David yawned; however, few linguists would conclude on this basis that *previously* is an argument of *yawned* in a sentence like *David yawned previously*. Additionally, as pointed out by Anette Frank (p.c.), the entailment test incorrectly predicts that the object argument of an intensional verb such as *deny* or *seek* is not a governable grammatical function, since a sentence like *David is seeking a so-*

lution to the problem does not imply that a solution exists. Further, syntactically required but semantically empty phrases that are governed by a predicate are not classified as syntactic arguments by this test; the existence of some entity denoted by the subject of *rained* is not entailed by the sentence *It rained.*

Dowty's second test is also problematic. It clearly fails in "pro-drop" languages — languages where some or all arguments of a predicate can be omitted — but even in English the test does not work well. The test implies that because a sentence like *David ate* is possible, the object *lunch* in *David ate lunch* is not an argument but a modifier.

Even though Dowty's tests do not succeed in correctly differentiating arguments and modifiers, certain valid implications can be drawn from his claims. If a phrase is an argument, it is either obligatorily present or it is entailed by the predicate. If a phrase is a modifier, it can be omitted. Stronger conclusions do not seem to be warranted, however.

A number of other tests have been shown to illuminate the distinction between arguments and modifiers:

MULTIPLE OCCURRENCE: Modifiers can be multiply specified, but arguments cannot, as noted by Kaplan and Bresnan (1982):

(6) a. *The girl handed the baby a toy on Tuesday in the morning.*
b. **David saw Tony George Sally.*

ANAPHORIC BINDING PATTERNS: In some languages, binding patterns are sensitive to the syntactic argument structure of predicates and therefore to the argument/modifier distinction. For example, the Norwegian reflexive pronoun *seg selv* requires as its antecedent a coargument of the same predicate. Since a modifier is not an argument of the main predicate, the reflexive *seg selv* may not appear in a modifier phrase if its antecedent is an argument of the main verb (Hellan 1988; Dalrymple 1993). The subscript *i* in the glosses of the following examples indicates coreference between an anaphor and its intended antecedent:

(7) *Jon forakter seg selv.*
Jon despises self
'Jon$_i$ despises himself$_i$.'

(8) *Jon fortalte meg om seg selv.*
Jon told me about self
'Jon$_i$ told me about himself$_i$.'

(9) **Hun kastet meg fra seg selv.*
She threw me from self
'She$_i$ threw me away from herself$_i$.'

ORDER DEPENDENCE: The contribution of modifiers to semantic content can depend upon their relative order, as noted by Pollard and Sag (1987, section 5.6). The meaning of a sentence may change if its modifiers are reordered:

(10) a. *Kim jogged for twenty minutes twice a day.*

b. *Kim jogged twice a day for twenty years.*

(11) a. *Kim jogged reluctantly twice a day.*

b. *Kim jogged twice a day reluctantly.*

In contrast, reordering arguments may affect the rhetorical structure of the sentence, focusing attention on one or another argument, but does not alter the conditions under which the sentence is true.

EXTRACTION PATTERNS: A long-distance dependency cannot relate a wh-phrase that appears in sentence-initial position to a position inside some modifiers, as noted by Pollard and Sag (1987, section 5.6) (see also Huang 1982; Rizzi 1990):

(12) a. **Which famous professor did Kim climb K2 without oxygen in order to impress ___?*

b. *Which famous professor did Kim attempt to impress ___ by climbing K2 without oxygen?*

This generalization is not as robust as those discussed above, since as Pollard and Sag point out, it is possible to extract a phrase from some modifiers:

(13) *Which room does Julius teach his class in ___?*

1.3. Terms and Nonterms

The governable grammatical functions can be divided into *terms* or direct functions, and *nonterms* or obliques. The subject and object functions are grouped together as terms:[2]

(14) Terms and nonterms:

$$\underbrace{\text{SUBJ OBJ OBJ}_\theta}_{\text{TERMS}} \quad \underbrace{\text{OBL}_\theta \text{ XCOMP COMP}}_{\text{NONTERMS}}$$

A number of tests for termhood in different languages have been proposed:

[2] Relational grammar (Perlmutter and Postal 1983a) also recognizes this basic division of grammatical functions into "term relations" and "oblique relations." Terms are also sometimes referred to as "core functions" (Andrews 1985; Bresnan 2001b).

AGREEMENT: In some languages, termhood is correlated with verb agreement; in fact, this observation is encoded in Relational Grammar as the Agreement Law (Frantz 1981): "Only nominals bearing term relations (in some stratum) may trigger verb agreement." Alsina (1993), citing Rosen (1990) and Rhodes (1990), notes that all terms, and only terms, trigger verb agreement in Ojibwa and Southern Tiwa.

ANAPHORIC BINDING PATTERNS: In some languages, terms behave differently from obliques with respect to anaphoric binding. Sells (1988) shows that in Albanian, a term can antecede a term or oblique reflexive, while an oblique only antecedes another oblique. Among the term arguments, possible binding relations are constrained by a thematic hierarchy. Hellan (1988), Dalrymple and Zaenen (1991), and Dalrymple (1993) discuss Norwegian data that point to a similar conclusion.

CONTROL: Kroeger (1993) shows that in Tagalog, only a term can be the controllee in the participial complement construction, and only a term can be a controller in the participial adjunct construction.

Alsina (1993) provides an extensive discussion of termhood in a number of typologically very different languages, and Andrews (1985) further discusses the term/nonterm distinction.

Often, discussion of terms focuses exclusively on the status of nominal arguments of a predicate and does not bear on the status of verbal or sentential arguments. The infinitive phrase *to be yawning* in example (15) bears the open grammatical function XCOMP:

(15) *Chris seems to be yawning.*

The sentential complement *that Chris was yawning* bears the grammatical function COMP in (16):

(16) *David thought that Chris was yawning.*

The XCOMP function differs from the COMP function in not containing an overt SUBJ internal to its phrase; XCOMP is an open function, whose SUBJ is determined by means of lexical specifications on the predicate that governs it, as discussed in Section 1.7 of this chapter. What is the termhood status of the XCOMP and COMP arguments?

Zaenen and Engdahl (1994) classify XCOMP as a kind of oblique in their analysis of the linking of sentential and predicative complements, though without providing specific evidence in support of this classification. Oblique arguments are nonterms, and so if Zaenen and Engdahl are correct, XCOMP would be classified as a nonterm.

Word order requirements on infinitival and finite complements in English provide some support for this position. Sag (1986) claims that in English, term phrases always precede obliques:

(17) a. *David gave a book to Chris.*

b. **David gave to Chris a book.*

Infinitival and sentential complements bearing the grammatical functions XCOMP and COMP obey different word order restrictions from term noun phrases. The following data indicate that XCOMPs are obliques:

(18) a. *Kim appeared to Sandy to be unhappy.*

b. *Kim appeared to be unhappy to Sandy.*

Since the XCOMP *to be unhappy* is not required to precede the oblique phrase *to Sandy* but can appear either before or after it, Sag's diagnostic indicates that the XCOMP must also be an oblique. Similar data indicate that the COMP is also an oblique phrase:

(19) a. *David complained that it was going to rain to Chris.*

b. *David complained to Chris that it was going to rain.*

We will return to a discussion of COMP and XCOMP in Section 1.7 of this chapter.

1.4. Semantically Restricted and Unrestricted Functions

The governable grammatical functions can be divided into *semantically restricted* and *semantically unrestricted* functions (Bresnan 1982a):

(20) Semantically unrestricted and restricted functions:

$$\underbrace{\text{SUBJ OBJ}}_{\text{SEMANTICALLY UNRESTRICTED}} \qquad \underbrace{\text{OBJ}_\theta\ \text{OBL}_\theta}_{\text{SEMANTICALLY RESTRICTED}}$$

Semantically unrestricted functions like SUBJ and OBJ can be associated with any semantic role, as Fillmore (1968) shows:

(21) a. *He hit the ball.*

b. *He received a blow.*

c. *He received a gift.*

d. *He loves her.*

e. *He has black hair.*

The examples in (21) show that the SUBJ of different verbs can be associated with different semantic roles: AGENT in a sentence like *He hit the ball*, GOAL in a sentence like *He received a blow*, and so on. Similar examples can be constructed for OBJ.

In contrast, members of the semantically restricted family of functions OBJ$_\theta$ and OBL$_\theta$ are associated with a particular semantic role. For example, the OBJ$_{\text{THEME}}$ function is associated only with the semantic role of THEME, and the OBL$_{\text{GOAL}}$ is associated with GOAL. Languages may differ in the inventory of semantically restricted functions they allow. For example, English allows only OBJ$_{\text{THEME}}$:

(22) a. *I gave her a book.*

b. *I made her a cake.*

c. *I asked him a question.*

Other semantic roles cannot be associated with the second object position:

(23) a. **I made a cake the teacher.*

b. **I asked a question David.*

Section 1.6 of this chapter provides a more complete discussion of the double object construction and verb alternations; see also Levin (1993).

The division between semantically restricted and semantically unrestricted arguments predicts what in Relational Grammar is called the Nuclear Dummy Law (Frantz 1981; Perlmutter and Postal 1983a): only semantically unrestricted functions can be filled with semantically empty arguments like the subject *it* of *It rained*. This is because the semantically restricted functions are associated only with a particular semantic role; since a semantically empty argument is incompatible with these semantic requirements, it cannot appear in these positions.

The functions XCOMP and COMP seldom figure in discussions of semantically restricted and unrestricted arguments, and it is not completely clear how they should be classified. There does seem to be some pretheoretic evidence for classifying COMP as semantically *unrestricted*, since different semantic entailments can attach to different uses of XCOMP and COMP. If these different semantic entailments are taken to delineate distinctions among different members of a set of semantic roles, then this would mean that XCOMP and COMP should be classified as semantically unrestricted.

In a pioneering paper, Kiparsky and Kiparsky (1970) note that sentential arguments bearing the COMP function may be *factive* or *nonfactive* with respect to their complements: for factive complements, “the embedded clause expresses a true proposition, and makes some assertion about that proposition,” whereas such a presupposition is not associated with nonfactive complements. Kiparsky and Kiparsky also distinguish *emotive* from *nonemotive* sentential arguments; emotive complements are those to which a speaker expresses a "subjective, emotional, or evaluative reaction":

(24) a. Factive emotive: *I am pleased that David came.*

b. Factive nonemotive: *I forgot that David came.*

c. Nonfactive emotive: *I intend that David come.*

d. Nonfactive nonemotive: *I suppose that David came.*

It is not clear, however, whether the semantic differences explored by Kiparsky and Kiparsky should be taken to indicate that these arguments, which all bear the grammatical function COMP in English, bear different semantic roles. We leave this question for future research.

We have explored several natural classes of grammatical functions: governable grammatical functions and modifiers, terms and nonterms, semantically restricted and unrestricted functions. We now turn to an examination of particular grammatical functions, beginning with the subject function.

1.5. SUBJ

The subject is the term argument that ranks the highest on the Keenan-Comrie relativization hierarchy. As discussed in Section 1.1 of this chapter, their hierarchy is applicable to other processes besides relativization: if only a single type of argument can participate in certain processes for which a functional hierarchy is relevant, that argument is often the subject.

There is no lack of tests referring specifically to the subject function:

AGREEMENT: The subject is often the argument that agrees with the verb in languages in which verbs bear agreement morphology; indeed, Moravcsik (1978) proposes the following language universal:

> There is no language which includes sentences where the verb agrees with a constituent distinct from the intransitive subject and which would not also include sentences where the verb agrees with the intransitive subject. (Moravcsik 1978, page 364)

English is a language that exhibits subject-verb agreement; the fullest paradigm is found in the verb *to be*:

(25) *I am / You are / He is*

HONORIFICATION: Matsumoto (1996) calls this the most reliable subject test in Japanese. Certain honorific forms of verbs are used to honor the referent of the subject:

(26) *sensei wa hon o o-yomi ni narimashi-ta*
teacher TOPIC book ACC HONORIFIC-read COPULA become.POLITE-PAST
'The teacher read a book.'

The verb form *o-V ni naru* is used to honor the subject *sensei* 'teacher'. It cannot be used to honor a nonsubject, even if the argument is a "logical subject"/AGENT:

(27) **Jon wa sensei ni o-tasuke-rare ni nat-ta*
John TOPIC teacher by HONORIFIC-help-PASSIVE COPULA become-PAST
'John was saved by the teacher.'

SUBJECT NONCOREFERENCE: Mohanan (1994) shows that the antecedent of a pronoun in Hindi cannot be a subject in the same clause, although a nonsubject antecedent is possible:

(28) *Vijay ne Ravii ko uskii saikil par biṭhaayaa*
Vijay ERG Ravi ACC his bicycle LOC sit.CAUSATIVE.PERFECT
'Vijay$_i$ seated Ravi$_j$ on his$_{*i,j}$ bike.'

LAUNCHING FLOATED QUANTIFIERS: Kroeger (1993, page 22) shows that the subject launches *floating quantifiers*, quantifiers that appear outside the phrase they quantify over, in Tagalog.[3]

(29) *sinusulat lahat ng-mga-bata ang-mga-liham*
IMPERFECT.write.OBJECTIVE all GEN-PL-child NOM-PL-letter
'All the letters are written by the/some children.'
(Does not mean: 'All the children are writing letters.')

Bell (1983, pages 154 ff.) shows that the same is true in Cebuano.

This is only a sampling of the various tests for subjecthood. Many other tests could, of course, be cited (see, for example, Li 1976; Zaenen 1982; Zaenen et al. 1985).

The question of whether all verbal predicates in every language must contain a subject is a vexed one. The Subject Condition[4] was discussed by Bresnan and

[3] Kroeger attributes example (29) to Schachter (1976).

[4] The Subject Condition is called the *Final 1 Law* in Relational Grammar (Frantz 1981; Perlmutter and Postal 1983a) and the *Extended Projection Principle* in Government and Binding Theory (Chomsky 1981).

Kanerva (1989), who attribute it originally to Baker (1983) (see also Andrews 1985; Levin 1987; Butt et al. 1999):

(30) Subject Condition:

Every verbal predicate must have a SUBJ.

Though the Subject Condition seems to hold in English, and perhaps in many other languages as well, there are languages in which the requirement does not so clearly hold. For example, German impersonal passives, as in (31), are traditionally analyzed as subjectless clauses:

(31) ... *weil getanzt wurde*
because danced was
'because there was dancing'

However, Berman (1999) claims that clauses like (31) contain an unpronounced expletive subject, and thus that the Subject Condition is not violated.

Other cases of apparently subjectless clauses are also found. Simpson (1991, page 29) notes that subjects of participial modifiers in Russian are required to corefer with the matrix subject:

(32) *bystro temneja, tuča pokryla vse nebo.*
quickly darken.PARTICIPLE cloud.FEM.NOM cover.PAST.FEM all sky
'As it quickly darkened, the cloud covered the whole sky.'

However, some weather verbs in Russian appear to be subjectless and cannot appear with participles which require subject control:

(33) * *temneja, stalo očen' xolodno.*
darken.PARTICIPLE become.PAST.NEUT very cold.NEUT
'When getting dark, it became very cold.'

If Russian obeyed the Subject Condition, example (33) would be expected to be grammatical. It may be, then, that the Subject Condition is a language-particular requirement imposed by some but not all languages, rather than a universal requirement.

1.6. The Object Functions

Grammatical phenomena in which a grammatical function hierarchy is operative may sometimes group subject and object arguments together in distinction to other arguments, and in fact a number of grammatical processes refer to the subject and object functions in distinction to other grammatical functions. Other phenomena are describable specifically in terms of the object function; for purposes of our current discussion, these object tests are more interesting. Some of these are:

AGREEMENT: As noted in Section 1.3 of this chapter, terms are often registered by agreement morphemes on the verb. Often, the object is uniquely identified by agreement: some languages have object agreement. For example, Georgopoulos (1985) describes OBJ agreement in Palauan:

(34) *ak-uldenges-terir a resensei er ngak*
1SG.PERFECT-honor-3PL teachers PREP me
'I respected my teachers.'

In (34), the morpheme *-terir* shows third person plural agreement with the OBJ *a resensei* 'teachers'.

CASEMARKING: In some limited circumstances, objects can be distinguished by casemarking, though this test must be used with care: in general, there is no one-to-one relation between the morphological case that an argument bears and its grammatical function, as we will see in Section 4.1 of this chapter. Mohanan (1982) discusses casemarking in Malayalam, showing that ACCusatively marked noun phrases are unambiguously objects (see also Mohanan 1994, pages 89–90) :

(35) *kuṭṭi aanaye ṉuḷḷi*
child elephant.ACC pinched
'The child pinched the elephant.'

However, Mohanan goes on to show that many phrases in Malayalam that are OBJ are not marked with ACC case. That is, every phrase in Malayalam that is ACC is an OBJ, but not all OBJS are ACC.

RELATIVIZATION: Givón (1997, section 4.4.3) notes that only subjects and objects can be relativized in Kinyarwanda, and only objects can be relativized with a gap; relativization of subjects requires the use of a resumptive pronoun.

Further discussion of object tests is provided by Baker (1983) for Italian and Dahlstrom (1986b) for Cree. Andrews (1985) also gives a detailed discussion of object tests in various languages.

1.6.1. MULTIPLE OBJECTS

Many languages have more than one phrase bearing an object function. English is one such language:

(36) *He gave her a book.*

Zaenen et al. (1985) discuss Icelandic, another language with multiple object functions, and note the existence of asymmetries between the two kinds of objects. For instance, the primary object can be the antecedent of a reflexive contained in the secondary object:

(37) *Ég gaf ambáttina [konungi sínum].*
I gave slave.DEF.ACC king.DAT self's
'I gave the slave$_i$ (OBJ) to self's$_i$ king (OBJ2).'

However, the secondary object cannot antecede a reflexive contained in the primary object:

(38) **Sjórinn svipti manninum [gömlu konuna sína].*
sea.DEF deprived man.DEF.DAT old wife.DEF.ACC self's
'The sea deprived of the man$_i$ (OBJ2) self's$_i$ old wife (OBJ).'

Dryer (1987) also presents an extensive discussion of the behavior of objects in languages with multiple OBJ functions and of their groupings with respect to semantic roles.

Earlier work in LFG concentrated on languages like English and Icelandic, which each have two object functions. In such languages, the primary object was called the OBJ, and the secondary object was called the OBJ2, as in examples (37–38). Further research has expanded our knowledge of the properties of objects, and in later work, it became evident that this simple classification was neither sufficient nor explanatory.

In fact, languages allow a single *thematically unrestricted* object, the primary OBJ. In addition, languages may allow one or more secondary, *thematically restricted* objects. That is, the argument that was originally identified as OBJ2 in English is only one member of a family of semantically restricted functions, referred to collectively as OBJ$_\theta$ (Bresnan and Kanerva 1989). This classification more clearly reflects the status of secondary objects as restricted to particular semantic roles, and also encompasses analyses of languages whose functional inventory includes more than two object functions.

In English, as discussed in Section 1.4 of this chapter, the thematically restricted object must be a theme; other semantic roles, such as goal or beneficiary, are not allowed:

(39) a. *I made her a cake.*

b. **I made a cake her.*

In contrast, as Bresnan and Moshi (1990) show, languages like Chaga allow multiple thematically restricted objects with roles other than THEME:[5]

[5] Numbers in the glosses indicate the noun class of the arguments.

(40) *n-ấ-lꜝé-kú-shí-kí-kóṛ-í-à*
FOCUS-1SUBJ-PAST-17OBJ-8OBJ-7OBJ-cook-APPLICATIVE-FV
'She/he cooked it with them there.'

This example contains three object markers, representing a locative object, an instrumental object, and a patient object. According to Bresnan and Moshi's analysis, in this example the instrumental OBJ is the unrestricted OBJ; the locative and patient arguments bear thematically restricted OBJ functions OBJ_{LOC} and $\text{OBJ}_{\text{PATIENT}}$. Bresnan and Moshi provide much more discussion of OBJ_{θ} in Chaga and other Bantu languages.

1.6.2. 'DIRECT' AND 'INDIRECT' OBJECT

In traditional grammatical descriptions, the grammatical function borne by *her* in the English example in (41) has sometimes been called the "indirect object," and *the book* has been called the "direct object":

(41) *He gave her a book.*

The phrase *the book* is also traditionally assumed to be a direct object in examples like (42):

(42) *He gave a book to her.*

The classification of *the book* as a direct object in both (41) and (42) may have a semantic rather than a syntactic basis: there may be a tendency to assume that *the book* must bear the same grammatical function in each instance because its semantic role does not change. As we have seen, the LFG view differs: in example (41), the phrase *her* bears the OBJ function, while in example (42), the phrase *a book* is the OBJ.

Within the transformational tradition, evidence for the LFG classification for English came from certain formulations of the rule of passivization, which applies uniformly to "transform" an object into a subject:

(43) a. *He gave her a book.*
She was given a book.

b. *He gave a book to her.*
A book was given to her.

If the "passive transformation" is stated in terms of the indirect object/object distinction, or its equivalent in phrase structure terms, the generalization is complicated to state: the direct object becomes the passive subject only if there is no indirect object present; otherwise, the indirect object becomes the subject. On the other hand, the transformation is easy to state if the first noun phrase following the verb is classified as the object and the second bears some other function.

In LFG, however, the theory of grammatical function alternations is formulated in terms of a characterization of possible mappings between semantic roles and grammatical functions, as described in Chapter 8, and is not transformational in nature. Thus, we must look to other grammatical phenomena for evidence bearing on the classification of object functions.

Dryer (1987) presents several arguments that in English, an opposition between primary/unrestricted objects (OBJ) and secondary/restricted objects (OBJ_θ), as proposed in LFG, allows a more satisfactory explanation of the facts than the direct/indirect object distinction. Dryer primarily discusses evidence from prepositional casemarking and word order. For example, given a distinction between primary and secondary objects, we can succinctly describe word order within the English VP: the primary object immediately follows the verb, with the secondary object following it.[6]

In other languages, the situation is even clearer. Alsina (1993) examines the object functions in Chicheŵa and their role in the *applicative* construction. In this construction, an affix is added to the verb that signals a requirement for an additional syntactic argument besides the arguments ordinarily required by the verb; we focus here on the *benefactive applicative* construction, in which the applicative affix signals that an OBJ argument bearing a beneficiary thematic role is required. Alsina (1993) shows that the syntactic OBJ properties of the patient argument in the nonapplied form are displayed by the beneficiary argument in the applied form. The primary/nonrestricted OBJ is the argument that immediately follows the verb; this argument is the patient in the nonapplied form, and the beneficiary in the applied form of the verb:

(44) a. *nkhandwe zi-ku-mény-á njōvu*
10.foxes 10SUBJ-PRES-hit-FV 9.elephant
'The foxes are hitting the elephant.'

b. *nkhandwe zi-ku-mény-ér-a aná njōvu*
10.foxes 10SUBJ-PRES-hit-APPLICATIVE-FV 2.children 9.elephant
'The foxes are hitting the elephant for the children.'

The patient argument alternates with the OBJ marker in the nonapplied form, and the beneficiary argument alternates with the OBJ marker in the applied form:

(45) a. *nkhandwe zi-ku-í-me̋ny-a*
10.foxes 10SUBJ-PRES-9OBJ-hit-FV
'The foxes are hitting it.'

[6] Dryer assumes a more complicated crosslinguistic typology of object functions than is generally accepted in LFG. His richer typology turns out to be best explained in terms of different strategies for relating semantic roles to object grammatical functions, as described in Chapter 8.

b. *nkhandwe zi-ku-wá-mény-er-á* *njōvu*
10.foxes 10SUBJ-PRES-9OBJ-hit-APPLICATIVE-FV 9.elephant
'The foxes are hitting the elephant for them.'

This and other evidence is best explained by assuming that the patient arguments in (44a) and (45a) and the beneficiary arguments in (44b) and (45b) bear the non-restricted/primary OBJ function, while the patient arguments in (44b) and (45b) bear the restricted/secondary OBJ$_\theta$ function and behave differently. In other words, the syntactic behavior of the arguments in examples (44) and (45) is best explained by reference to a distinction between OBJ and OBJ$_\theta$, not between direct and indirect objects.

1.7. COMP, XCOMP, and XADJ

The COMP, XCOMP, and XADJ functions are clausal functions, differing in whether or not they contain a overt SUBJ noun phrase internal to their phrase. The COMP function is a *closed* function containing an internal SUBJ phrase. The XCOMP and XADJ functions are *open* functions that do not contain an internal subject phrase; their SUBJ must be specified externally to their phrase.[7]

(46) Open and closed functions:

$$\underbrace{\text{SUBJ OBJ COMP OBJ}_\theta\ \text{OBL}_\theta\ \text{ADJ}}_{\text{CLOSED}}\quad \underbrace{\text{XCOMP XADJ}}_{\text{OPEN}}$$

The COMP function is the function of sentential complements, familiar from traditional grammatical description. A COMP can be declarative, interrogative, or exclamatory (Culy 1994):

(47) a. *David complained that Chris yawned.*

b. *David wondered who yawned.*

c. *David couldn't believe how big the house was.*

The XCOMP function is an open complement function, the one borne by a phrase like *to yawn* in the examples in (48). In those examples, the SUBJ of the XCOMP is also an argument of the matrix verb, *David* in both of the examples in (48):

(48) a. *David seemed to yawn.*

b. *Chris expected David to yawn.*

[7] Arka and Simpson (1998) propose that some control constructions in Balinese involve an open SUBJ function: for instance, in the Balinese equivalent of*I tried to take the medicine*, the infinitive phrase *to take the medicine* can bear the SUBJ function, with its SUBJ controlled by the term argument *I*. We do not explore this possibility further here.

Like XCOMP, the XADJ function is an open function, whose SUBJ must be specified externally; unlike XCOMP, XADJ is a modifier, not a governable grammatical function. In example (49), the SUBJ of the XADJ *stretching his arms* is also the SUBJ of the matrix clause, *David*:

(49) *Stretching his arms, David yawned.*

We will return to a discussion of XCOMP, XADJ, and control in Chapter 12.

There has not been universal agreement as to the status of the grammatical function COMP. Alsina (1993) claims that the COMP function is actually superfluous and that sentential complements are best analyzed as bearing the function OBJ. On this view, any difference between nominal objects and sentential complements follows solely from their difference in phrase structure category, since at the functional level they both bear the OBJ function.

In fact, however, several arguments can be made against discarding the grammatical function COMP altogether: there are phenomena that can only be explained by assuming the existence of the grammatical function COMP as distinct from OBJ. First, if all sentential complements are OBJ and not COMP, they would be classified as terms. In this case, the evidence presented in Section 1.3 of this chapter, indicating that English has sentential complements that are not terms, would remain unexplained. Second, if English sentential complements are analyzed as objects, then we must assume that English admits sentences with three OBJ functions, but only when one of the OBJ functions is sentential rather than nominal:

(50) *David bet [Chris] [five dollars] [that she would win].*

Most importantly, there is evidence for a split in the syntactic behavior of sentential complements in a number of languages; as discussed by Dalrymple and Lødrup (2000), this evidence is best explained by analyzing some sentential complements in these languages as OBJ, and some as COMP. In Swedish, clausal complements bearing the OBJ function can be pronominalized and can topicalize, as shown in examples (51a–c):[8]

(51) a. *Man antar att sosserna vinner valet.*
One assumes that social.democrats.DEF win election.DEF
'One assumes that the Social Democrats will win the election.'

b. *Man antar det.*
One assumes that
'One assumes that.'

c. *Att sosserna vinner valet antar man.*
That social.democrats.DEF win election.DEF assumes one
'That the Social Democrats will win the election, one assumes.'

[8] Examples (51a–c) are due to Engdahl (1999).

In contrast, Swedish complement clauses bearing the COMP function do not display these properties:

(52) a. *Kassören yrkade att avgiften skulle höjas.*
cashier.DEF insisted that tax.DEF should be.increased
'The cashier insisted that the tax should be increased.'

b. **Kassören yrkade det.*
cashier.DEF insisted that
'The cashier insisted it.'

c. **Att avgiften skulle höjas yrkade kassören.*
That tax.DEF should be.increased insisted cashier.DEF
'That the tax should be increased, the cashier insisted.'

As Dalrymple and Lødrup (2000) show, other languages also show a similar split in behavioral properties of sentential complements, with some sentential complements patterning with nominal OBJ arguments and others exhibiting behavior typical of COMP arguments. Thus, the COMP grammatical function cannot be eliminated from grammatical description, since many sentential complements must be analyzed as bearing the COMP function.

1.8. Oblique Arguments: OBLique$_\theta$

Oblique arguments are those that are associated with particular semantic roles and marked to indicate their function overtly. In languages like English, oblique arguments are prepositional phrases, while in other languages, as discussed by Nordlinger (1998), oblique arguments are casemarked rather than appearing as prepositional or postpositional phrases.

LFG assumes that there are two types of oblique arguments (Bresnan 1982a). Arguments of the first type are marked according to the semantic role of the argument, such as the goal *to*-phrase of a verb such as *give*. This class corresponds to the category of *semantic case* in the casemarking classification scheme of Butt and King (1999a), since semantic case is governed by generalizations about the relation between case and semantic role.

Arguments of the second type are marked idiosyncratically, and the form of the casemarking is lexically specified by the governing predicate. This class corresponds to the category of *quirky case* in Butt and King's classification scheme.[9]

1.8.1. Semantically Marked Obliques

The phrase *to Chris* in example (53) bears the OBL$_{\text{GOAL}}$ grammatical function:

[9] As Butt and King (1999a) point out, semantic and quirky case can appear on terms as well as obliques. Butt and King also discuss *structural case* and *default case*, which appear on terms.

(53) *David gave the book to Chris.*

The semantic role of the OBL$_{\text{GOAL}}$ argument is marked by the preposition *to*. It is not possible for more than one oblique argument to have the same semantic role:

(54) **David gave the book to Chris to Ken.*

In languages like Warlpiri, an OBL$_{\text{LOC}}$ phrase such as *kirri-ngka* 'large camp' is marked with locative casemarking rather than a preposition or postposition (Simpson 1991; Nordlinger 1998):

(55) *kirri-ngka* *wiri-ngka-rlipa* *nyina-ja*
large.camp-LOC big-LOC-1PL.INCLUSIVE.SUBJ sit-PAST
'We sat in the large camp.'

Locative casemarking plays a similar role to the preposition in example (54), to mark the semantic role of the argument.

1.8.2. Idiosyncratic Prepositional Marking

An oblique phrase may also be required to bear a particular form unrelated to the semantic role of the argument. For such cases, Bresnan (1982a) suggests the presence of a FORM feature that is specified by the predicate, as in (56):

(56) *Chris relied on/*to/*about David.*

In this case, the form of the preposition *on* in the phrase *on David* is stipulated by the predicate *relied*. Butt et al. (1999) provide more discussion of oblique phrases with idiosyncratic prepositional marking.

1.9. Other Functional Attributes

The table on page 28 gives a list of some of the more commonly assumed f-structural features together with the values of these features (see also Butt et al. 1999). The appearance and distribution of these f-structural features is defined in terms of functional syntactic information, and so their presence at f-structure is crucial: CASE and agreement features are associated with particular grammatical functions; features specifying form, such as VFORM, are relevant at a functional syntactic level for specifying the required morphological form of an argument; and "sequence of tense" phenomena govern syntactic requirements on tense and aspect realization. Only features that can be argued to play a role in functional syntactic constraints are represented at f-structure; Chapter 7 discusses the non-syntactic structures of LFG, the features they contain, and their relation to functional structure.

	Feature	Value
Person:	PERS	Set of atomic values (see Chapter 13)
Gender:	GEND	Set of atomic values (see Chapter 13)
Number:	NUM	SG, DUAL, PL,...
Case:	CASE	Set of case values NOM, ACC,...(see Chapter 13)
Prepositional "case":	PCASE	The family of grammatical functions OBL_θ
Surface form:	FORM	Surface word form
Verb form:	VFORM	PASTPART, PRESPART,...
Complementizer form:	COMPFORM	Surface form of complementizer: THAT, WHETHER,...
Tense:	TENSE	PRES, PAST,...
Aspect:	ASPECT	F-structure representing complex description of sentential aspect. Sometimes abbreviated as e.g. PRES.IMPERFECT
Pronoun type:	PRONTYPE	REL, WH,...

2. SUBCATEGORIZATION

At a minimum, the information that must be lexically associated with a word is its meaning. Research has shown that the syntactic behavior of a word can be partially predicted from this information; this is because a number of regularities govern the relation between the meaning of a predicate and the grammatical functions of its arguments, as we will discuss in detail in Chapter 8. LFG and other linguistic theories define and capitalize on this relation in their theory of syntactic subcategorization.

LFG assumes that syntactic subcategorization requirements of predicates are stated at the f-structure level, in functional rather than phrasal terms. Predicates require a set of arguments bearing particular semantic roles. These roles are associated with grammatical functions according to a theory of argument mapping, to be discussed in Chapter 8. In turn, these grammatical functions are realized at the level of constituent structure in a variety of ways, as required by particular languages: in some languages, grammatical functions are associated with particular phrase structure positions, while in other languages, grammatical functions may be signaled by particular kinds of morphological marking on the head or on the argument (see Chapter 5, Section 4).

In contrast to this view, and in line with proposals in transformational grammar (Chomsky 1965), some linguistic theories state subcategorization requirements in phrase structure terms rather than in terms of abstract functional syntactic organization. There are many reasons to question the viability of this position, since the bulk of phrase structure information is never relevant to the satisfaction of subcategorization requirements. As Grimshaw (1982) points out, predicates never vary idiosyncratically in terms of which phrasal position they require their arguments to be in; for example, there are no exceptional verbs in English which require their objects to appear preverbally rather than postverbally. Subcategorization according to constituent structure configuration rather than functional structure leads to the incorrect expectation that such exceptional verbs should exist. In fact, however, we can cleanly state subcategorization requirements in terms of abstract functional structure; the claim that all phrasal and configurational information is always relevant to subcategorization is too strong.

There is evidence that one particular type of constituent structure information may in some cases be relevant to subcategorization requirements: cases in which a predicate idiosyncratically requires an argument of a particular phrasal category. Other kinds of phrasal information, such as position, never play a role in subcategorization requirements. However, one must take care in identifying situations in which such requirements seem to hold. Often, as Maling (1983) demonstrates, apparent evidence for subcategorization for a particular phrase structure category turns out on closer examination to be better analyzed as a requirement for an argument of a particular semantic type, together with a strong correlation between that type and a particular phrasal category most often used to express it. Maling notes that predicates like *seem* have often been claimed to require adjective phrase complements and to disallow prepositional phrase complements:

(57) a. *Sandy seems clever.*

b. **Sandy seems out of town.*

However, Maling shows that the true criterion at work in these examples is not based on phrase structure category, but is semantic in nature: only what Maling calls *gradable predicates*, those that can hold to a greater or lesser degree, are acceptable as complements of *seem*. Many prepositional phrases do not express gradable predicates, accounting for the unacceptability of example (57b). However, prepositional phrases that denote gradable predicates are acceptable as complements of *seem*:

(58) a. *That suggestion seemed completely off the wall.*

b. *Lee sure seems under the weather.*

Further, as Maling shows, adjective phrases that are not gradable predicates are unacceptable as complements of *seem*. In the following examples, the adjective *irrational* as a description of a mental state is gradable and can be used as the complement of *seems*, while as a technical mathematical term it is not gradable and cannot be used:

(59) a. *Lee seems irrational.*

b. **The square root of two seems irrational.*

In some cases, then, requirements that appear to depend on phrase structure category prove on closer inspection to be functional or semantic in nature.

In other cases, however, the particular constituent structure category of the complement is at issue, and no functional or semantic distinction is involved. The circumstances under which these extra specifications are necessary are rare: subcategorization for a particular phrasal category is a marked exception rather than the general rule. In Chapter 6, Section 4.3, we discuss these cases, showing that the phrase structure category of a complement can be specified in these limited cases.

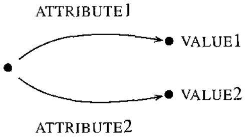
3. FUNCTIONAL STRUCTURE REPRESENTATION

In LFG, functional information is formally represented by the *functional structure* or *f-structure*. Mathematically, the f-structure can be thought of as a function[10] from attributes to values, or equivalently as a set of pairs, where the first member of the pair is an attribute and the second is its value. There is a simple and common way of representing f-structures in tabular form, that is, as a table of attributes and values:[11]

(60)
$$\begin{bmatrix}\text{ATTRIBUTE1} & \text{VALUE1}\\ \text{ATTRIBUTE2} & \text{VALUE2}\end{bmatrix}$$

[10] A function is a special kind of relation which assigns a*unique* value to its argument. For example, the relation between a person and his or her age is a function, since every person has exactly one age. The relation between a person and his or her children is not a function, since some people have no children and some people have more than one child.

[11] In some literature, particularly in Head-Driven Phrase Structure Grammar (see, for example, Pollard and Sag 1994), the objects that are represented in LFG as structures like (60) are instead represented via diagrams such as:



Attributes are labeled arcs in the diagram, and values are nodes. A sequence of attributes, a*path* through the f-structure, corresponds to the traversal of several labeled arcs. A possible source of confusion for those trained within the HPSG framework is that the same formal notation used to represent LFG functional structures in examples like (60) is used to represent*constraints* on structures in HPSG. What is depicted in (60) is not a constraint; it is a formal object.

3.1. Simple F-Structures

The following is a simplified f-structure for the proper noun *David*:

(61)
$$\begin{bmatrix} \text{PRED} & \text{`DAVID'} \\ \text{NUM} & \text{SG} \end{bmatrix}$$

This f-structure does not contain all the syntactic information that *David* contributes. We assume here and elsewhere that the full f-structure representation for the examples we exhibit contains at least the information shown, but may also contain other information not relevant to the particular point under discussion.

The f-structure in (61) contains two attributes: PRED and NUM. The value of NUM is SG, indicating a value of singular for the number feature. The value SG is an *atomic* value.

For the sentence *David yawned*, we have the following f-structure:

(62)
$$g\begin{bmatrix} \text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & f\begin{bmatrix} \text{PRED} & \text{`DAVID'} \\ \text{NUM} & \text{SG} \end{bmatrix} \end{bmatrix}$$

As (62) shows, f-structures can themselves be values of attributes: here, the value of the attribute SUBJ is the f-structure for the subject of the sentence. We can annotate f-structures with labels for subsequent reference; in (62), we have annotated the SUBJ f-structure with the label f and the f-structure for the sentence with the label g.

3.2. Semantic Forms

The value of the PRED attribute is special: it is a *semantic form.* A full discussion of semantic forms will be presented in Chapter 5, Section 2.2; additionally, Chapter 9 presents a more complete discussion of the information that semantic forms represent. In example (62), the semantic form value of the PRED for the f-structure labeled f is 'DAVID', and the value of the PRED feature of g is 'YAWN⟨SUBJ⟩'. The single quotes surrounding a semantic form indicate that its value is *unique*: for example, each instance of use of the word *David* gives rise to a uniquely instantiated occurrence of the semantic form 'DAVID'.

We use English names for semantic forms throughout. For example, we provide the semantic form 'MAN' for the Warlpiri noun *wati* 'man'. This is done for ease of readability and to emphasize the distinction between the semantic form associated with a word and its surface realization; uniform use of Warlpiri names instead of English ones for semantic forms would be equally satisfactory, though generally less clear for an English-speaking audience.

The list of grammatical functions mentioned in a semantic form is called the *argument list.* We discuss its role in determining wellformedness constraints on f-structures in Section 3.6 of this chapter.

3.3. Attributes with Common Values

It is possible for two different attributes of the same f-structure to have the same value. When the value is an *atom* like SG or MASC, rather than an f-structure, we simply repeat the value each time:

(63)
$$\begin{bmatrix} \text{ATTRIBUTE1} & \text{V} \\ \text{ATTRIBUTE2} & \text{V} \end{bmatrix}$$

It is also possible for two different attributes to have the same f-structure as their value. Here the situation is slightly more complex. Recall that an f-structure is a set of pairs of attributes and values: f-structures, like other sets, obey the *Axiom of Extension*, which states that two sets are the same if and only if they have the same members (Partee et al. 1993, section 8.5.8). Thus, two f-structures are indistinguishable if they contain the same attribute-value pairs.[12]

Notationally, it is in some cases clearer to represent two identical f-structures separately, repeating the same f-structure as the value of the two attributes:

(64)
$$\begin{bmatrix} \text{ATTRIBUTE1} & \begin{bmatrix} \text{A1} & \text{V1} \\ \text{A2} & \text{V2} \end{bmatrix} \\ \\ \text{ATTRIBUTE2} & \begin{bmatrix} \text{A1} & \text{V1} \\ \text{A2} & \text{V2} \end{bmatrix} \end{bmatrix}$$

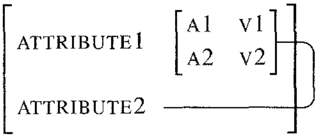
Care must be taken if a *semantic form*, the value of the attribute PRED, is repeated. Since each instance of a semantic form is unique, a repeated semantic form must be explicitly marked with an index to indicate identity; see Chapter 5, Section 2.2.1 for more discussion of this point. If no such index appears, the two semantic forms are assumed to be different.

In other cases, it may be easier and more perspicuous not to repeat the f-structure, but to use other notational means to indicate that the same f-structure

[12] This view of f-structures is different from the view of similar structures in HPSG (Pollard and Sag 1994); the attribute-value structures of HPSG are*graphs*, not set-theoretic objects. On the HPSG view, two attribute-value structures can contain the same attributes and values and can nevertheless be different structures. To state the same point in a different way: HPSG relies on a*type-token* distinction in attribute-value structures (Shieber 1986), meaning that two attribute-value structures are of the same *type* if they have the same set of attributes and values, but may be different*tokens* of that type. In the set-theoretic view of LFG, the Axiom of Extension precludes a type-token distinction, so two f-structures that have the same attributes and values are not distinguished.

appears as the value of two different attributes. We can accomplish this by drawing a line from one occurrence to another, a common practice in LFG literature; this notation conveys exactly the same information as in (64):

(65)



This convention is notationally equivalent to another way of representing the same structure:

(66)

$$\left[\begin{array}{ll} \text{ATTRIBUTE1} & \boxed{1}\left[\begin{array}{ll}\text{A1} & \text{V1}\\ \text{A2} & \text{V2}\end{array}\right] \\ \text{ATTRIBUTE2} & \boxed{1}\end{array}\right]$$

There is no substantive difference between these two conventions; following LFG tradition, we will generally represent identical values for two features by drawing a line connecting the two values, as in (65).

3.4. Sets

Sets are also valid structures, and may appear as values of attributes. Sets are often used to represent structures with an unbounded number of elements. For instance, there is in principle no limit to the number of modifiers that can appear with any phrase, and so the value of the ADJ feature is the set of all modifiers that appear:

(67) *David yawned quietly.*

$$\left[\begin{array}{ll} \text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ \text{TENSE} & \text{PAST}\\ \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`DAVID'}\\ \text{NUM} & \text{SG}\end{array}\right]\\ \text{ADJ} & \left\{\left[\begin{array}{ll}\text{PRED} & \text{`QUIETLY'}\end{array}\right]\right\}\end{array}\right]$$

In (67) only a single modifier appears, but other sentences may contain more modification:

(68) *David yawned quietly yesterday.*

$$\begin{bmatrix} \text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`DAVID'} \\ \text{NUM} & \text{SG} \end{bmatrix} \\ \text{ADJ} & \left\{ \begin{matrix} \begin{bmatrix} \text{PRED} & \text{`QUIETLY'} \end{bmatrix} \\ \begin{bmatrix} \text{PRED} & \text{`YESTERDAY'} \end{bmatrix} \end{matrix} \right\} \end{bmatrix}$$

Any valid structure can be an element of a set: for example, some sets can have atomic values as their elements. In Chapter 13, we will discuss how these can be used to represent the values of the PERS, GEND, and CASE features in a succinct treatment of feature resolution. The following f-structure for *We yawned* contains the fully specified value {S,H} (mnemonic for speaker and Hearer) of the PERS feature of the first person subject *we*:

(69) *We yawned.*

$$\begin{bmatrix} \text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \\ \text{PERS} & \{\text{S,H}\} \\ \text{NUM} & \text{PL} \end{bmatrix} \end{bmatrix}$$

3.5. Sets With Additional Properties

Since there is no limit to the number of conjuncts in a coordinate structure, we also use sets in their representation. Sets representing coordinate structures are special in that the set as a whole is a *hybrid object* that can have its own attributes and values as well as having elements; we will discuss this further in Chapter 13.

As shown above, we represent sets in curly brackets that contain the element f-structures. If a set has additional attributes, we enclose the set in square brackets and list the attributes and values of the set within the square brackets. For example, if a set f has the attribute a with value v it looks like this:

(70)
$$f\begin{bmatrix} a & v \\ \{ & \} \end{bmatrix}$$

In the following example, the conjuncts of the coordinate subject *David and Chris* are each singular, but the coordinate structure as a whole is a plural phrase. Thus, the set bears the feature NUM with value PL:

(71) *David and Chris yawn.*

$$\left[\begin{array}{ll} \text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ \text{TENSE} & \text{PRES} \\ \text{SUBJ} & \left[\begin{array}{l} \text{NUM} \quad \text{PL} \\ \left\{\begin{array}{l} \left[\begin{array}{ll} \text{PRED} & \text{`DAVID'} \\ \text{NUM} & \text{SG} \end{array}\right] \\ \left[\begin{array}{ll} \text{PRED} & \text{`CHRIS'} \\ \text{NUM} & \text{SG} \end{array}\right] \end{array}\right\} \end{array}\right] \end{array}\right]$$

3.6. Wellformedness Conditions on F-Structures

F-structures are required to meet certain wellformedness conditions: Completeness, Coherence, and Consistency (Kaplan and Bresnan 1982). The Completeness and Coherence conditions ensure that all the arguments of a predicate are present and that there are no additional arguments that the predicate does not require. The Consistency condition ensures that each attribute of an f-structure has a single value. We also discuss these requirements in Chapter 5, Section 2.2.

3.6.1. Completeness

The Completeness requirement tells us what is wrong with a sentence like:

(72) **David devoured.*

Intuitively, some required material is missing from a sentence that is incomplete. The required material is specified as a part of the value of the PRED feature, the semantic form. The PRED and semantic form for a verb like *devoured* are:

(73) $\left[\begin{array}{ll} \text{PRED} & \text{`DEVOUR}\langle\text{SUBJ,OBJ}\rangle\text{'} \end{array}\right]$

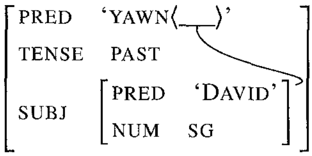
The *argument list* of a semantic form is a list of *governable grammatical functions*[13] that are *governed*, or mentioned, by the predicate: in example (73), *devour* governs the grammatical functions SUBJ and OBJ. Example (72) contains a SUBJ but no OBJ; this accounts for its unacceptability according to the Completeness requirement.

Previous LFG literature has contained a variety of notations for the argument list. In the notation employed here, the argument list consists of a list of names

[13] Recall from Section 1.2 of this chapter that the governable grammatical functions are:
SUBJ OBJ OBJ$_\theta$ XCOMP COMP OBL$_\theta$

of governable grammatical functions. In other work, the argument list is sometimes depicted as a list of f-structures which are the values of the subcategorized functions:

(74)



It is also common for the argument list to be represented in the following way, where (↑ SUBJ) represents the subject f-structure, as explained in Chapter 5, Section 3.1:

(75)

$$\begin{bmatrix} \text{PRED} & \text{`YAWN}\langle(\uparrow\ \text{SUBJ})\rangle\text{'} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`DAVID'} \\ \text{NUM} & \text{SG}\end{bmatrix}\end{bmatrix}$$

These notational variants are equivalent, though technically the variant shown in (75) is incorrect, since it contains the uninstantiated f-structure metavariable ↑; here, we choose the more succinct representation in (62) to save space and make the f-structures more readable.

There is a difference between grammatical functions that appear inside the angled brackets and those that appear outside. In (73), the functions SUBJ and OBJ appear inside the brackets. This indicates that the SUBJ and OBJ are semantic as well as syntactic arguments of *devour*, contributing to its meaning as well as filling syntactic requirements. In contrast, the semantically empty subject *it* of a verb like *rain* makes no semantic contribution; thus, the SUBJ function appears outside the angled brackets of the argument list of the semantic form of *rain*:

(76) *It rained.*

$$\begin{bmatrix} \text{PRED} & \text{`RAIN}\langle\rangle\text{SUBJ'} \end{bmatrix}$$

Similarly, the SUBJ argument of the verb *seem* is not a semantic argument of that verb and appears outside the angled brackets:

(77) *It seemed to rain.*

$$\begin{bmatrix} \text{PRED} & \text{`SEEM}\langle\text{XCOMP}\rangle\text{SUBJ'} \end{bmatrix}$$

This intuitive difference is reflected in the formal requirement that arguments of a predicate that appear inside angled brackets must contain a PRED attribute whose value is a semantic form; this is not required for arguments outside angled brackets.

Following Kaplan and Bresnan (1982), the Completeness requirement can be formally defined as follows:

(78) Completeness:

> An f-structure is *locally complete* if and only if it contains all the governable grammatical functions that its predicate governs. An f-structure is *complete* if and only if it and all its subsidiary f-structures are locally complete.

Chapter 9 will provide further discussion of the role of the PRED feature in ensuring syntactic wellformedness and its place in the theory of the syntax-semantics interface.

3.6.2. Coherence

The Coherence requirement disallows f-structures with extra governable grammatical functions that are not contained in the argument list of their semantic form:

(79) **David yawned the sink.*

The f-structure for this sentence is ill-formed:

(80) Ill-formed f-structure:

$$\begin{bmatrix} \textsc{pred} & \text{`}\textsc{yawn}\langle\textsc{subj}\rangle\text{'} \\ \textsc{subj} & \begin{bmatrix}\textsc{pred} & \text{`}\textsc{David}\text{'}\end{bmatrix} \\ \textsc{obj} & \begin{bmatrix}\textsc{pred} & \text{`}\textsc{sink}\text{'}\end{bmatrix} \end{bmatrix}$$

The governable grammatical function OBJ is present in this f-structure, though it is not governed by the semantic form of *yawn*. Consequently, the f-structure is incoherent.

Of course, the Coherence requirement applies only to *governable* grammatical functions, not functions that are ungoverned, such as modifying adjuncts. The following f-structure is perfectly coherent. Besides the single governable function SUBJ, it contains a modifier ADJ, which is not a governable function:

(81) *David yawned yesterday.*

$$\begin{bmatrix} \textsc{pred} & \text{`}\textsc{yawn}\langle\textsc{subj}\rangle\text{'} \\ \textsc{subj} & \begin{bmatrix}\textsc{pred} & \text{`}\textsc{David}\text{'}\end{bmatrix} \\ \textsc{adj} & \left\{ \begin{bmatrix}\textsc{pred} & \text{`}\textsc{yesterday}\text{'}\end{bmatrix} \right\} \end{bmatrix}$$

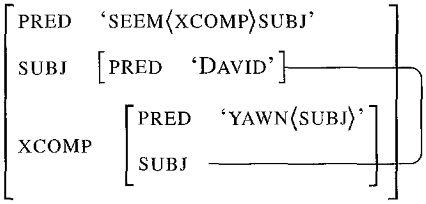
Coherence requires every f-structure bearing a governable grammatical function to be governed by some predicate: that is, every governable grammatical function that is present in an f-structure must be mentioned in the argument list of the PRED of that f-structure. The following f-structure is incoherent, since there is no PRED in the larger f-structure whose argument list contains OBJ:

(82) Ill-formed f-structure:

$$\left[\begin{array}{ll}\text{OBJ} & \left[\begin{array}{ll}\text{PRED} & \text{‘DAVID’}\end{array}\right]\end{array}\right]$$

Note that it is not a violation of any condition for more than one predicate to govern an f-structure with a semantic form. In fact, this is a common situation with "raising" verbs like *seem*, whose subject is also the subject of its XCOMP argument (see Chapter 12):[14]

(83) *David seemed to yawn.*



The line connecting the f-structure for *David* to the SUBJ position of *seem* indicates that the same f-structure is the value of two different attributes: it is both the SUBJ of *seem* and the SUBJ of *yawn*. Coherence is satisfied for both predicates: each requires a SUBJ, and this requirement is satisfied for each verb.

It is usual but not necessary for the argument list of a predicate to mention grammatical functions, expressions of length one, and not lists of functions or paths through the f-structure. In some treatments of subcategorized oblique phrases, however, the argument list of a predicate contains expressions, such as OBL_{ON} OBJ, of length greater than one; see, for example, Levin (1982) and Falk (2001):

(84) *David relied on Chris.*

$$\left[\begin{array}{ll}\text{PRED} & \text{‘RELY}\langle\text{SUBJ,OBL}_{\text{ON}}\ \text{OBJ}\rangle\text{’}\\ \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{‘DAVID’}\end{array}\right]\\ \text{OBL}_{\text{ON}} & \left[\begin{array}{ll}\text{OBJ} & \left[\begin{array}{ll}\text{PRED} & \text{‘CHRIS’}\end{array}\right]\end{array}\right]\end{array}\right]$$

This f-structure is coherent because the governable grammatical functions it contains are mentioned in the argument list of *rely*. That is, ‘RELY⟨SUBJ,OBL_{ON} OBJ⟩’

[14] Since the subject of *seem* is a syntactic but not a semantic argument of the *seem* predicate, the SUBJ in the value of the PRED attribute of *seem* appears outside the angled brackets, as explained in Section 3.6.1 of this chapter.

governs the OBJ of the oblique function OBL$_{\text{ON}}$. We do not adopt this treatment of oblique phrases here, but merely display an example to illustrate this possibility.

The Coherence requirement can be formally defined as follows (Kaplan and Bresnan 1982):

(85) Coherence:

An f-structure is *locally coherent* if and only if all the governable grammatical functions that it contains are governed by a local predicate. An f-structure is *coherent* if and only if it and all its subsidiary f-structures are locally coherent.

3.6.3. CONSISTENCY

The Consistency requirement, or Uniqueness Condition, reflects the functional (as opposed to relational) nature of the f-structure. An attribute of an f-structure may have only one value, not more (Kaplan and Bresnan 1982):

(86) Consistency:

In a given f-structure a particular attribute may have at most one value.

This requirement disallows f-structures satisfying incompatible constraints:

(87) **The boys yawns.*

Ill-formed f-structure:

$$\begin{bmatrix} \text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`BOYS'} \\ \text{NUM} & \text{SG/PL} \end{bmatrix} \end{bmatrix}$$

The SUBJ noun phrase *the boys* is a plural phrase, but the verb *yawns* requires its subject to be singular. These two requirements cannot be simultaneously met: the value of the attribute NUM must be either SG or PL, and it cannot have both values at the same time.

4. THE AUTONOMY OF FUNCTIONAL ORGANIZATION

LFG does not assume that abstract grammatical functions are defined in terms of their phrase structural position in the sentence or in terms of morphological properties like casemarking; instead, they are primitive concepts of the theory. However, there is also clear evidence for structure at other levels: for example, there is abundant evidence for morphological and phrasal organization and structure. Given this, one might conclude that constituent structure is the only structure with a firm linguistic basis, and that the appearance of abstract grammatical functions is actually only an illusion. On this view, the generalizations that traditional grammarians assumed are actually derivative of phrasal organization and structure. We will see in the following that this view is misguided: attempts to define functional structure in terms of morphological or phrase structure concepts do not succeed.

4.1. Grammatical Functions Defined?: Casemarking

It is clear that arguments of predicates have certain superficial morphological properties, and it is equally clear that it is not possible to define grammatical functions in terms of these properties. A cursory look at languages with complex casemarking systems is enough to show that the relation between case and grammatical function is not at all straightforward.

Examples given in Section 1.6 of this chapter show that it is possible to demonstrate a correlation between grammatical function and casemarking in Malayalam: if an argument is ACC, it is an object. However, the overall picture is much more complex; Mohanan (1982) argues convincingly against defining grammatical functions in terms of superficial properties like case. Objects in Malayalam are marked ACC if animate, NOM if inanimate:

(88) a. Nominative subject, object:

kuṭṭi waat̲il aṭaccu
child.NOM door.NOM closed
'The child closed the door.'

b. Nominative subject, accusative object:

kuṭṭi aanaye kaṇṭu
child.NOM elephant.ACC saw
'The child saw the elephant.'

In Malayalam, then, there is clearly no one-to-one relation between casemarking and grammatical function, since a grammatical function like OBJ is marked with one of a variety of cases.

Similarly, arguments that can be shown to bear the SUBJ function in Icelandic are marked with a variety of cases, as shown by Andrews (1982). These cases also appear on arguments filling nonsubject grammatical functions; for instance, as examples (89a) and (89b) show, ACC case can appear on both subjects and objects, and examples (89c) and (89d) show that subjects can bear other cases as well:

(89) a. Accusative subject:

Hana dreymdi um hafið.
She.ACC dreamed about sea.DEF
'She dreamed about the sea.'

b. Accusative object:

Stúlkan kyssti drengina.
girl.NOM kissed boys.ACC
'The girl kissed the boys.'

c. Dative subject:

Bátnum hvolfdi.
boat.DEF.DAT capsized
'The boat capsized.'

d. Genitive subject:

Verkjanna gætir ekki.
pains.DEF.GEN is.noticeable not
'The pains are not noticeable.'

In sum, the relation between grammatical function and case is complex. Even when there is a close relation between case and grammatical function, as discussed in Section 1.6 of this chapter, a clear and explanatory description of case-marking and other morphosyntactic properties is best obtained by reference to abstract functional properties.

4.2. Grammatical Functions Defined?: Constituent Structure

Another visible, easily testable property of languages is their surface phrase structure. Given the necessity for this structure, one might claim that grammatical functions are not universally manifest, but instead that the appearance of grammatical functions in a language like English is due to the fact that grammatical functions are associated with certain phrasal configurations in English syntax: in a nutshell, English has subjects and objects because English is configurational. This claim entails that nonconfigurational languages would not be expected to exhibit the same abstract functional relations.

Warlpiri is a language whose phrasal syntactic structure is completely different from that of languages like English. Warlpiri (like many Australian languages) is known for displaying "nonconfigurational" properties, including free word order and "discontinuous phrases." The following Warlpiri sentences involve permutations of the same words; they are all grammatical and have more or less the same meaning (Hale 1983, page 6):

(90) a. *ngarrka-ngku ka wawirri panti-rni*
man-ERG AUX kangaroo spear-NONPAST
'The man is spearing the kangaroo.'

b. *wawirri ka panti-rni ngarrka-ngku*
kangaroo AUX spear-NONPAST man-ERG

c. *panti-rni ka ngarrka-ngku wawirri*
spear-NONPAST AUX man-ERG kangaroo

It would be difficult to find a language less like English in its phrase structure configurational properties. Thus, Warlpiri would seem to be a good candidate to test the hypothesis that evidence for grammatical functions can be found only in English-like configurational languages.

However, as Hale (1983) demonstrates, languages like Warlpiri do show evidence of abstract grammatical relations, just as English-like configurational languages do. Hale discusses person marking, control, and interpretation of reflexive/reciprocal constructions, showing that constraints on these constructions are not statable in terms of surface configurational properties. Simpson and Bresnan (1983) and Simpson (1991) provide further evidence that properties like control in Warlpiri are best stated in terms of abstract functional syntactic relations. In particular, Simpson and Bresnan (1983) examine the *karra*-construction, in which the subject of a subordinate clause with subordinate clause affix *karra* is controlled by the subject of the matrix clause:

(91) *ngarrka ka wirnpirli-mi [karli jarnti-rninja-karra]*
man.ABS AUX whistle-NONPAST boomerang.ABS trim-INF-COMP
'The man is whistling while trimming a boomerang.'

As Simpson and Bresnan show, the controller subject may be discontinuous or absent, and it may be marked with NOM, ABS, or ERG case. The correct generalization about this construction involves the abstract grammatical function SUBJ of the controller, not any of its surface configurational properties.

Thus, even in a language that appears to have completely different phrase structure properties from English, and which has been analyzed as "nonconfigurational," evidence for abstract functional syntactic relations is still found. The hypothesis that functional structure is epiphenomenal of surface configurational properties is not viable. This position is fairly widely accepted, although proposals for the representation of abstract syntactic structure are more variable; in Chapter 4, Section 5, we will discuss proposals by Hale and others for representing abstract syntactic structure, relations like SUBJ and OBJ, by means of phrase structure trees.

4.3. Grammatical Functions Defined?: Semantic Composition

Dowty (1982) proposes to define grammatical functions like SUBJ and OBJ in compositional semantic terms, by reference to order of combination of a predicate with its arguments: a predicate must combine with its arguments according to a functional obliqueness hierarchy, with the SUBJ defined as the last argument to combine with the predicate. This approach is also adopted by Gazdar et al. (1985) for Generalized Phrase Structure Grammar, and has to some extent carried over to Head-Driven Phrase Structure Grammar (Pollard and Sag 1994).

There are several ways in which an approach like Dowty's, where grammatical functions are defined as an ordering on the arguments of a predicate, might lead to incorrect predictions. First, if the order of semantic composition is very closely tied to the order of composition of the surface configurational structure, this approach would predict that the subject could not intervene between the verb and the object; of course, this prediction is not correct, since many languages exhibit VSO word order. The theory that Dowty and most other adherents of this position advocate does not suffer from this difficulty, however, since the hypothesized relation between the surface order of arguments in a sentence and the order of semantic composition is more complex.

A more subtle problem does arise, however. It is not clear that such an approach can make certain distinctions that are necessary for syntactic analysis: in particular, it does not seem possible to distinguish between predicates that take the same number of arguments with the same phrasal categories. For example, any two-argument verb that requires a noun phrase subject and a sentential complement should behave like any other such verb.

However, there are languages in which some sentential complements bear the OBJ function, while others bear the COMP function, as discussed in Section 1.7 of this chapter. In a theory like LFG, this distinction is reflected in a difference in the grammatical function of the sentential complement; some sentential complements are OBJ, and others are COMP. It is not clear how such a distinction can be drawn in a theory in which grammatical functions are defined purely by order of combination with the verb.

Dowty also argues against theories which, like LFG, assume that grammatical functions are undefined primitives by claiming that in his approach "grammatical relations play an important role in the way syntax relates to compositional semantics." This statement is a non sequitur. In LFG, grammatical functions are primitive concepts and also play an important role in compositional semantics and the syntax-semantics interface. Indeed, this is the topic of Chapter 9 and the following chapters (see also Bresnan 1982a, page 286).

Leaving aside these difficulties, there is a strong degree of similarity between theories that define grammatical functions in terms of abstract properties such as order of semantic combination and theories like LFG, in which grammatical functions are not definable in terms of phrasal or argument structure. For both types of theories, grammatical functions are abstract and are analyzed independently from phrasal and other structures.

5. FURTHER READING AND RELATED ISSUES

Within LFG, there has been more discussion of grammatical functions and functional structure than can be summarized in a brief space. Besides the works cited earlier, Andrews (1985) provides a good overview of the grammatical functions of nominals. Butt et al. (1999) provide a general overview of English, French, and German functional and phrasal structure; in particular, they discuss constructions involving the open complement XCOMP and propose a new grammatical function, PREDLINK, for closed nonverbal complements. Falk (2000) also discusses the theory of grammatical functions, proposing the addition of a new grammatical function PIVOT. We will discuss other work on functional structure and its relation to other linguistic structures in the following chapters.

3

CONSTITUENT STRUCTURE

We have seen that there is a large degree of unity in the abstract functional syntactic structure of languages. In contrast, the phrasal structure of language varies greatly: some languages allow phrases with no lexical heads, and some have no such categories; some languages have a VP constituent, and others do not; and so on. In this chapter, we will discuss the organization of overt surface phrasal syntactic representation, the *constituent structure* or *c-structure*. We will explore commonalities in constituent structure across the world's languages, and also talk about how languages can differ in their phrasal organization.

Section 1 of this chapter begins by discussing some traditional arguments for constituent structure representation. Many of these arguments turn out to be flawed, since the theory of phrase structure has a different status in LFG than in theories in which grammatical functions are defined configurationally and abstract syntactic relations are represented in phrase structure terms. Many arguments that have been made for particular phrase structure theories and configurations are based on phenomena which in LFG are better treated in f-structure terms, and do not constitute good arguments for constituent structure at all. For this reason, we must examine the status of arguments for and against particular phrase structures or phrase structure theories particularly carefully. Section 2 proposes some valid criteria within LFG for phrase structure determination.

Research on constituent structure has revealed much about the universally available set of categories and how they can combine into phrases. We adopt the view of phrase structure that is most widely accepted in current LFG work, incorporating insights primarily from the work of Kroeger (1993), King (1995), and Bresnan (2001b) (though our proposal differs slightly from these treatments); for similar proposals, see Sadler (1997) and Sells (1998). Section 3 examines the inventory of constituent structure categories that are crosslinguistically available: we assume lexical categories like N and V, as well as functional categories like I and C. These categories appear as the heads of phrases like NP and IP; the theory of the organization of words and categories into phrases is outlined in Section 4. The general theory of constituent structure organization is exemplified in Section 5, where we provide more specific discussion of the constituent structure organization of clauses and the role of the functional categories I and C in clausal structure.

1. TRADITIONAL ARGUMENTS FOR CONSTITUENT STRUCTURE

The sorts of constituent structure rules and representations used in LFG and most other theories of constituent structure are prefigured in work by Bloomfield (1933), who assumes complex phrasal structures described in terms of "immediate constituent analysis": the combination of words into phrases. These structures were originally motivated within the transformational tradition by the desire to formulate a finite characterization of the infinite set of sentences of a language. As Chomsky (1955, page 116) says:

> If there were no intervening representations between *Sentence* and words, the grammar would have to contain a vast (in fact, infinite) number of conversions of the form *Sentence*→*X*, where *X* is a permissible string of words. However, we find that it is possible to classify strings of words into phrases in such a way that sentence structure can be stated in terms of phrase strings, and phrase structure in terms of word strings, in a rather simple way. Further, a phrase of a given type can be included within a phrase of the same type, so that a finite number of conversions will generate an infinite number of strings of words.

Chomsky's first point is that the same sequence of categories may appear in more than one environment, and any adequate grammar must characterize this regularity. For instance, the phrase *the dachshund* can appear in many positions in a sentence:

(1) a. *The dachshund is barking.*
    b. *David petted the dachshund.*
    c. *Matty took the dachshund to the vet.*

No matter where it appears, it can be replaced by a phrase with additional modifiers:

(2) a. *The black dachshund is barking.*
    b. *David petted the black dachshund.*
    c. *Matty took the black dachshund to the vet.*

Generalizations about the structure of the phrase *the dachshund* and the phrases that can be substituted for it are captured by assuming that *the dachshund* is a noun phrase and can appear wherever other noun phrases can appear. This is intuitively appealing, since it captures the intuition that *the dachshund* (and *the black dachshund*) are phrasal units in a sense that *dachshund is* or *petted the* are not. Further, it simplifies the linguistic description: the different ways in which a noun phrase can be formed do not have to be separately enumerated for each environment in which a noun phrase can appear.

Chomsky's second point is that a phrase may contain subconstituents of the same type: a clause can contain a subordinate clause, a noun phrase can contain other noun phrases, and so on. Our linguistic descriptions therefore need the same kind of recursive character.

This formal motivation for a level of constituent structure analysis and representation is buttressed by a range of diagnostics for phrase structure consistency. However, current syntactic theories vary greatly in their criteria for determining the validity of these diagnostics, and tests that are accepted in some theoretical frameworks are not recognized as valid in other frameworks. For instance, Chomsky (1981) and work growing out of the transformationally based theory presented there (see, for example, Webelhuth 1995) propose that abstract syntactic properties and relations are represented in terms of phrase structure trees and that the relation between these trees is statable in terms of the movement of constituents ("move-α"). Given such a theory, the criterion for constituenthood is fairly straightforward: any unit that plays a role in abstract syntactic structure is a phrase structure constituent, since it must be represented as a syntactic unit and must be eligible to undergo the rule of move-α.

Such arguments have no force in a theory like LFG. We have seen that abstract syntactic structure is represented by functional structure, not in phrasal terms; further, phrase structural transformations play no role in LFG theory. LFG's constituent structure trees represent tangible phrasal configurations, not more abstract functional syntactic relations. Thus, many criteria commonly proposed within the transformational tradition to identify phrase structure constituents turn out to be irrelevant.

In this light, it is interesting to examine carefully one particularly clear and explicit proposal for determining constituency within a transformationally based theory. Radford (1981, page 69) gives the following set of criteria for constituency:

(3) A given string of elements is a *constituent* just in case it has one or more of the following properties:

(i) It behaves distributionally as a single structural unit — i.e. it recurs as a single unit in a variety of other sentence positions.

(ii) It can be coordinated with another similar string.

(iii) It does not readily permit intrusion of parenthetical elements internally (intrusion generally being permitted only at the boundaries of major — especially phrasal — constituents).

(iv) It can be replaced by, or serve as the antecedent of, a proform.

(v) It can be omitted, under appropriate discourse conditions.

Test (i), the distributional test, is a useful, widely proposed criterion, as originally noted by Chomsky and discussed earlier. Test (iii) deals particularly with the distribution of one type of phrase, parentheticals. This test is also useful; in many languages, it is generally possible to insert parenthetical elements only at a boundary between major phrases:

(4) a. *[Chris], it seems, [wrote a bestselling novel].*

b. **Chris wrote a, it seems, bestselling novel.*

Test (ii), coordination, does not prove to be as successful. It is well known that many strings that are not shown to be constituents by other criteria can be coordinated:

(5) a. *David gave [a flower] [to Chris] and [a book] [to Pat].*

b. *[David] [likes], and [Chris] [dislikes], carrots.*

On most theories of the constituent structure of English, the phrases *a flower to Chris* and *David likes* are not constituents. It does seem to be true, at least in English, that if a string is a constituent it can be coordinated, barring semantic unacceptability. See Chapter 13 for discussion of coordination in general and nonconstituent coordination in particular.

Tests (iv) and (v) are also relatively unsuccessful. Many constituents cannot be replaced by a proform:

(6) a. *Chris gave a book [to David].*

b. **Chris gave a book him/thus/there.*

Many constituents cannot be omitted:

(7) a. *Chris told me that David yawned.*

b. **Chris told that David yawned.*

c. **Told me that David yawned.*

Conversely, some nonconstituents can be replaced by a proform. In example (8), the discontinuous phrase *two pies ... eat* is replaced by the proform *it*:

(8) a. *Chris can eat one pie, but two pies he won't be able to eat.*

b. *Yes, he will be able to do it.*

The tests proposed by Radford are, then, not uniformly successful, attesting to the difficulty of formulating unequivocal criteria for determining constituent structure units.

Nonsyntactic arguments are sometimes given to motivate particular theories of constituent structure: for example, constituent structure groupings have been claimed to be motivated by semantic facts. Andrews (1983b) claims that the relative scope of adverbial modifiers is reflected in their phrasal structure:

(9) a. *John [[[knocked on the door] intentionally] twice].*

b. *John [[[knocked on the door] twice] intentionally].*

Examples (9a) and (9b) have different meanings: example (9a) means that the action of knocking on the door intentionally took place twice, while example (9b) means that there was an action of knocking on the door twice that was performed intentionally. Andrews proposes that this fact should be reflected in the constituent structure of the examples, as indicated by the bracketing.

However, the theory of modification, scoping, and the syntax-semantics interface to be presented in Chapter 9 relies on a different explanation of these facts, one which is defined in terms of functional relations rather than phrase structure grouping. We do not consider semantic scoping facts like these to constitute evidence for phrase structure constituency.

2. EVIDENCE FOR CONSTITUENT STRUCTURE

Criteria for phrase structure constituency in LFG depend on the surface syntactic properties of utterances, not on semantic intuitions or facts about abstract functional syntactic structure. Since these surface properties vary from language to language, the tests discussed below make reference to properties of particular languages, and may not be applicable to every language. This is just what we would expect, since surface form and organization vary from language to language, while the more abstract functional structure is more uniform.

INTONATION: A particularly interesting sort of evidence for phrase structure constituency arises from the interaction of constituent structure and intonation. King (1995, pages 129–130) notes that a Russian constituent may be signaled as in focus by falling intonation on the right edge of the focused constituent; conversely, then, focus intonation may be viewed as indicating a right-edge phrasal boundary. Here, the feature +F represents this right-edge focus intonation:[1]

(10) *kolxoz zakončil [uborku urožaja*$_{+\text{F}}$*].*
kolxoz finished harvest crop
'The *kolxoz* finished [the crop harvest]-FOCUS.'

An entire sentence can be put in focus in this way (Junghanns and Zybatow 1997):

(11) (What happened?)

[sgorela ratuša$_{+\text{F}}$*].*
burn.down town.hall
'The town hall burned down.'

See Zybatow and Mehlhorn (2000) for further discussion of focus intonation in Russian, and Zec and Inkelas (1990) for discussion of the relation between syntax and prosody.

CLITIC PLACEMENT: Zwicky (1990) proposes other "edge tests," tests that pick out the first or last word in a phrase. He argues that the distribution of the possessive clitic in an English possessive noun phrase is best described by referring to the right edge of the possessive phrase:

(12) *[my friend from Chicago]'s crazy ideas*

VERB-SECOND: Besides these tests, which pick out phrasal boundaries, there are some generalizations that refer specifically to phrase structure constituents. For instance, in some languages, the position of the verb provides a test for constituency: the verb must appear in second position in the sentence, after the first constituent. Simpson (1991) shows that Warlpiri is such a language; any sequence of words that precedes the auxiliary must be a constituent:

[1] King attributes example (10) to Krylova and Khavronina (1988, page 80).

(13) *[watiya-rlu wiri-ngki] ji paka-rnu*
stick-ERG big-ERG AUX hit-PAST
'He hit me with a big stick.'

This argument can be repeated for many other verb-second languages: Kroeger (1993) discusses verb-second phenomena in Germanic languages and in Tagalog.

QUESTION FORMATION: A similar argument for constituency comes from the English wh-question construction. Zwicky (1990) proposes that in English, only a single displaced constituent can appear in clause-initial position:

(14) a. *[Which people from California] did you introduce to Tracy?*
b. **[Which people from California] [to Tracy] did you introduce?*
c. *[To how many of your friends] did you introduce people from California?*
d. **[People from California] [to how many of your friends] did you introduce?*

DISTRIBUTION OF ADVERBS: It is also possible to determine the presence and distribution of phrases of a particular type. Kaplan and Zaenen (1989), following work by Thráinsson (1986), discuss the distribution of adverbs in Icelandic; Sells (1998) builds on their work, providing further discussion of Icelandic constituent structure. In an Icelandic sentence containing an auxiliary or modal verb such as *mun* 'will', an adverb like *sjaldan* 'seldom' has a restricted distribution:

(15) a. *Hann mun sjaldan stinga smjörinu í vasann.*
he will seldom put butter.DEF in pocket.DEF
'He will seldom put the butter in the pocket.'

b. * *Hann mun stinga sjaldan smjörinu í vasann.*
he will put seldom butter.DEF in pocket.DEF

c. * *Hann mun stinga smjörinu sjaldan í vasann.*
he will put butter.DEF seldom in pocket.DEF

d. *Hann mun stinga smjörinu í vasann sjaldan.*
he will put butter.DEF in pocket.DEF seldom

In sentences with no modal, the distribution of the adverb is free:

(16) a. *Hann stingur sjaldan smjörinu í vasann.*
he puts seldom butter.DEF in pocket.DEF
'He seldom puts the butter in the pocket.'

b. *Hann stingur smjörinu sjaldan í vasann.*
he puts butter.DEF seldom in pocket.DEF

c. *Hann stingur smjörinu í vasann sjaldan.*
he puts butter.DEF in pocket.DEF seldom

Kaplan and Zaenen (1989) propose that the distribution of the adverb depends on the presence or absence of a VP constituent: an adverb appears as a daughter of S, not of VP. Sentences with an auxiliary verb contain a VP constituent:

(17) *Hann mun [stinga smjörinu í vasann]*$_{\mathrm{VP}}$.
he will put butter.DEF in pocket.DEF
'He will put the butter in the pocket.'

The adverb can appear as a daughter of S, but not as a daughter of VP, accounting for the ungrammaticality of examples (15b,c). In contrast, sentences with no auxiliary verb have a flat structure with no VP, allowing for the wider range of possible adverb positions shown in (16).

3. CONSTITUENT STRUCTURE CATEGORIES

LFG does not postulate constituent structure positions filled only by affixes or (as Kroeger 1993 puts it) "disembodied morphological features." Instead, the leaves of the constituent structure tree are individual words filling a single constituent structure node; there is no syntactic process of word assembly, though individual words can make complex syntactic contributions at the functional level (Chapter 4, Section 4). In the following, we discuss the inventory of constituent structure categories.

3.1. Lexical Categories

We assume the following set of major lexical categories:

(18) Major lexical categories:

N(oun), P(reposition), V(erb), A(djective), Adv(erb)

Chomsky (1986) assumes that N, P, V, and A are major lexical categories; following Jackendoff (1977), we also assume that Adv is a major lexical category. These major lexical categories are heads of phrases of the corresponding category:

(19) a. NP: *the boy*

b. PP: *on the boat*

c. VP: *sail the boat*

d. AP: *very fearful of the storm*

e. AdvP: *quite fearfully*

We also assume a set of more crosslinguistically variable minor lexical categories that do not project full phrasal structure, such as Part, the category of English particles such as *up* in examples like (20):[2]

(20) *David called Chris [up]*$_{\text{Part}}$.

3.2. Functional Categories

Besides the lexical categories described in the previous section, we assume a set of "functional" phrase structure categories; the terminology is standard but somewhat confusing, given the existence of functional structure (with a different sense of the word "functional") as a separate syntactic level in LFG. As noted by Bresnan (2001b), functional phrase structure categories were originally proposed by researchers working in a lexicalist, nontransformational setting, and have since been widely adopted within both transformational and nontransformational theories of phrase structure (Kroeger 1993; King 1995; Bresnan 2001b).

In this work, we assume the functional categories C and I. Other functional categories have also been proposed in work within LFG, particularly in work on the structure of noun phrases, and we will discuss these briefly in Section 3.2.3 below. Börjars et al. (1999) provide further discussion of functional categories in LFG.

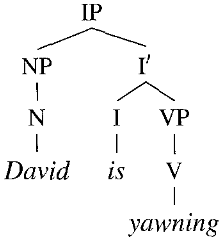
3.2.1. The Functional Category I

As Kroeger (1993) notes, the clausal syntax of a number of languages that are genetically unrelated and typologically quite different nevertheless organizes itself around a finite verbal element, either an auxiliary or a main verb, which appears in a specified (often second) position; languages as different as English, German, Warlpiri, and Tagalog all have a distinguished position in which a finite main or auxiliary verb appears. The position in which this element appears is called I (originally for INFL). The idea that the functional position I is the head of a finite clause was originally proposed by Falk (1984) in his LFG-based analysis of the English auxiliary system, and has since been incorporated into both transformational and nontransformational analyses of clausal structure (Chomsky 1986; Pollock 1989; Kroeger 1993; King 1995; Nordlinger 1998; Bresnan 2001b).

Languages can differ as to which lexical categories can fill the I position. In English, the tensed auxiliary appears in I, but nonauxiliary verbs may not appear there:

[2] This position is controversial, and future research may reveal advantages of a less idiosyncratic treatment of minor categories: for example, Part might better be analyzed as a distinctive subclass of a major category like P, as advocated by Zwicky (1985) and Toivonen (2001).

(21) *David is yawning.*



In other languages such as Russian (King 1995), all finite verbs appear in I, as shown in example (32).

3.2.2. The Functional Category C

As Bresnan (2001b) notes, the functional category C was first proposed as the head of CP by Fassi-Fehri (1981), following unpublished work on English auxiliary inversion by Ken Hale. Like the functional category I, the C position may be filled by a verbal element. Other elements may also appear in C position: in English, as shown in example (22) (page 55), the C position can be filled by a complementizer like *that*.[3]

In other languages, the rules may differ. King (1995, Chapter 10) provides a thorough discussion of I and C in Russian, and Sells (1998) discusses fillers of I in Icelandic and Swedish.

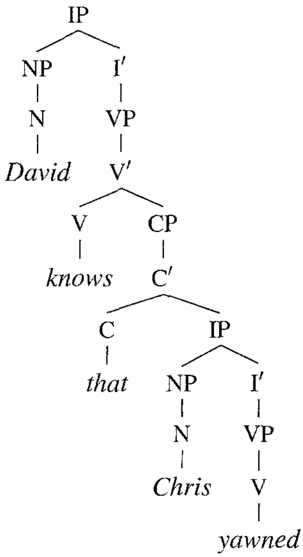
3.2.3. The Functional Categories D and K

In this work, we use the traditional category NP for noun phrases like *the boy*, and we will not make detailed proposals concerning the internal syntactic structure of noun phrases. Here we provide a brief overview of alternative approaches to the analysis of noun phrases.

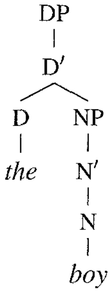
Following work by Brame (1982) and Abney (1987), some researchers have proposed that at least some instances of the category traditionally labeled "noun phrase" are more accurately treated as a determiner phrase or DP. According to this theory, the head of a phrase like *the boy* is the determiner *the* rather than the noun *boy*, as shown in example (23) (page 55).

[3] As example (22) shows, we do not assume that a phrase likeIP must dominate an I head; we will discuss phrase structure optionality in Section 4.4 of this chapter.

(22) *David knows that Chris yawned.*



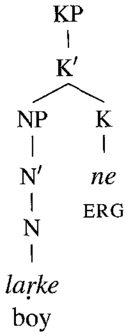
(23) *the boy*



Some recent work within LFG (for example, Bresnan 1997, 2001c) follows this assumption; Sadler (1997) presents an analysis of Welsh pronominal clitics as Ds as a part of a more complete DP analysis, and Börjars (1998) discusses the internal structure of noun phrases and the status of D in a range of languages.

The category KP for phrases headed by a case marker K was prefigured in the seminal work of Fillmore (1968); in an LFG setting, Butt and King (1999b) propose that Hindi-Urdu has the category KP (see also Davison 1998). Butt and King provide the following c-structure for the ergatively casemarked Hindi-Urdu phrase *laṛke ne* 'boy-ERG':

(24) *laṛke ne*
boy ERG



See Laczkó (1995) for an extensive discussion of the functional and phrasal syntax of noun phrases within the LFG framework.

4. CONSTITUENT STRUCTURE ORGANIZATION AND RELATIONS

4.1. X-Bar Theory

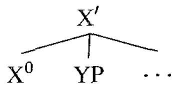
Constituent structure organization obeys the basic principles of *X-bar (X' or $\overline{X}$) theory* (Jackendoff 1977; Chomsky 1986): lexical items appear as heads of phrases and may be associated with *specifier* and *complement* positions within the same phrase. X' theory allows for general and abstract statements about the organization of phrases crosslinguistically and within a language.

Using X as a variable over lexical or functional categories like N, V, I, or C, the basic principle of X' theory is that a lexical or functional category X is related to *projections* of that phrase, often written as X' (with one "bar level"), X'' (with two "bar levels"), and so on. Equivalently, we can speak of a lexical or functional category X *projecting* the phrases X' and X''. For example, we can speak of V' as a projection of the lexical category V, or of V as projecting the category V'. A lexical category such as V is sometimes written with a superscript 0 to indicate that it has zero "bar levels," so that V^0 is an alternative way of representing the lexical category V. In the following, we will frequently omit the 0 superscript on lexical categories, using the term V instead of V^0 for the category of a verb.

A standard assumption of X' theory is that one of the projections of a category is a *maximal* phrase, and is thus usually written as XP. In other words, the category XP is the *maximal projection* of the category X. We adopt a simple two-level version of X' theory in which X'' is the a maximal phrase: thus, a phrase of category XP dominates a *nonmaximal* projection of category X' which, in turn, dominates a lexical item of category X^0, for any lexical or functional category X^0.

We further assume that a lexical item of category X^0 is sister to a series of *complement* and *adjunct* phrases (YP...) and forms a constituent of category X' whose phrasal head is X^0:[4]

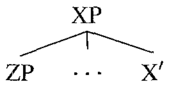
(25)



The X' node may dominate any number of daughter phrases; we do not assume that constituent structure trees must be binary branching.

The X' category is sister to a series of specifier phrases (ZP...) and forms an XP phrasal constituent with X' as its head:

(26)



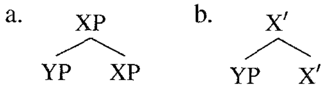
Some languages allow only a single specifier phrase; other languages (for example, languages like Russian in which multiple wh-constituents can appear in sentence-initial specifier position) allow multiple specifiers. To avoid notational clutter, we will not generally display the intermediate X' node unless it is necessary to explicate the point at hand.

The theory of the relation between constituent structure and functional structure is fairly well developed. As shown by Zaenen (1983), King (1995), Bresnan (2001b), and others, phrasal specifiers and complements are systematically related to certain f-structure categories. Chapter 4 is devoted to an explication of regularities in the mapping between c-structure and f-structure.

4.2. Adjunction

Besides these X'-theoretic structures, linguists have proposed other permissible phrase structure configurations. It is generally assumed that it is possible to adjoin a maximal phrase to either a maximal (XP) or a nonmaximal (X') projection, so-called "Chomsky-adjunction":

(27)

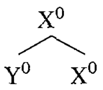


A phrase adjoined in this way is often an adjunct phrase (Bresnan 2001b).

[4] The configurations in (25) and (26) illustrate the general X'-theoretic configuration we assume; languages vary as to whether the head of the phrase precedes or follows the other daughters.

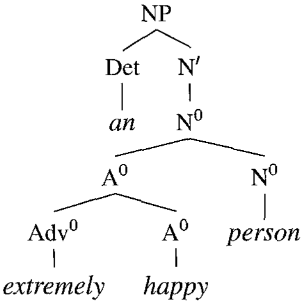
According to Sadler and Arnold (1994), lexical categories like A, which usually project full phrasal categories, can also appear as a "small" or X^0 category, adjoined to other X^0 categories. Sadler and Arnold call these adjoined structures "small" constructions, building on work by Poser (1992).[5] They argue that English prenominal adjectives participate in these constructions, producing structures like the following:

(28)



Sadler and Arnold (1994, page 214) propose the following structure for the phrase *an extremely happy person*:

(29) *an extremely happy person*



Sells (1994) also shows that these "small" constructions play an important role in the phrasal syntax of Korean.

4.3. Categorial Inventory

The inventory of phrasal categories is not universally fixed, but may vary from language to language; we do not assume that a phrasal category exists in a language unless there is direct evidence for it. Criteria for determining the presence of a phrasal category differ according to whether the phrase is a projection of a lexical or a functional category.

The existence of a projection of a lexical category in a language must in the first instance be motivated by the presence in the language of some word of that lexical category (King 1995). That is, for example, the existence of a VP phrase in the constituent structure of a language implies that there are lexical items of category V in that language. If a lexical category does not appear in a language, its corresponding phrasal projection does not appear either.

[5] A "small" construction in Sadler and Arnold's sense is different from a "small clause" (Stowell 1981), a tenseless phrase consisting of a subject and a predicate.

Even when there is evidence for a lexical category in a language, the corresponding phrasal category may in some cases not appear. For instance, although some Warlpiri words are of category V, Simpson (1991) shows that there is strong evidence against the existence of VP in Warlpiri. As discussed in Section 2 of this chapter, the material preceding the Warlpiri auxiliary must form a single constituent:

(30) *[watiya-rlu wiri-ngki] ji paka-rnu*
stick-ERG big-ERG AUX hit-PAST
'He hit me with a big stick.'

If the V formed a VP constituent with the object phrase, we would expect example (31) to be grammatical; in fact, however, it is not grammatical:

(31) **[wawirri] [panti-rni] ka ngarrka-ngku*
kangaroo.ABS spear-NONPAST AUX man-ERG
'The man speared the kangaroo.'

Simpson (1991) presents a number of other arguments to show that Warlpiri has no VP, although it does have a V$'$ constituent.

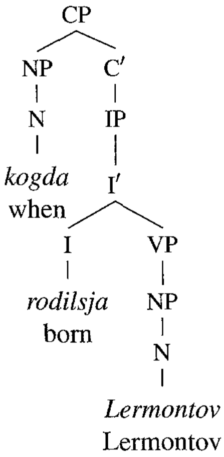
In the case of functional categories, evidence is often less directly available; here too, however, categories are only presumed to exist if they are motivated by direct evidence. For instance, Austin and Bresnan (1996) show that there is no evidence in Warlpiri for the existence of the category CP. One might expect the Warlpiri finite complementizer *kuja-* to appear as the head of a CP phrase. Actually, though, it appears in I, the second position in the clause, in the same place where the auxiliary appears. Austin and Bresnan conclude that *kuja-* is a special auxiliary base, like other elements in I^0 position, which does not appear in a C position or project a CP phrase. Further, the question word in a Warlpiri complement clause need not appear in initial position (that is, in the specifier position of a CP phrase). From this, Austin and Bresnan conclude that there is no CP in Warlpiri.

4.4. Optionality of Constituent Structure Positions

In the theory of constituent structure proposed by Chomsky (1986), heads of phrases are obligatorily present, complements (sisters to X^0 categories) are present according to predicate valence, and specifier positions (sisters to X$'$ categories) are optional. In contrast to this view, we assume with Kroeger (1993), King (1995), and Bresnan (2001b) that all constituent structure positions are optional. As discussed in Chapter 2, subcategorization requirements are most appropriately specified at the level of f-structure, and so there is no necessity for predicate valence to be reflected in c-structure representation. And since heads can appear outside their phrases (cases of so-called "head movement": see Zaenen and Kaplan 1995, King 1995, Nordlinger 1998), the position of the head must also be optional.

The following Russian example illustrates the optionality of a number of constituents (King 1995, page 172):

(32) *kogda rodilsja Lermontov?*
when born Lermontov
'When was Lermontov born?'



As the tree in (32) illustrates, there is no specifier position of IP if there is no topicalized or focused non-WH constituent (King 1995, page 172). Additionally, the tree illustrates "headless" constructions; the VP constituent does not dominate a V node, since the tensed verb in Russian appears in I. For more discussion of endocentric phrasal categories that do not contain a lexical head, see Zaenen and Kaplan (1995).

5. CLAUSAL ORGANIZATION

5.1. IP and CP

In many languages, IP corresponds to a sentence, and CP corresponds to what has sometimes been called S′, a sentence with a complementizer or a displaced phrase in sentence-initial position. Here we examine some basic structures in English and draw some contrasts with constituent structures in other languages. Of course, no conclusions for the structure of other languages should be drawn from the organization of English: LFG does not claim that the phrasal organization of every language is the same. Since there is a great deal of crosslinguistic typological variation in constituent structure organization, what is true for English cannot be assumed to be true in other languages. We must carefully examine each language on its own terms to determine its phrasal structure and organization.

5.1.1. The IP Phrase

In English, the tensed auxiliary verb appears in I, and the rest of the verb complex appears inside the VP, as shown in example (21). Evidence that the English VP forms a constituent comes from the fact that, like other focused constituents, it can be preposed:

(33) a. *David wanted to win the prize, and [win the prize] he will.*

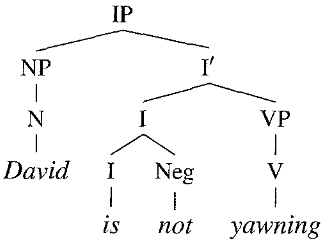
b. *... there is no greater deed than to die for Iran. And [dying] they are, ...* (from Ward 1990)

It is not possible to prepose only the auxiliary and verb, since they do not form a constituent:

(34) **David wanted to win the prize, and [will win] he the prize.*

Since the sentential negation morpheme *not* must be preceded by a tensed auxiliary verb, we assume that it is adjoined to the tensed verb in I:

(35) *David is not yawning.*

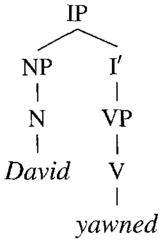


It is not possible in English for the negation marker to follow a nonauxiliary verb (in contrast with French; see Pollock 1989):

(36) **David yawned not.*

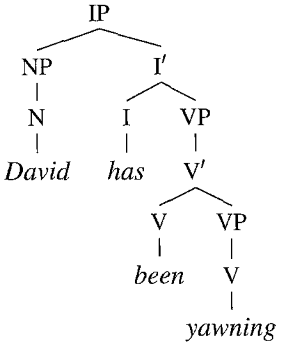
Thus, in English only tensed auxiliary verbs can appear in I, and main verbs, whether tensed or untensed, appear inside the verb phrase. When there is no auxiliary verb, the I position is empty:

(37) *David yawned.*



Nontensed auxiliaries appear in V and not I positions in English:

(38) *David has been yawning.*



In contrast to English, only nontensed verbs appear within the VP in Russian. King (1995) provides several pieces of evidence supporting this claim. A VP constituent that contains a nontensed verb can be scrambled to a higher clause in colloquial Russian:[6]

(39) *mne [otpustit' Katju odnu] kažetsja, čto bylo by bezumiem.*
me let.go.INF Katja alone seem that would be insane
'It seems to me that it would be insane to allow Katja to go alone.'

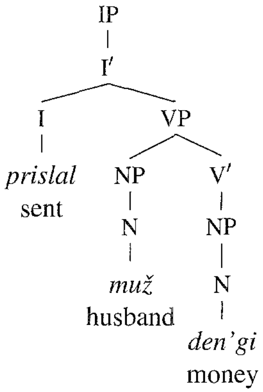
A similar construction involving a tensed verb is not grammatical, since the tensed verb is not a constituent of VP:

(40) **ja [pošel v školu] skazal, (čto) on.*
I went to school said that he
'I said that he had gone to school.'

[6] King attributes examples (39–40) to Yadroff (1992).

King concludes from this and other evidence, based on coordination and the distribution of the negative marker *ne*, that nontensed verbs are of category V in Russian. In contrast, all tensed verbs in Russian appear in I position (King 1995), as shown in example (41):.

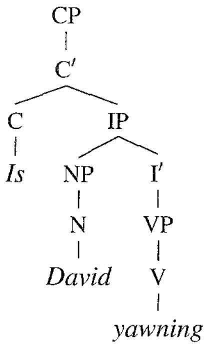
(41) *prislal muž den'gi.*
sent husband money
'My husband sent (me) money.'



5.1.2. The CP Phrase

In English questions, the auxiliary verb appears in C:

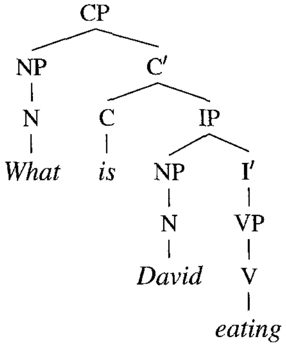
(42) *Is David yawning?*



Thus, a tensed English auxiliary appears in C in constructions involving subject-auxiliary inversion, and in I otherwise. Constraints on the functional structure of constructions requiring or forbidding subject-auxiliary inversion ensure that the auxiliary appears in the proper position in each instance.

The question word appears in the specifier position of CP in English, and the auxiliary verb appears in C (King 1995, Chapter 10):

(43) *What is David eating?*



Many languages are unlike English in configurational structure. We now examine some of this variability and discuss how languages can vary within the limits imposed by X′ theory and the universally available mappings between constituent structure and functional structure, which we examine in Chapter 4.

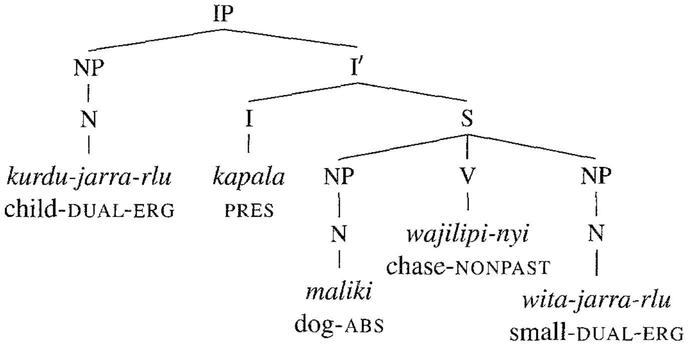
5.2. Exocentricity and Endocentricity: The Category S

We have seen that the constituent structure of English abides by X′-theoretic principles, with a head X of a maximal phrase XP and a nonmaximal phrase X′. Headed categories like XP and X′ are called *endocentric* (Bloomfield 1933), and the tendency for languages to make use of endocentric categories is what Bresnan (2001b) refers to as the *Principle of Endocentricity*.

In contrast, some languages allow an *exocentric* category, one that has no lexical head: the category S (Bresnan 1982a; Kroeger 1993; Austin and Bresnan 1996; Nordlinger 1998; Bresnan 2001b). S is a constituent structure category that contains a predicate together with any or all of its arguments, including the subject; for this reason, Austin and Bresnan (1996) call languages with the category S, including Tagalog, Hungarian, Malayalam, and Warlpiri, "internal subject" languages. We assume that S is the only exocentric category. As an exocentric category, the category S can dominate a series of either lexical or phrasal constituents.

Austin and Bresnan (1996) demonstrate that the phrase structure of a typical Warlpiri sentence is as follows:

(44) *kurdu-jarra-rlu kapala maliki wajilipi-nyi wita-jarra-rlu*
child-DUAL-ERG PRES dog.ABS chase-NONPAST small-DUAL-ERG
'The two small children are chasing the dog.'

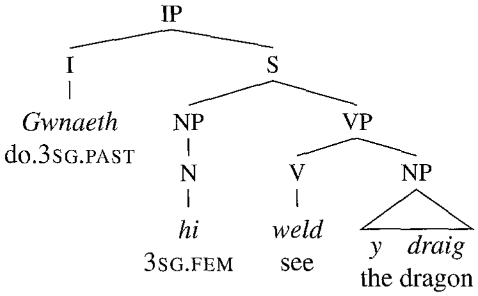


The Warlpiri I′ consists of an I, in which the clitic auxiliary appears, and an S complement. A phrasal constituent of any category appears as the specifier daughter of IP; in (44), a noun phrase fills the specifier of IP. Importantly, the words in this sentence can appear in any order, so long as the auxiliary appears in second position (Simpson 1991; Austin and Bresnan 1996). The daughters of S can be reordered freely, and any phrasal constituent can appear in the specifier of IP position.

In Chapter 4, we will discuss the close correlation between specifier and complement constituent structure positions as defined by X′ theory and the abstract grammatical functions of the phrases that appear in those positions. For example, in many languages the subject must appear in the specifier position of IP. In contrast, the relation between constituent structure positions dominated by S and their abstract functional role does not obey the same X′-theoretic constraints. Thus, in languages which, like Warlpiri, make use of the exocentric category S, constituent structure position is often not an indicator of grammatical function. Instead, grammatical functions are often marked morphologically, by means of case endings, with a concomitant tendency to freer word order.

The category S has also been proposed for languages that have relatively fixed word order. Sadler (1997) proposes a clausal structure like the following for Welsh, a VSO language:

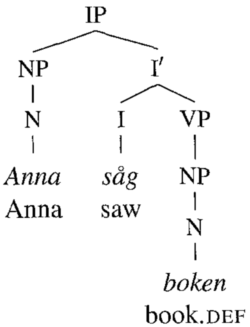
(45) *Gwnaeth hi weld y draig.*
do.3SG.PAST 3SG.FEM see the dragon
'She saw the dragon.'



Sadler notes that the tensed verb in Welsh is always of category I. The complement of I is S, which dominates the subject and a VP predicate.

Sells (1998) proposes an analysis of the constituent structure of Swedish and Icelandic according to which both languages have VP, a constituent that does not include the subject, but Icelandic also makes use of the exocentric category S. According to Sells's analysis, a typical Swedish sentence has a structure like the following:

(46) *Anna såg boken.*
Anna saw book.DEF
'Anna saw the book.'

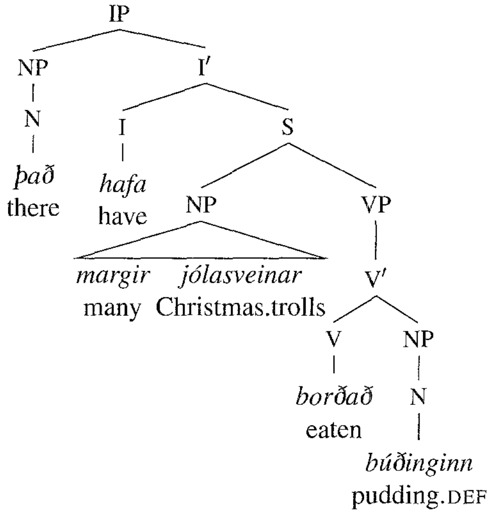


As Sells notes, Swedish (like English) does not allow the "transitive expletive construction," exemplified by (47): in this construction, the verb is in second position in the sentence, immediately preceded by an expletive subject and immediately followed by a thematic subject:

(47) **Det har många män ätit puddingen.*
there have many men eaten pudding.DEF
'Many men have eaten the pudding.'

Since Swedish does not allow the category S, it has only a single subject position and does not allow a sentence with two phrases filling subject positions. In contrast, the transitive expletive construction is permitted in Icelandic (Kaplan and Zaenen 1989; Sells 1998):

(48) *það hafa margir jólasveinar borðað búðinginn.*
there have many Christmas.trolls eaten pudding.DEF
'Many Christmas-trolls have eaten the pudding.'



Sells hypothesizes that this structure is possible in Icelandic because the presence of the S node makes available an additional phrase structure position that can be associated with a subject: the first daughter of S as well as the specifier position of IP can be filled by a subject phrase.

6. FURTHER READING AND RELATED ISSUES

There is a large literature on LFG treatments of morphology and morphosyntax in a variety of languages, much of which has not been covered in the preceding discussion. Of particular note is work by Dahlstrom (1986b) on Cree, Ackerman (1987) on Hungarian, Kanerva (1987) on Finnish, Arka (1993) on Indonesian, Huang (1997) on Chinese, Japanese, and English, Dahlstrom (1997) on Fox, Ackerman and Nikolaeva (1997) on Armenian and Ostyak, Börjars et al. (1997) on English and Latin, Schwarze (1997) on Romance, Ackerman and Webelhuth (1998) on German and other languages, Alsina (1999) on Chichewâ, Spencer (2000) on Slavic, and Sadler and Spencer (2000) on formal and architectural issues.

We will not discuss the phenomenon of "prosodic inversion," examined by Halpern (1995) and, in an LFG setting, by Kroeger (1993), Austin and Bresnan (1996), and Nordlinger (1998). LFG treatments of prosodic inversion often involve assuming a distinction between what Halpern calls "surface order" — the order of the words in a sentence — and "syntactic order," their order in the constituent structure tree. Newman (1996) outlines several proposals for the formal treatment of prosodic inversion in LFG.

4

SYNTACTIC CORRESPONDENCES

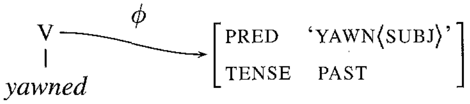
LFG hypothesizes that constituent structure and functional structure are mutually constraining structures and that the relation between these structures is given by information associated with words and phrases. We turn now to an investigation of the relation between these two facets of linguistic structure: in this chapter, we explore universally valid generalizations regarding the correlation between phrasal positions and grammatical functions.

Section 1 of this chapter discusses the formal representation of the relation between c-structure and f-structure. Section 2 presents a detailed examination of the relation between c-structure and f-structure: how c-structure phrases and their heads relate to f-structure, and the c-structure/f-structure relation of arguments and modifiers. Next, we examine apparent mismatches between units at c-structure and those at f-structure; Section 3 shows that these cases have a natural explanation within LFG. In Section 4, we discuss the *Lexical Integrity Principle*, the concept of wordhood, and the possibly complex contribution of words to functional structure. Finally, Section 5 discusses the formal structures that LFG uses to represent constituent and functional structure, and how this heterogeneous view contrasts with other theories.

1. RELATING CONFIGURATIONAL AND FUNCTIONAL STRUCTURE

As aspects of a single complex linguistic structure, c-structure and f-structure are related to one another in a finely specified way: the pieces of the c-structure that constitute the subject are related to the f-structure for the SUBJ, for example. Formally, the relation between the two structures is given by a function[1] called ϕ that relates c-structure nodes to f-structures. Each c-structure node is related by this function to a particular f-structure, and since ϕ is a function, no c-structure node can be related to more than one f-structure. Pictorially, the ϕ function is represented by an arrow labeled ϕ from the c-structure node to its corresponding f-structure:

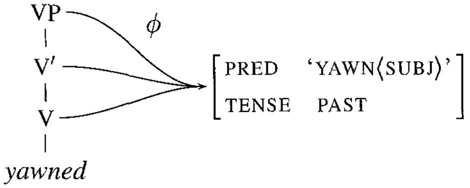
(1)



This diagram indicates that the word *yawned* is of category V and that the V node is associated with certain functional syntactic information: the f-structure corresponding to the V node has a PRED with value 'YAWN⟨SUBJ⟩', and the feature TENSE with value PAST.

Often, more than one c-structure node is related to the same f-structure. For instance, a phrase and its head correspond to the same f-structure:

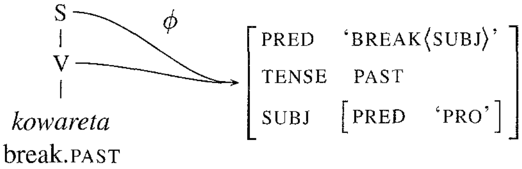
(2)



There are also f-structures that are not related to any c-structure node. In Japanese (Kameyama 1985), a "pro-drop" language with no verbal agreement morphology, the single word *kowareta* 'broke' can appear without an overt subject noun phrase. In such cases the SUBJ f-structure does not correspond to any node in the c-structure:

[1] Chapter 2, Footnote 10 (page 30) provides a definition of the term "function."

(3) *koware-ta*
break-PAST
'[It/Something] broke.'



2. REGULARITIES IN THE C-STRUCTURE/F-STRUCTURE MAPPING

Universally, the mapping function from c-structure to f-structure obeys principles relating X-bar-theoretic phrasal configurations to f-structural configurations: a phrase and its head always correspond to the same f-structure, for example, and the specifier and complements of a phrase always play particular functional syntactic roles. In this section, we examine these regularities and how they constrain the relation between the two syntactic structures of LFG.

2.1. Heads

The functional properties and requirements of the head of a phrase are inherited by its phrasal projections and become the functional properties and requirements of the phrases projected by the head. This means that a constituent structure head and the phrases it projects are mapped onto the same f-structure, as shown in (2). This condition was originally proposed by Bresnan (1982a, page 296) and discussed by Zaenen (1983), who calls it the *Head Convention*.

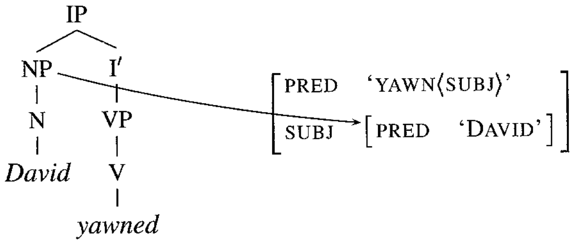
2.2. Specifier Positions

Recall that the specifier position of a constituent structure phrase is the position that is daughter of XP and sister to X′. Modifier phrases fill the specifier of a lexical category (Sadler 1998). Specifiers of functional categories such as IP or CP play special roles, mapping to the syntacticized discourse functions SUBJ, TOPIC, or FOCUS (Bresnan 2001b).

2.2.1. Specifier of IP

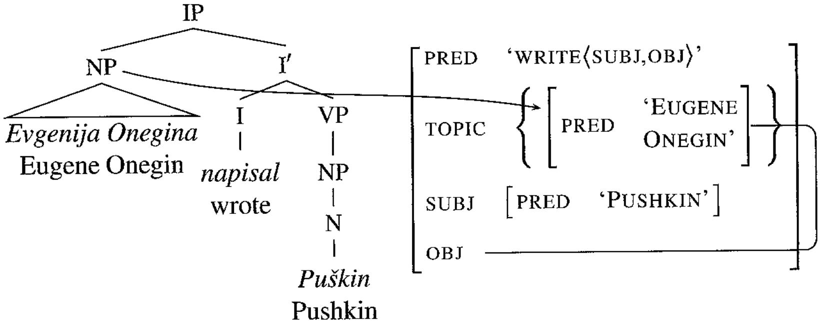
Specifier of IP is filled by SUBJ: In English and many other languages, the specifier position of IP is associated with the subj function:[2]

(4) *David yawned.*



Specifier of IP is filled by Topic/Focus: King (1995) shows that in Russian, the function associated with the specifier of IP is not the subject, but the discourse topic or focus. Since a Russian sentence may contain more than one topic, King represents the value of the topic feature as a set. The topic additionally bears some grammatical function within the clause; in example (5), it is the object (King 1995, page 206):

(5) '*Evgenija Onegina' napisal Puškin.*
Eugene Onegin wrote Pushkin
'Pushkin wrote 'Eugene Onegin'.'

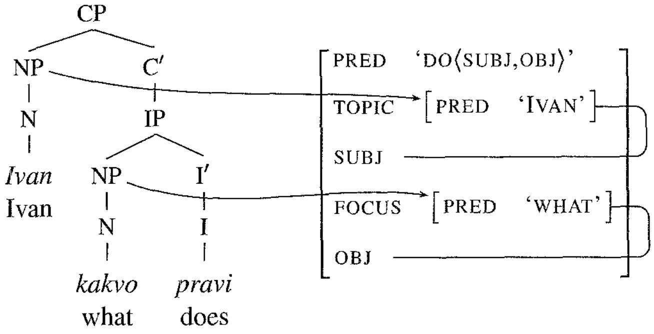


Here we have represented the topic as a part of the f-structure, as King does. In Chapter 7, Section 3, we will discuss other approaches to dealing with information-structural notions like topic and focus, which require a different level of representation for information structure, related to but separate from the f-structure.

[2] The tree in example (4) has a IP and an I′ node, but no node I; recall from our discussion in Chapter 3, Section 4.4 that all c-structure positions are optional.

Wh-phrases in Bulgarian fill the specifier position of IP (see Rudin 1985) and bear the syntacticized FOCUS function:

(6) *Ivan kakvo pravi?*
Ivan what does
'What is Ivan doing?'

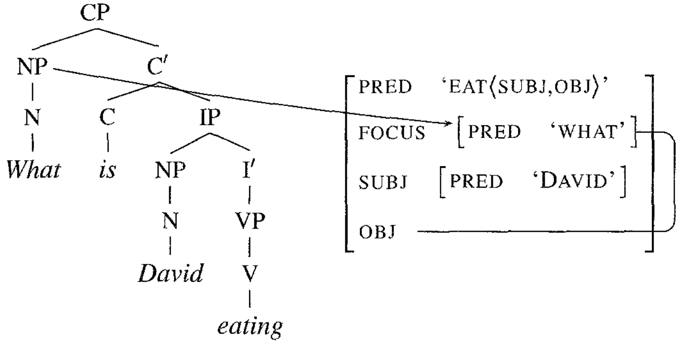


Izvorski (1993) provides further discussion of Bulgarian constituent structure.

2.2.2. Specifier of CP

In English, a wh-phrase like *what* appears in the specifier position of CP and bears the FOCUS role, as in example (7):

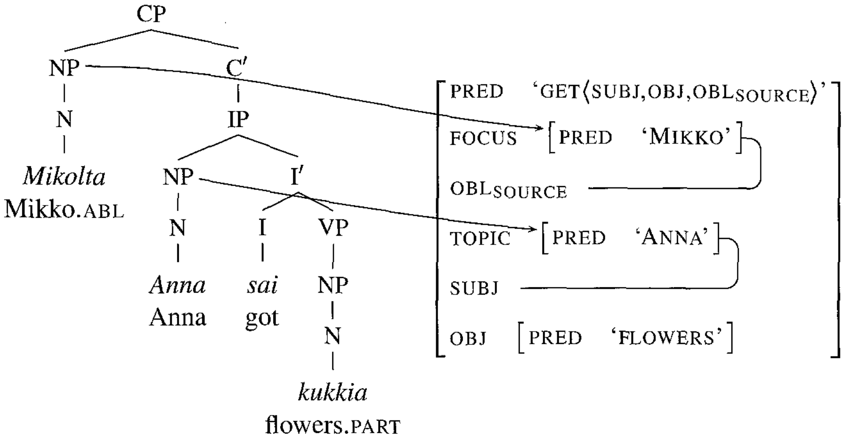
(7) *What is David eating?*



This is also true in Russian, as King (1995) shows.

In contrast, as shown in example (6) above, the specifier position of CP in Bulgarian is associated with the TOPIC function. Vilkuna (1995) argues that in Finnish, contrastive focus appears in the specifier position of CP, while TOPIC appears in the specifier position of IP, as shown in example (8):

(8) *Mikolta Anna sai kukkia.*
Mikko.ABL Anna got flowers.PART
'From Mikko, Anna got flowers.'

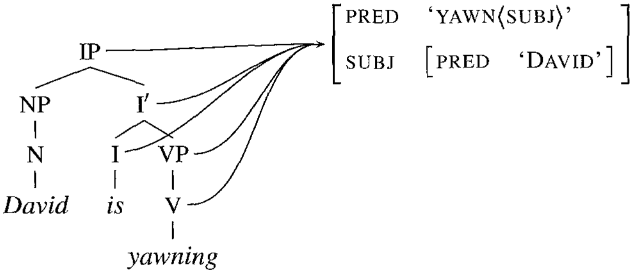


2.3. Complements

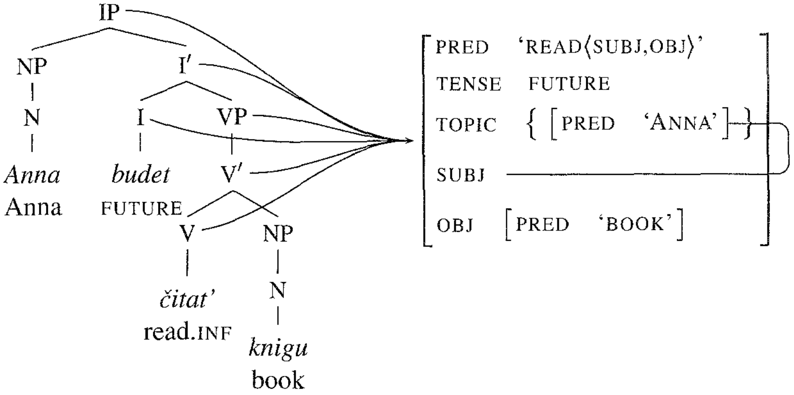
2.3.1. Complements of Functional Categories

Recall that complement positions are sisters to the head of a phrase. Complements of functional categories are f-structure co-heads (Kroeger 1993; King 1995; Bresnan 2001b), meaning that the functional properties of a functional category are shared with its complements. For example, the English IP shares the functional syntactic properties of its VP complement and its verbal head. Thus, an f-structure can be associated with two different c-structure heads, one a functional category and the other a lexical category, as shown in the English example in (9) and the Russian example in (10) (from King 1995, page 227):

(9) *David is yawning.*



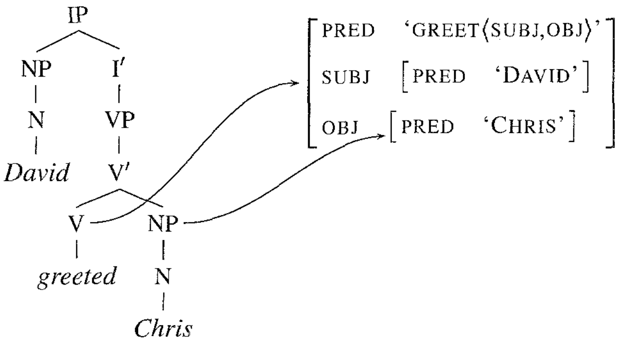
(10) *Anna budet čitat' knigu.*
Anna FUTURE read.INF book
'Anna will be reading a book.'



2.3.2. Complements of Lexical Categories

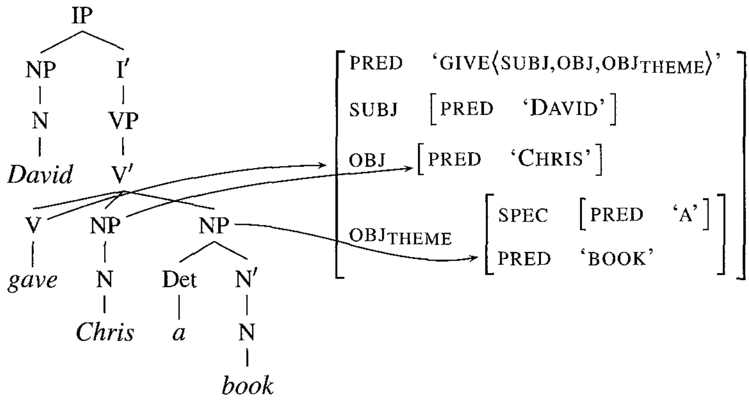
Bresnan (2001b) shows that complements of lexical categories are the non-discourse syntactic functions: that is, all of the governable grammatical functions except for SUBJ, as well as modifying adjuncts. In the following English example, the OBJ phrase *Chris* is a sister to V:

(11) *David greeted Chris.*



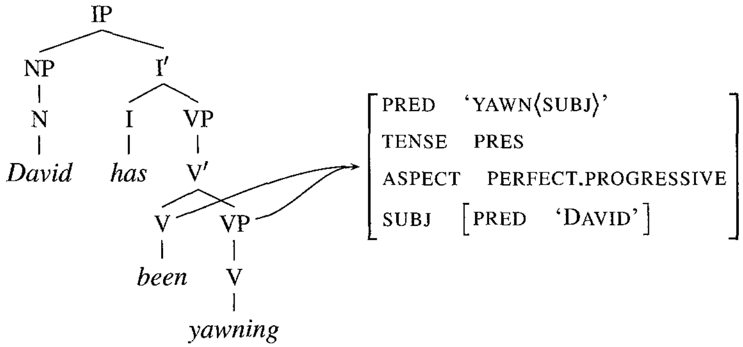
Of course, more than one complement may appear; in example (12), the first complement bears the OBJ role, and the second complement is OBJ$_{\text{THEME}}$:

(12) *David gave Chris a book.*



A phrase in complement position can also share the same f-structure with the head of the phrase, as in the case of English auxiliaries and their complement VPs:

(13) *David has been yawning.*



See Falk (1984, 2001), Butt et al. (1996), Sadler (1998), and Chapter 7 for more discussion of the English auxiliary system.

Since LFG does not define subcategorization properties of predicates in c-structure terms, the arguments of a predicate may but are not required to appear in complement positions. They may, for example, appear in the specifier position of some functional projection, bearing a discourse function like TOPIC or FOCUS. They are then linked to their within-clause function by means of purely functional specifications; see Chapter 14 for discussion.

2.4. Minor Categories and Idiosyncratic Constructions

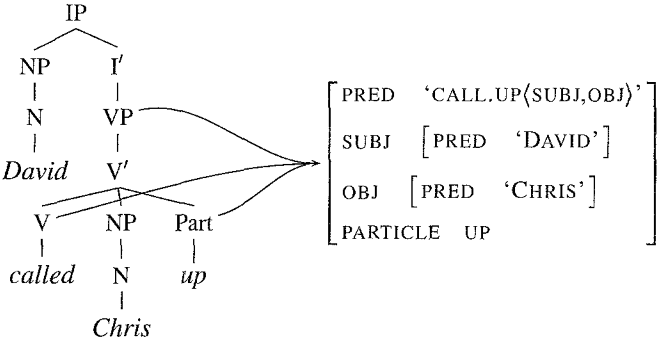
There is some degree of tension between the general principles relating constituent structure and functional structure, which we have been discussing, and the demands of idiosyncratic words and constructions in particular languages. Like Kay and Fillmore (1997), LFG aims to provide analyses of idiomatic language patterns as well as the relatively general properties of languages. In some cases, generalizations can be drawn about particular categories other than the major lexical and functional categories discussed; for example, Zaenen (1983) proposes the *Minor category convention*:

(14) Minor category convention:

Minor categories map onto the same f-structure as the node that immediately dominates them.

We adopt this treatment of minor categories such as the English particle, according to which a particle like *up* in the sentence *David called Chris up* contributes information to the f-structure of its verb phrase:

(15) *David called Chris up.*



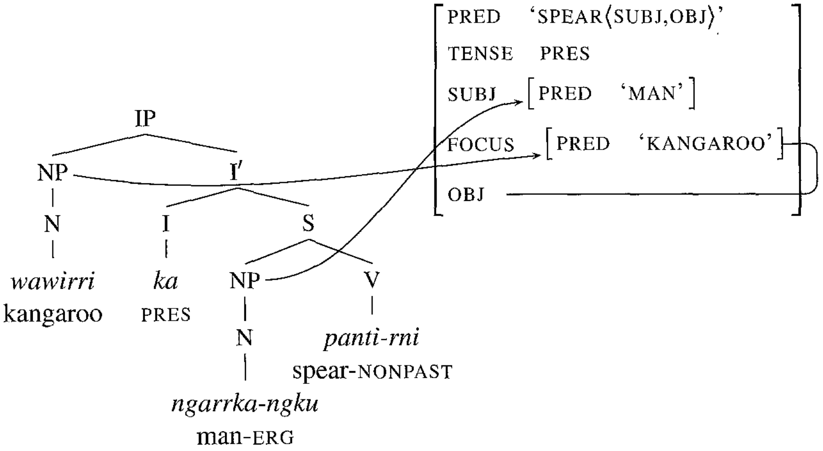
Comparatively little research has been done on the constituent structure properties of minor categories, so it is not easy to draw significant crosslinguistic generalizations about their distribution and phrasal properties. It seems clear that at least in some instances, their distribution is language dependent and does not fall under the general rules we have been discussing. We must examine other idiomatic, language-specific, or construction-specific syntactic properties on a case-by-case basis to determine their properties.

2.5. The Exocentric Category S

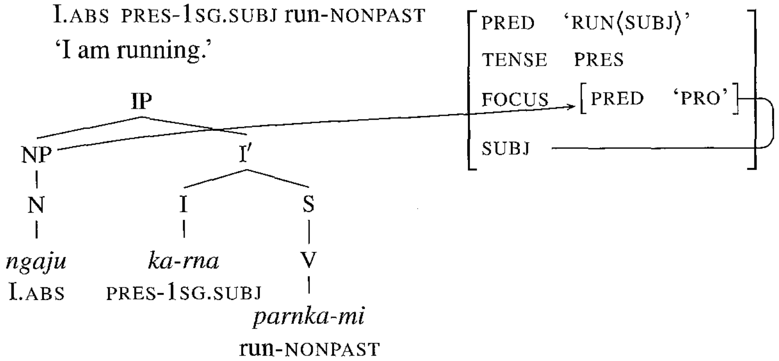
The exocentric category S is not an X′-theoretic category and does not obey the X′-theoretic generalizations governing the relation between c-structure positions and f-structure functions. Thus, in languages that are unlike English in having the exocentric category S, phrase structure configuration is not always an unambiguous indicator of grammatical function: phrases with different grammatical functions may appear in the same constituent structure position. For this reason, as Nordlinger (1998) points out, languages with S often allow relatively free word order and rely more heavily on morphological marking rather than phrase structure configuration for the identification of grammatical functions.

The S node can dominate phrases with any grammatical function, including SUBJ. In Warlpiri, the subject *ngarrka-ngku* 'man' may appear inside S (Simpson 1991), as shown in example (16a); in contrast, in example (16b) the subject appears outside S.

(16) a. *wawirri ka ngarrka-ngku panti-rni*
kangaroo PRES man-ERG spear-NONPAST
'The man is spearing the kangaroo.'



b. *ngaju ka-rna parnka-mi*
I.ABS PRES-1SG.SUBJ run-NONPAST
'I am running.'

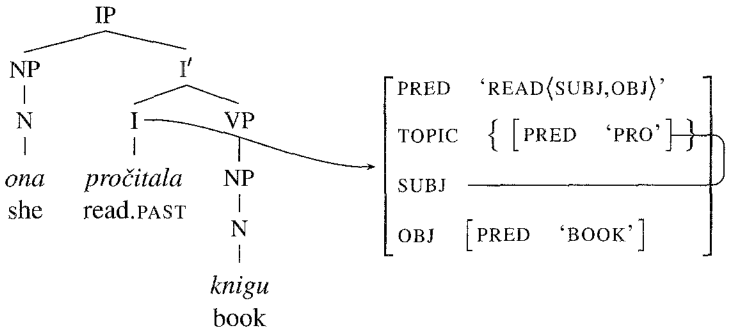


3. "MOVEMENT" AND DISCONTINUITY

3.1. Apparent Head "Movement"

As discussed in Chapter 3, Section 4.4, all constituent structure positions are optional; in particular, the head of a phrase need not be realized. Consider the following Russian example in which the tensed verb appears in I (King 1995, Chapter 10):

(17) *ona pročitala knigu.*
she read.PAST book
'She read the book.'



Examples such as these need no special analysis within our theory, and the verb is not thought of as having "moved" to the position in which it appears. Rather, the principles we have outlined so far predict that this configuration is possible and well formed. In Russian, all tensed verbs have the phrase structure category I; the verb in example (17) appears in I and not within VP. It is not possible for there to be two main verbs in a single Russian sentence, one in the I position and one within the VP, since this would produce an ill-formed f-structure; each verb contributes a PRED value to its f-structure, and the Consistency Principle forbids an f-structure from containing a PRED feature with two different semantic forms as its value. Conversely, a sentence with no verb is ill-formed, since in that case the main f-structure would contain no PRED, and the Completeness Principle would be violated. Exactly one verb must appear, and it must appear in the appropriate position for its constituent structure category.

3.2. Clitic Doubling

In most dialects of Spanish, full noun phrase objects appear after the verb, while clitic object pronouns appear preverbally. This has been analyzed in other theories as an instance of apparent movement (see, for example, Baltin 1983):

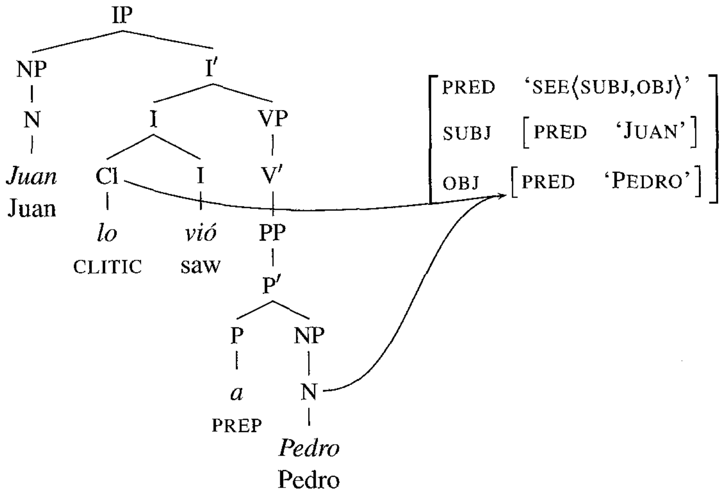
(18) a. *Juan vió a Pedro.*
Juan saw PREP Pedro
'Juan saw Pedro.'

b. *Juan lo vió.*
Juan him.ACC.SG.CLITIC saw
'Juan saw him.'

In fact, though, no special analysis is required for Spanish, and no "movement" need be assumed. Two phrase structure positions are associated with the OBJ function in Spanish, a preverbal clitic position and a postverbal phrasal position. If a verb is transitive, an OBJ must appear, and one of these positions must be filled, so that either a clitic or a full noun phrase must appear. In standard Spanish, it is not possible to fill both phrase structure positions: this would cause the OBJ function to be associated with a PRED with two different values, and the Consistency Principle would rule out this ill-formed possibility. Thus, exactly one object phrase must appear, and it must appear in the constituent structure position appropriate for its phrase structure category.

Interestingly, however, complementary distribution between a clitic and a full noun phrase is not always found:

(19) *Juan lo vió a Pedro.*
Juan him.ACC.SG.CLITIC saw PREP Pedro
'Juan saw Pedro.'



In dialects of Spanish which allow *clitic doubling*, it is possible for both OBJ positions to be filled, as in example (19). This is true of the River Plate and

Peruvian dialects of Spanish as described by Jaeggli (1980) (see also Grimshaw 1982; Bresnan 2001b). In these dialects, the object clitic pronoun *lo* is undergoing reanalysis as an agreement marker, and has begun to lose its semantic status as a pronominal. Formally, this reanalysis is reflected as *optionality* of the PRED value of the form (see Chapter 5, Section 2.4): *lo* need not contribute a PRED value to the f-structure. Since its other features are compatible with the full phrasal OBJ, both the clitic and the full noun phrase can appear in the same sentence.

3.3. Category Mismatches

Phrases not appearing in their canonical positions can sometimes exhibit *category mismatches*, as discussed by Kaplan and Bresnan (1982), Kaplan and Zaenen (1989), and Bresnan (2001b). The verb *aware* is ambiguous, subcategorizing either for a sentential COMP argument as in (20a), or an oblique phrase as in (20b):

(20) a. *We weren't aware that Chris yawned.*

b. *We weren't aware of the problem.*

Notably, grammaticality judgements differ when the argument of *aware* is displaced, appearing at the beginning of the sentence, as the examples in (21) show.

(21) a. **That Chris yawned we weren't aware.*

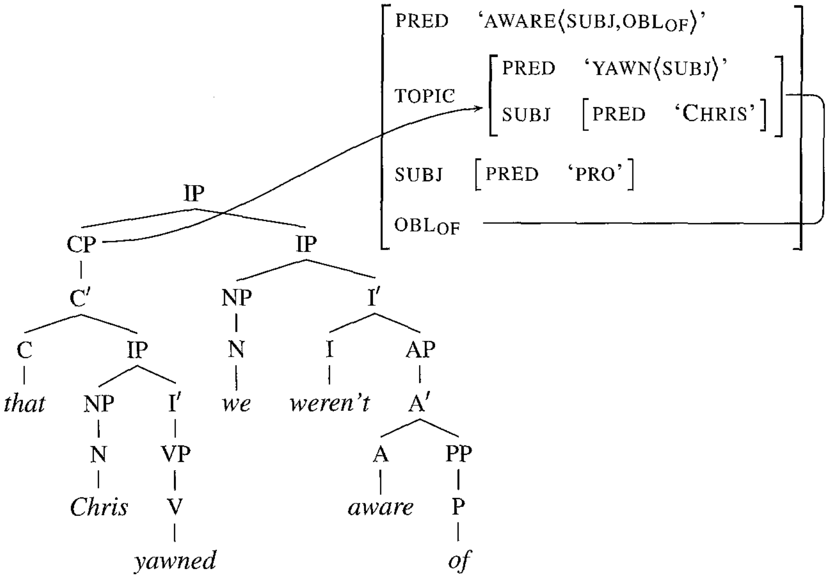
b. **We weren't aware of that Chris yawned.*

c. *That Chris yawned we weren't aware of.*

Example (21a) contrasts in grammaticality with (20a); further, although example (21b) is ungrammatical, example (21c) is fully acceptable. Such examples are mysterious on theories that analyze displaced phrases in terms of phrase structure movement; on such theories, there is no reason to expect the "moved" constituent to have different properties from its in-situ counterpart. That is, if we assume that an example like (21a) is derived from a source like (20a) by movement of the complement to initial position, it is not clear what would account for the different status of the two examples.

As noted by Kaplan and Zaenen (1989), a nontransformational account fares better in accounting for instances of apparent mismatch. In LFG, the relation between a sentence-initial constituent in an example like (21a) and its within-clause role is defined at f-structure rather than c-structure, as shown in (22) (page 82). There is no sense in which the phrase *that Chris yawned* has been moved or displaced in the c-structure, and thus no reason to suspect that the c-structure category of this phrase is required to satisfy the phrase structure requirements that hold of prepositional phrase complements; rather, the relation between the topicalized phrase and its within-clause function is specified in functional rather

(22) *That Chris yawned we weren't aware of.*



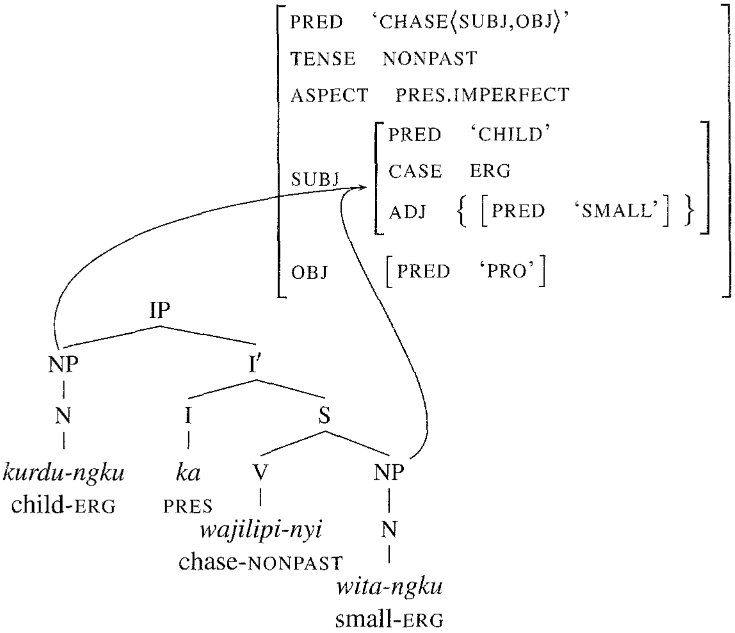
than constituent structure terms, and phrase structure category is not a part of the specification.[3]

3.4. Constituent Structural Discontinuity

Simpson (1991, Chapter 5) provides a very interesting discussion of nonconfigurationality and discontinuity in Warlpiri (see also Austin and Bresnan 1996; Bresnan 2001b). She shows that the subject of a Warlpiri sentence can appear as a daughter of S (as in example 16a) as well as in the specifier position of IP (as in example 16b). As we would expect, then, phrases in each of these positions — or, crucially, in *both* positions — can contribute to the structure of the SUBJ of a sentence. In example (23), parts of the SUBJ appear in S while other parts appear in the specifier of IP: the SUBJ phrase *kurdu-ngku* 'child' appears in the specifier of IP, while a modifier of the SUBJ, *wita-ngku* 'small', appears within the S.

[3] The relation between a topicalized phrase and its within-clause function is specified by means of *functional uncertainty*, see Chapter 6, Section 1.1 for definition and discussion of functional uncertainty and Chapter 14 for a complete discussion of long-distance dependencies.

(23) *kurdu-ngku ka wajilipi-nyi wita-ngku*
child-ERG PRES chase-NONPAST small-ERG
'The small child is chasing it.'



These two phrases are functionally compatible, since both bear ERG case, one represents the head of the subject phrase, and the other represents a modifier. Here too, we need say nothing special about such examples; our theory predicts that we will find such examples, and we need no special analysis to encompass them.

4. THE LEXICAL INTEGRITY PRINCIPLE

The *lexicalist hypothesis* was first discussed within a transformational framework by Chomsky (1970), who advanced the claim that rules of word formation are lexical rules rather than transformations: words and phrases are built up from different elements and by different means. The hypothesis exists in two forms. Its weaker form (the "weak lexicalist hypothesis") states that derivational morphology is defined in the lexicon, not by transformational rule. This hypothesis is generally assumed to hold: the majority of current linguistic theories, even those that assume transformational rules, do not propose transformations that change the phrasal category of words.

Bresnan (1978) adopts a stronger form of the hypothesis, claiming that "syntactic transformations do not play a role in word formation," either in derivational or inflectional morphology. Lapointe (1980) builds on this claim, proposing what is usually called the "strong lexicalist hypothesis" or the *lexical integrity principle*:[4]

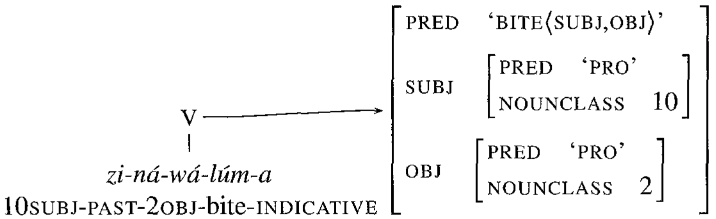
(24) No syntactic rule can refer to elements of morphological structure. (Lapointe 1980, page 8)

Current work in LFG adheres to this principle: LFG does not assume the existence of processes that assemble words in constituent structure or reorder subparts of words during syntactic composition.

There is a great deal of evidence supporting this view. Niño (1995) provides a particularly clear discussion of evidence from Finnish, arguing against views in which inflectional morphemes are assembled by syntactic movement; she shows that one cannot analyze Finnish inflectional morphemes as syntactically independent forms. Rather, as she states, "their syntactic role is limited to contributing the features they carry, and they combine into fully inflected morphological words exclusively in the lexicon."

Although words are not subject to processes of assembly in the constituent structure, their syntactic contributions can be complex. A word that forms a unit at the level of c-structure may introduce complex functional structure, structure that in other languages might be associated with a phrase rather than a single word. For example, Bresnan and Mchombo (1987) present evidence that subject and object markers on the Chicheŵa verb can function as incorporated pronouns. These affixes form a part of the verb at c-structure, meaning that the f-structure corresponding to this verb is complex, similar to the f-structure for a sentence in a language like English.[5] In (25), the marker *zi* represents a pronominal SUBJ of noun class 10, and the marker *-wá-* represents a pronominal OBJ of noun class 2:

(25)



[4] Lapointe (1980) refers to this as the "generalized lexicalist hypothesis."

[5] See Chapter 5, Section 4.3 and Chapter 11 for more discussion of the pronominal status of the subject and object markers in Chicheŵa.

As this example shows, the notion of word is multifaceted and must be fully defined at each syntactic level of representation; units at c-structure need not correspond to simple units at other levels.

Much other work has been done within LFG in support of the Strong Lexicalist Hypothesis, demonstrating convincingly that a word that is atomic at the level of c-structure can project the structure of a phrase at functional structure. Bresnan (2001b, Chapter 6) discusses Lexical Integrity and its relation to the principle of *Economy of Expression*, a principle that requires the choice of the simplest and smallest phrase structure tree that allows for the satisfaction of f-structural constraints as well as expressing the intended meaning. Simpson (1983, 1991), Ackerman (1987), O'Connor (1987), Bresnan and Mchombo (1995), Nordlinger and Bresnan (1996), Mohanan (1994, 1995), Matsumoto (1996), Sadler (1997), Ackerman and LeSourd (1997), and Börjars (1998) have also made valuable contributions to our understanding of wordhood and lexical integrity.

5. LINGUISTIC REPRESENTATIONS AND RELATIONS

We have seen the need for an abstract representation of functional organization, separate from phrasal organization. LFG shares with a number of current linguistic theories the view that functional, configurational, and other linguistic structures reflect what Sadock (1991) calls "parallel organizational principles": the various facets of linguistic organization are copresent, and each aspect of linguistic structure is organized according to its own cohesive set of rules and principles. This view contrasts with a commonly held view of traditional transformational grammar and its descendents, where different levels of linguistic structure are formally similar structures, derived from one another via transformational rules.

Does it matter how linguistic structure is represented and how the different facets of structure are related? In fact, this issue is vitally important: the use of inappropriate representations and relations between representations makes it difficult to form solid intuitions about the linguistic structure underlying the representations, and can lead to incorrect and obscured views of linguistic typology. Below we will discuss some alternative views and why they are unsuccessful.

5.1. The LFG View

LFG does not require that structures representing different sorts of linguistic information must be of the same formal type. Rather, the choice of how a particular kind of information is represented is made on the basis of how that kind of information is best expressed.

For instance, dominance and precedence conditions and phrasal groupings are clearly and succinctly represented in terms of a familiar phrase structure tree, the c-structure. In the case of functional syntactic information, however, a phrase structure tree is not the best way of perspicuously and unambiguously representing this information, as trees carry with them certain presuppositions about the information being represented. For instance, trees encode a linear order, which does not make sense in the functional realm; further, nodes in a tree are required to have a single mother, implying that a phrase can play only a single functional role.

Instead, as originally proposed by Kaplan (1975a,b), an attribute-value matrix like the f-structure is a very good way of representing functional information. The functional structure does not contain information about linear order relevant for the description of constituent structure but irrelevant to functional organization; the fact that a single f-structure may be the value of more than one functional structure attribute allows us to represent the fact that a single phrase may play multiple functional roles. Using different structures and relations for representing constituent structure and functional information allows more freedom in associating different c-structure positions with the same grammatical functions, as in a language like Spanish (discussed in Section 3.2 of this chapter), or different functions to the same c-structure position, as in a language like Warlpiri (discussed in Section 2.5 of this chapter).

Other linguistic information might be best represented in terms of different formal structures. For instance, as we will see in Chapter 9, a deductive semantic approach meshes well with the overall LFG architecture; such an approach is well suited to the expression of meanings in terms of formulas in a logical language like those standardly used in formal semantic theory.

5.2. The Transformational View

Transformationally based syntactic theories characterize the relation between abstract functional structure and phrasal structure in terms of syntactic transformations. In one influential view (Chomsky 1981, 1993), linguistic structures are uniformly represented as trees like those used in LFG to represent constituent structure and are systematically transformed from one representation to another one by the rule of move-α, according to which a constituent moves to another position in the tree. This approach is appealing in its simplicity, but it has some unfortunate consequences. Because transformations are defined as minimal changes to a particular formal representation, yielding a representation of the same formal type, both the abstract syntactic representation and the surface representation must be represented as phrase structure trees.

This means that, among other things, a mechanism must be devised for expressing abstract functional syntactic relationships configurationally, in constituent structure terms. A theory that defines grammatical functions configurationally describes languages that are unlike English by proposing a more abstract constituent structure to encode the functional information that is found in f-structure, related to a representation of surface phrasal groupings by transformational rules. This theoretical move forces the conflation of very different kinds of information in the same linguistic structure. In particular, the relation between the more abstract functional syntactic structure and the constituent structure is recorded by means of traces in the surface constituent structure representation. These represent a residue of functional information in the constituent structure that is motivated only by theory-internal considerations, and mean that the final tree must contain a confusing mix of phrasal and functional information. Additional phrase structure constituents must also be introduced in the analysis of cases in which the surface utterance does not have the right properties or contain the correct constituents for abstract syntactic analysis.

Let us make a closer examination of two prominent proposals for representing abstract syntactic information in terms of a phrase structure tree. Hale (1983) analyzes "nonconfigurational" languages like Warlpiri, noting that they nevertheless have abstract functional syntactic structure similar to "configurational" languages like English (see Chapter 2, Section 4.2): for example, subjects behave differently from objects in Warlpiri. To account for the difference between configurational (English-like) and nonconfigurational (Warlpiri-like) languages, Hale proposes that the syntactic structure of nonconfigurational languages is typologically completely different from configurational languages.

Specifically, Hale posits two abstract syntactic structures: "Lexical Structure" or LS, and "Phrase Structure" or PS. In configurational languages like English, LS and PS are very similar. In contrast, LS and PS are different in nonconfigurational languages like Warlpiri: in these languages, LS is an English-like structure, with the subject phrase c-commanding the object phrase, while PS is a flat structure with no asymmetry between the subject and the object. Hale further proposes that the Projection Principle (Chomsky 1981) applies at PS as well as LS in configurational languages. This requirement entails that argument selection must be maintained across levels in the transformational derivation in such languages; that is, each constituent representing an argument of a predicate must appear at LS as well as PS. However, in nonconfigurational languages like Warlpiri, the Projection Principle applies only at LS.

As argued in detail by Speas (1990), there is good reason to doubt the existence of a sharp division like this one, which predicts the existence of two completely different types of languages. As Speas notes, many languages that have been classified as "nonconfigurational" do not share the properties we would expect of a Warlpiri-type language, while many languages that have been classified as "configurational" behave in some ways like Warlpiri. The distinction between languages of the English type and those of the Warlpiri type is, then, not as sharp as Hale's theory would predict. In fact, as Klavans (1982) points out, the architecture of LFG leads to a different and more satisfactory prediction: languages vary greatly in c-structure organization, whereas functional structure is much less variant across languages. Thus, we would expect a gradation in the spectrum between "configurational" and "nonconfigurational" languages, not a black-and-white distinction as Hale proposes, since grammatical function encoding can vary from language to language: grammatical functions can be encoded completely or partially configurationally, or completely or partially morphologically.

Another transformationally based view is presented by Speas (1990), who argues that all languages have the same abstract syntactic structure, defined in phrase structure terms. That is, she proposes that English-like phrase structure is universal, and all languages have configurationally defined grammatical functions.

Besides the above-noted objections to such an approach, Van Valin (1987) argues convincingly that Speas's approach does not work for a number of languages. In particular, he examines Lakhota and shows conclusively that Lakhota has no VP: his arguments are based on facts involving free word order, lack of subject/object extraction asymmetries, lack of weak crossover effects, and dependence of binding facts on linear order rather than c-structure hierarchical distinctions. In a theory like Speas's, these facts run counter to the assumption that the phrase structure of every language is English-like. However, assuming that Lakhota has no VP creates other difficulties for Speas's theory in that it leads to a prediction that Van Valin shows to be false: that there are no differences between subject and object in Lakhota. To account for casemarking and binding patterns, it is necessary to appeal to the abstract grammatical functions subject and object, which in Speas's theory can only be accounted for by assuming a VP. Thus, Speas's approach leads to a contradiction in the case of Lakhota: some facts indicate that Lakhota does not have a VP, while other facts are only explicable by assuming that it does have a VP.

More generally, defining abstract functional properties like subjecthood and objecthood in constituent structure terms predicts that a phrase in a particular phrase structure position will either exhibit the full range of properties that are supposed to be associated with that position or will exhibit none of these properties; this is because appearing in a particular structural position implies a particular range of functional behavior. In fact, in theories that treat abstract functional structure in this way, the notion of "grammatical function" is not really defined in a modular way. Instead, a range of grammatical phenomena are taken to be sensitive to a particular phrasal configuration, for instance whether a phrase is inside or outside the VP. The inflexibility of such a model is suspect. On such a view, it is difficult to determine how different phrase structure positions can be associated with the same functional syntactic properties; it is also difficult to see how different grammatical functions could be expected to behave similarly or how a grammatical function hierarchy can be made to follow naturally from such a view.

The architecture of LFG is based on the assumption that linguistic structure is multifaceted and that different kinds of information may be best represented in different ways. Forcing everything into the same mold obscures generalizations and is not conducive to the formation of solid intuitions about the nature and characteristics of the various structures.

5.3. Other Views

LFG shares with a number of other grammatical theories the view that representations of different aspects of linguistic structure need not be of the same formal types. The formal architecture of Construction Grammar (Kay 1998), in which the syntactic objects are trees whose nodes are associated with feature structures, is fairly close to that of LFG. And some versions of Categorial Grammar also allow linguistic representations of different formal types; Oehrle (1999) provides a view of LFG as labeled deduction in categorial terms (Gabbay 1996), where the correspondences between different structures are represented as relations between labels on formulas.

Other theories that are very similar in spirit to LFG reject this view, however. In particular, Head-Driven Phrase Structure Grammar (Pollard and Sag 1994) is like LFG in representing different facets of linguistic information in terms of a set of parallel representations; it is unlike LFG in assuming that these structures are subparts of a single, homogeneous structure, the *sign*, represented as an attribute-value structure like the f-structure.

As we have seen, attribute-value structures are valuable representations for many kinds of information, in particular abstract functional syntactic information. It is of course possible to represent other kinds of information in terms of attribute-value structures; the phrase structural information that LFG represents as a constituent structure tree can also be represented in terms of an attribute-value structure.[6]

However, representing phrase structure information in terms of attribute-value structures has the potential for leading to confusion. The same formal properties that render trees an inappropriate representation for functional information make them a very good representation for phrase structure information. Attribute-value structures are not inherently ordered, and they allow a node to have more than one mother. On the other hand, the words of an utterance do appear in a particular order, and (on most theories of phrase structure organization) a c-structure node can-

[6] A tree is nothing more than a special kind of attribute-value structure: one in which a linear ordering between nodes is imposed, so that a node can be said to precede or follow another node; cycles are disallowed, so that a node cannot both dominate and be dominated by another node; and values of attributes may not be shared.

not be dominated by more than one mother node.[7] The particular characteristics of phrasal structure are not concisely and intuitively captured by an attribute-value representation; the additional properties that make constituent structure different from functional structure are not a part of the way these structures are represented.

On a different level, problems arise in forming theories of how linguistic structures can be computed (in either a psycholinguistic or a computational sense). The formal properties of trees are well understood, and simple ways of computing with trees are also well known. In computing with unenriched attribute-value structures, less structure is imposed by the choice of representation and so less advantage can be taken of the inherent properties of the structure. Representing semantic information in attribute-value terms is also problematic, as pointed out by Pereira (1990) and Dalrymple et al. (1997b).

On the LFG view, the representation of each type of linguistic information by means of structures that reflect the nature of the information allows for a systematic, flexible, and principled account of relations between levels, promotes clear intuitions about the formal and linguistic properties of each level, and aids in developing reasonable ways of thinking about interactions between levels and processing at each level.

6. FURTHER READING AND RELATED ISSUES

There has been a great deal of work in LFG on the relation between constituent structure and functional structure in a typologically diverse set of languages. Of particular note, besides the work mentioned earlier, are Johnson (1986), Dahlstrom (1987), Kaplan and Zaenen (1989), and Huang (1990), who discuss discontinuity, phrasal constituency, and the c-structure/f-structure relation; and Sells (1995), who discusses evidence from raising in Philippine languages and its bearing on clause structure. Bresnan (2001b, Chapter 5) also provides a lucid discussion of the *fragmentability of language*, the fact that the c-structure and f-structure of sentence fragments can be easily inferred based on lexically and phrasally specified constraints on syntactic structures. Butt et al. (1999) present a comparative overview of English, French, and German constituent and functional structures and the relation between them, discussing a wide range of syntactic constructions and issues that arise in the analysis of these languages.

[7] Some approaches allow phrase structure trees that violate some of these formal conditions; see, for example, McCawley (1988). McCawley's trees are, then, a more abstract representation of syntactic information than the constituent structures of LFG.

5

DESCRIBING SYNTACTIC STRUCTURES

Up to this point, our discussion has concentrated on the nature and representation of the two syntactic structures of LFG. We will now demonstrate how to formulate descriptions and constraints on c-structure, f-structure, and the relation between them, and we will see how these constraints are important in the statement of universal typological generalizations about linguistic structure. These constraints are a part of the *formal architecture* of LFG theory.

The job of the designer of a formal linguistic framework, like the framework of LFG, is to provide a way of stating linguistic facts and generalizations clearly and precisely, in a way that is conducive to a solid understanding of the linguistic structures that are described and how they are related. As we have seen, choosing the right way of representing a linguistic structure is very important in understanding it clearly and in avoiding confusion about the properties of the structure and its relation to other structures. A formal linguistic theory must provide efficient and transparent ways of describing the facts of a language. Linguistically important generalizations should be easy to express, and the linguistic import of a constraint should be evident.

The job of a linguist working within a formal framework is to discover the facts of a language or of languages and to express these facts by using the tools provided by the framework so that the linguistic facts emerge precisely and intuitively. Most importantly, the linguist must distinguish between constraints that are needed and used in describing the facts of language, the *linguistically relevant* constraints, and those that may be expressible within the overall formal framework but do not play a role in linguistic description. A well-designed formal framework aids the linguist in deciding which constraints are relevant and which are not in the description of linguistic structures.

1. CONSTITUENT STRUCTURE RULES

1.1. Phrase Structure Rule Expansions

Most linguistic theories that represent phrasal information in terms of phrase structure trees make use of *phrase structure rules* to express the possible and admissible phrase structure configurations in a language:

(1) S $\longrightarrow$ NP VP

This rule permits a node labeled S to dominate two nodes, an NP and a VP, with the NP preceding the VP. In LFG, phrase structure rules are interpreted as node admissibility conditions, as originally proposed by McCawley (1968): a phrase structure tree is admitted by a set of phrase structure rules if the rules license the tree. In other words, phrase structure rules are thought of as descriptions of admissible trees, and the trees of the language must meet these descriptions. McCawley's groundbreaking work constituted an important alternative to the way of thinking about phrase structure rules prevalent in the mid-1960s, which viewed phrase structure rules as a procedural, derivational set of instructions to perform a series of rewriting steps.

In many theories of phrase structure specification, all phrase structure rules are of the type illustrated in (1): the right-hand side of the rule (here, NP VP) specifies a particular unique admissible configuration. Constituent structure rules in LFG are more expressive than this, in that the right-hand side of a phrase structure rule consists of a *regular expression* (Kaplan and Bresnan 1982, page 277), which allows a sequence of category labels in which some categories may be optional, some categories may be repeated any number of times, and disjunction is permitted.[1] This added expressivity does not take us beyond the power of context-free languages, as demonstrated by Woods (1970).

Since phrase structure rules represent constraints on phrase structure configurations, it is reasonable and desirable to abbreviate a large or even an infinite number of rules by means of a regular expression. A phrase structure rule is not thought of as a large or infinite set of rules that can be applied individually in a series of rewriting steps, but as a characterization of the daughters that any node can dominate. This constraint-based view of linguistic descriptions pervades the formal theory of LFG: a grammar of a language consists of a set of constraints on linguistic structures, and these constraints define the analysis of an utterance.

A simple LFG phrase structure rule can have the following form:

(2) IP ⟶ {NP|PP} I′

This rule indicates that either a NP or a PP can appear in the specifier position of IP. The curly brackets mark a disjunction of phrase structure categories, with the possibilities separated by a vertical bar |. This rule abbreviates the following two rules, where the disjunction in (2) has been fully expanded into two separate rules:

(3) a. IP ⟶ NP I′
    b. IP ⟶ PP I′

The abbreviation in (2) is not only more compact than the two-fold expansion in (3), but it is more revealing of the linguistic facts: the specifier position of IP can be filled either by NP or by PP without affecting any properties of the second daughter of IP, I′. Stating this fact by means of two separate rules, as in (3), makes it appear that the I′ might have different properties in the two different rules.

[1] See Partee et al. (1993, section 17.2) for a detailed explication of regular expressions and the languages they describe, called *regular languages*. Formally, a regular expression obeys the following rules:

1. The empty string or a single symbol is a regular expression.
2. The disjunction of two regular expressions is a regular expression.
3. The concatenation of two regular expressions is a regular expression.
4. A regular expression annotated with the *Kleene star operator* *, indicating that the expression may be repeated zero or more times, is a regular expression. A regular expression annotated with the *Kleene plus operator* +, indicating that the expression may be repeated one or more times, is a regular expression.

Additionally, the regular languages are closed under the operations of union, intersection, and complementation. In other words, the union or intersection of two regular languages is also a regular language, and the set complement of a regular language is also a regular language. Since this is true, regular languages can also be defined in terms of these operators.

In example (4), the parentheses around the NP indicate optionality, and the Kleene star annotation * on the PP indicates that any number of PPs may appear in the expansion of the rule:

(4) VP ⟶ V (NP) PP*

Thus, this rule admits trees in which a VP node dominates a V, an optional NP, and any number of PPs. The use of the Kleene star means that an infinite set of possible rules — rules with any number of PPs — can be abbreviated with a single expression. Thus, phrase structure generalizations can be stated once rather than separately for each possible phrase structure configuration licensed by the rule.

1.2. Metacategories

Example (2) illustrates the use of disjunction over a set of categories in a rule. It is also possible to introduce an abbreviation over a set of categories in a rule: for instance, an abbreviation like XP is often used to represent a set of categories that behave similarly in some way. In such a situation, XP is a *metacategory* representing several different sets of categories.

King (1995) uses metacategory abbreviations like the following in her analysis of Russian:

(5) CP ⟶ XP C′
IP ⟶ XP I′

The use of XP in these rules indicates that a full phrase of any category (NP, PP, and so on) can appear as the first daughter of CP and IP. XP is defined as follows:[2]

(6) XP ≡ {NP | PP | VP | AP | AdvP }

The use of abbreviations like this allows for the expression of more general statements about all phrases that appear in a particular phrase structure position.

In example (6), the metacategory XP stands for any one of a number of phrasal categories. In fact, a metacategory can abbreviate a longer sequence of categories, not just a single category (Kaplan and Maxwell 1996). This is shown in the following putative definition of VP:

(7) VP ≡ V NP

More generally, a metacategory can be used as an abbreviation for any regular predicate over categories. What do such abbreviations mean, and how are they used?

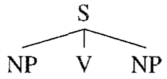
[2] The symbol ≡ connects two expressions that are defined to be equivalent; the expression in (6) can be read as: "XP is defined as the disjunction {NP|PP|VP|AP|AdvP}."

An abbreviation like VP ≡ V NP can be used to express a generalization about where a sequence of categories like V NP can appear in the grammar *without* introducing a node dominating those categories into the tree. Instead, wherever a phrase structure rule refers to the metacategory VP, the sequence of categories V NP is permitted to appear in the phrase structure tree. For example, the rule in (8) refers to the definition of VP given in (7):

(8) S $\longrightarrow$ NP VP

The phrase structure rule in (8) and the definition of VP in (7) admit the following tree:

(9)



Notably, there is no VP node in this tree.

The possibility of using a metacategory to characterize a sequence of categories in this way has an interesting impact on one of the traditionally clearest motivations for phrase structure constituency, described in Chapter 3, Section 1: generalizations governing the distribution of sequences of categories. In many theories of phrase structure, the fact that a phrase like *the dachshund* has the same syntactic distribution as a phrase like *the black dachshund* is taken as evidence that both phrases are phrase structure constituents that are dominated by an NP node; on this view, generalizations about the distribution of the two phrases are stated in terms of the distribution of an NP constituent. The use of a metacategory like S in example (9) allows for the statement of generalizations about sequences of categories in the same way. Importantly, however, the resulting phrase structure tree does not contain a constituent labeled VP; the V NP sequence does not form a phrasal unit in the constituent structure tree. Thus, although the definition of the metacategory VP in (7) allows for an economical account of the distribution of the V NP sequence, use of the metacategory VP predicts that most tests for phrase structure constituency discussed in Chapter 3 — intonation, clitic placement, and so on — will fail to reveal the presence of a VP constituent.

It is interesting to note that some (but not all) of the criteria for phrase structure constituenthood presented in Chapter 3, Section 2 are based in part on the distribution of sequences of categories. Further research may reveal more about the possibility and desirability of capturing generalizations about category distribution by means of metacategories defined over sequences of categories, rather than by assuming the existence of a phrasal constituent dominating these categories in the constituent structure tree.

1.3. ID/LP Rules

ID/LP rules were introduced by Gazdar and Pullum (1981), and independently developed within the LFG framework by Falk (1983), to allow separate statements of dominance relations and precedence relations in phrase structure rules. Dominance relations are stated in terms of Immediate Dominance or ID rules, and precedence constraints are stated in terms of Linear Precedence or LP rules. These rules allow the statement of generalizations across families of phrase structure rules: for example, that the head of a phrase of any category precedes its complements.

An ID rule expressing only dominance relations is written with commas separating the daughter nodes in the rule:

(10) VP $\longrightarrow$ V, NP

This rule states that a VP node dominates two other nodes in a tree, a V node and a NP node, but does not specify the order of V and NP. Thus, it can be regarded as an abbreviation for the two rules in (11):

(11) VP $\longrightarrow$ V NP
VP $\longrightarrow$ NP V

If we wish to specify the order, we can write a separate LP ordering constraint:

(12) VP $\longrightarrow$ V, NP V<NP

The ID phrase structure rule, combined with the linear precedence constraint V<NP stating that V must precede NP, is equivalent to the standard ordered phrase structure rule VP $\longrightarrow$ V NP. A more complicated example is given in (13):

(13) VP $\longrightarrow$ V, NP, PP V<NP, V<PP

The ID phrase structure rule requires VP to dominate three nodes, V, NP, and PP. The LP ordering constraints require V to precede both NP and PP, but do not place any constraints on the relative order of NP and PP. Thus, this rule is equivalent to the following more complex rule, in which ordering is fully specified:

(14) VP $\longrightarrow$ {V NP PP | V PP NP}

ID/LP rules are used in King's (1995) analysis of Russian. She proposes rules of the following form:

(15) CP $\longrightarrow$ XP, C′
C′ $\longrightarrow$ C, IP
IP $\longrightarrow$ XP, I′
XP<Y′, Y<XP

XP is defined as an abbreviation for any maximal phrasal category, Y' is an abbreviation of any nonmaximal category of bar level one, and Y is an abbreviation for any lexical or functional category of bar level zero. The constraint $XP<Y'$ means that the specifier XP of a phrase of category Y appears before the head Y', and the constraint $Y<XP$ means that the head Y appears before its complement phrases XP.

1.4. Regular Languages and Rule Descriptions

The ID/LP rule format allows the decomposition of a standard phrase structure rule into two aspects so that dominance constraints can be specified separately from precedence constraints. This is an example of the use of a formal device, the ID/LP rule format, to express generalizations about word order across classes of phrase structure rules: for example, whether the phrases in a language are head-initial or head-final, or whether the specifier of a phrase precedes or follows the head.

More generally, it is possible to write rules that combine descriptions of separate aspects of phrasal structure, allowing for the succinct expression of linguistic generalizations about the structure of phrases. The relevant generalizations can be expressed in terms of separate constraints that must be simultaneously satisfied. Formally, this is possible because we are allowed to specify any regular language as the right-hand side of a phrase structure rule; this means that the operations of union, intersection, and complementation can be used to combine regular expressions.[3]

We have already seen that disjunctions over various possibilities for phrase structure expansion can be specified:

(16) XP $\longrightarrow$ { X1 X2 X3 | Y1 Y2 Y3 }

This schematic rule indicates that the phrase XP dominates either the series of daughters X1 X2 X3 or the series Y1 Y2 Y3. Disjunction in a phrase structure rule corresponds to the union of two regular languages, since the union of two regular languages encompasses all of the alternatives in each language.

Intersection of two regular languages corresponds to the combination of two descriptions; each description must hold of the result:

(17) XP $\longrightarrow$ X1 & X2

This schematic rule indicates that XP dominates a sequence of categories that must satisfy the description represented by X1 as well as the description represented

[3] It is also possible to formulate regular expressions over*annotated phrase structure rules* (Section 3.1 of this chapter), with the expected interpretations. See Kaplan and Maxwell (1996) for discussion.

by X2. Formally, this just corresponds to characterizing a regular language by intersecting two regular languages, represented by X1 and X2.

To take a concrete example, we can restate ID/LP rules in this way, though the result is much more cumbersome than the standard notation. We will take the following rule as an example:

(18) $C' \longrightarrow C, IP \qquad C{<}IP$

We can think of the ID part of the rule, indicating dominance relations, as representing one aspect of the constraints on possible dominance relations; the ID rule in (18) can be expressed in an equivalent but much less revealing way as

(19) $C' \longrightarrow \{C\ IP \mid IP\ C\}$

That is, C′ dominates either C IP or IP C; the order is not determined. The LP requirement for C to precede IP can be written as follows:

(20) $\neg[\Sigma^*\ IP\ \Sigma^*\ C\ \Sigma^*]$

Σ is a metacategory that ranges over all category labels. The expression in (21) is a regular expression standing for all strings in which an IP precedes a C, together with any amount of additional material represented by Σ:

(21) $\Sigma^*\ IP\ \Sigma^*\ C\ \Sigma^*$

The complement of a regular expression is represented by the negation symbol ¬; thus, the expression in (22) stands for all strings where an IP does not precede a C:

(22) $\neg[\Sigma^*\ IP\ \Sigma^*\ C\ \Sigma^*]$

The combination of these two requirements, given in (23), is equivalent to the ID/LP rules in (18):

(23) $C' \longrightarrow \{C\ IP \mid IP\ C\}\ \&\ \neg[\Sigma^*\ IP\ \Sigma^*\ C\ \Sigma^*]$

This example illustrates two points. First, we can state constraints on constituent structure configurations by defining a sequence of daughter nodes in a phrase structure rule as a regular language. This is a powerful and general tool for the linguist to use in the description of phrase structure.

Second, this way of stating generalizations is not always the clearest or most revealing. Although the encoding of linguistic facts in terms of regular languages leads to a solid understanding of their underlying formal and computational properties, it may be preferable to devise a special notation for some common operations, since (as can be seen above, from the restatement of ID/LP rules in terms of the intersection of two regular expressions) it may not be perspicuous or revealing to state rules in terms of combinations of regular expressions: the ID/LP notation is simpler and easier to understand than the expression in (23), where two regular expressions are combined.

Kaplan and Maxwell (1996) define and discuss some operators that allow for the compact expression of particular kinds of regular languages:

THE "IGNORE" OPERATOR: Using the Ignore operator, written as a forward slash /, (Kaplan and Kay 1994), we can write a rule that allows for a category or a sequence of categories to be interspersed with the other categories:

(24) XP ⟶ X1 X2 X3 / Cat

This rule can be read as: "XP expands as X1 X2 X3, ignoring occurrences of Cat." In other words, XP must dominate X1 and X2 and X3 in that order, and may also dominate any number of occurrences of Cat at any position. This rule is equivalent to the following more complicated regular expression:

(25) XP ⟶ Cat* X1 Cat* X2 Cat* X3 Cat*

The Ignore operator rule can be used to describe the appearance of parenthetical elements, elements that can be inserted before or after any phrasal constituent (see McCawley 1982).

THE "SHUFFLE" OPERATOR: Using the Shuffle operator, represented as a comma, we can specify two different sequences of nodes, each of which appear in a particular order but which may be interspersed or "shuffled" with each other. For instance:

(26) XP ⟶ [X1 X2 X3], [Y1 Y2 Y3]

According to this rule, XP must dominate nodes labeled X1 X2 X3 and Y1 Y2 Y3. The relative order of the Xs and the Ys must be preserved, but no order is specified across these sequences. The effect of this rule is similar to an *ID/LP rule* (see Section 1.3 of this chapter) where an ordering is specified among the X daughters and among the Y daughters, but not between the Xs and the Ys; in fact, as pointed out by Ron Kaplan (p.c.), ID/LP is simply a special instance of the Shuffle operator.

The rule in (26) allows all of the following sequences, since in each case X1 precedes X2 and X3 and X2 precedes X3, and similarly for the Ys:

(27) a. X1 X2 X3 Y1 Y2 Y3
b. X1 X2 Y1 X3 Y2 Y3
c. X1 Y1 X2 Y2 X3 Y3

Any ordering between the Xs and the Ys is allowed, as long as the order X1 X2 X3 and the order Y1 Y2 Y3 are preserved. The regular expression corresponding to this rule is quite complex and will not be displayed.

The Shuffle operator is used to characterize constraints involving partial orders holding among constituents in languages with otherwise fairly free word order. It has been proposed within the Head-Driven Phrase Structure Grammar framework by Reape (1994) in his analysis of word order in German; however, the phenomena Reape analyzes using Shuffle are best analyzed within LFG in terms of functional syntactic relations, as shown by Kaplan and Zaenen (1989).

Other operators based on regular predicates can be proposed if their use simplifies linguistic description. The use of these operators assists the linguist in developing firm intuitions about linguistic phenomena; as long as the operators are definable in terms of regular predicates, no new formal power is added to the theory.

2. FUNCTIONAL CONSTRAINTS

Just as we have ways of talking about the set of permissible trees in a language, we would also like to have a way of describing acceptable f-structures. In the following, we discuss how simple constraints on functional structure are interpreted and show when these constraints are satisfied. Chapter 6 provides more information on c-structure and f-structure constraints.

Formally, an f-structure is a set of attribute-value pairs, or a function from attributes to values, as noted in Chapter 2, Section 3. The usual way of presenting an f-structure is in tabular form:[4]

(28) $\begin{bmatrix} \textsc{pred} & \text{`}\textsc{David}\text{'} \\ \textsc{num} & \textsc{sg} \end{bmatrix}$

The f-structure in (28) contains two attribute-value pairs: $\langle\textsc{pred}, \text{`}\textsc{David}\text{'}\rangle$ and $\langle\textsc{num}, \textsc{sg}\rangle$. We can place various requirements on f-structures: they may be required to contain certain attribute-value pairs or one of several possible attribute-value pairs, or they may be required not to contain certain material. The following sections explain how such constraints can be imposed.

2.1. Functional Equations

The equation in (29) specifies the f-structure named g as having an attribute NUM whose value is SG:

(29) $(g\ \textsc{num}) = \textsc{sg}$

[4] As explained in Chapter 2, Footnote 11 (page 30), what is depicted in example (28) is an f-structure, a formal object, not a constraint on f-structures.

This is a simple *functional description* or *f-description*. This f-description consists of only a single equation, but f-descriptions can in general consist of any number of such equations.

The f-structure labeled g in (30) satisfies the constraint in (29), since it has an attribute NUM with value SG:

(30) $g\begin{bmatrix}\text{NUM} & \text{SG}\end{bmatrix}$

In general, an equation requiring an attribute a of an f-structure to have a certain value v holds if (and only if) the pair consisting of the attribute a and its value v belongs to the f-structure.

There are many other f-structures (in fact, an infinite number) that also satisfy the constraint in (29). Any f-structure that has an attribute NUM with value SG as well as additional attributes and values will satisfy the constraint; for example:

(31) $$g\begin{bmatrix}\text{PRED} & \text{`DAVID'}\\ \text{GEND} & \text{MASC}\\ \text{NUM} & \text{SG}\end{bmatrix}$$

This f-structure contains the attribute NUM with value SG; it also contains additional attributes and values.

The f-structure in (30) is special in that it is the *smallest* f-structure that satisfies the constraint in (29). We call such an f-structure the *minimal solution* to the f-description in (29): it satisfies all the constraints in the f-description, and it has no additional structure that is not relevant in satisfying the constraints.

We require the f-structure for any utterance to be the smallest f-structure that satisfies all of the constraints imposed by the f-description for the utterance, and has no additional properties not mentioned in the f-description:

(32) The f-structure for an utterance is the *minimal solution* satisfying the constraints introduced by the words and phrase structure of the utterance.

The f-description for an utterance is given by annotations on the phrase structure rules and the lexical entries involved in the utterance. In Section 3 of this chapter, we will discuss how annotated phrase structure rules and lexical entries give rise to an f-description for a phrase.

Kaplan and Zaenen (1989) provide the following formal characterization of when an equation like the one in (29) holds of an f-structure:

(33) $(f\ a) = v$ holds if and only if f is an f-structure, a is a symbol, and the pair $\langle a, v\rangle \in f$.

It is also possible for an expression to involve multiple attribute names — that is, a longer path through the f-structure:

(34) (*f* SUBJ NUM) = SG

For these cases, Kaplan and Zaenen (1989) provide the following definition (see also Kaplan and Bresnan 1982):

(35) $(f\ as) \equiv ((f\ a)\ s)$ for a symbol a and a (possibly empty) string of symbols s.

$(f\ \epsilon) \equiv f$, where ϵ is the empty string.

This definition tells us that an expression like (*f* SUBJ NUM) denotes the same f-structure as ((*f* SUBJ) NUM): that is, the f-structure that appears at the end of the path SUBJ NUM in the f-structure *f* is the same f-structure that appears as the value of the attribute NUM in the f-structure (*f* SUBJ). Longer expressions are treated similarly. The second part of the definition tells us that the empty path ϵ can be ignored: the expression $(f\epsilon)$ is the same as f.

A Hindi sentence like (36) (McGregor 1972) produces the f-description given in (37) and (38) (some detail has been omitted):

(36) *Ram calegaa*
Ram go.FUTURE
'Ram will go.'

The constraints in (37) come from the lexical entries for the proper noun *Ram* and the verb *calegaa* 'will go':

(37) *Ram* (*g* PRED) = 'RAM'
(*g* CASE) = NOM
(*g* NUM) = SG

calegaa (*f* PRED) = 'GO⟨SUBJ⟩'
(*f* SUBJ CASE) = NOM
(*f* SUBJ NUM) = SG

The constraint in (38) comes from annotations on the phrase structure rules involved in the analysis of this sentence:

(38) (*f* SUBJ) = *g*

Given this equality, we can substitute *g* for (*f* SUBJ) in the constraints in (37), and we are left with the following equivalent set of constraints:

(39) $(g\ \text{PRED}) = \text{`RAM'}$
$(g\ \text{CASE}) = \text{NOM}$
$(g\ \text{NUM}) = \text{SG}$

$(f\ \text{PRED}) = \text{`GO}\langle\text{SUBJ}\rangle\text{'}$
$(g\ \text{CASE}) = \text{NOM}$
$(g\ \text{NUM}) = \text{SG}$

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

The minimal solution to this f-description — the f-structure that satisfies the f-description and contains no additional attribute-value pairs — is:

(40)
$$f\begin{bmatrix}\text{PRED} & \text{`GO}\langle\text{SUBJ}\rangle\text{'}\\ \text{SUBJ} & g\begin{bmatrix}\text{PRED} & \text{`RAM'}\\ \text{CASE} & \text{NOM}\\ \text{NUM} & \text{SG}\end{bmatrix}\end{bmatrix}$$

Notice that some of the equations in the f-description constrain the same attribute-value pairs. For example, the proper noun *Ram* requires its f-structure g to contain the attribute NUM with value SG:

(41) $(g\ \text{NUM}) = \text{SG}$

The verb *calegaa* 'go' requires its subject to contain the attribute NUM with value SG:

(42) $(f\ \text{SUBJ NUM}) = \text{SG}$

We also know that f's subject is g:

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

Thus, the two equations in (41) and (42) both require the f-structure labeled g, which is also called $(f\ \text{SUBJ})$, to contain the attribute NUM with value SG. It is common for an attribute and its value to be multiply constrained in this way; here, subject-verb agreement is enforced, since the NUM requirements of both the subject and the verb are satisfied.

We will not discuss particular methods or algorithms for solving systems of equations like the f-description in (37). Kaplan and Bresnan (1982) provide a particularly clear presentation of one algorithm for solving such sets of equations. There is a large body of work, summarized in detail by Rounds (1997), which explores the logic of feature structures and their descriptions. Some work in LFG and related frameworks discusses the operation of *unification*, first introduced in linguistic work by Kay (1979) (see also Shieber 1986). Unification is an operation that combines consistent feature structures into a new feature structure by taking the union of all the attribute-value pairs in the original structures. For instance, the unification of the two feature structures in (44a) is the feature structure in (44b):

(44) a. $$\begin{bmatrix} \text{A} & \text{B} \\ \text{C} & \text{D} \end{bmatrix} \quad \begin{bmatrix} \text{C} & \text{D} \\ \text{E} & \begin{bmatrix} \text{F} & \text{G} \end{bmatrix} \end{bmatrix}$$

b. $$\begin{bmatrix} \text{A} & \text{B} \\ \text{C} & \text{D} \\ \text{E} & \begin{bmatrix} \text{F} & \text{G} \end{bmatrix} \end{bmatrix}$$

The operation of unification on feature structures is related in a clear way to the conjunction of constraints on those structures. Consider two f-descriptions F and G, and their minimal solutions, the feature structures f and g. If the two f-descriptions F and G are consistent and are taken to describe the same f-structure, then the minimal solution to the conjunction of the two f-descriptions $F \wedge G$ is exactly the unification of the two f-structures f and g.

2.2. Semantic Forms

2.2.1. Uniqueness

The value of the PRED attribute, called a *semantic form*, behaves in a special way in terms of the constraints described here: as discussed in Chapter 2, Section 3.2, the semantic form is instantiated to a *unique* value for each use of the word with which it is associated.

As discussed by Simpson (1991, page 93), word order in Warlpiri is very free; the subject may appear in any position in the sentence. However, even though subjects may appear in either sentence-initial or sentence-final position, a sentence with two subjects (one sentence-initial and one sentence-final) is ungrammatical:[5]

(45) **wati* *ka* *parnka-mi* *karnta*
man.ABS PRES run-NONPAST woman.ABS
'The man runs the woman.'

Intuitively, the sentence is unacceptable because of the simultaneous presence of two different subjects. In slightly more formal terms, the presence of two different semantic forms for the subject of the sentence causes a clash, and the resulting f-structure is ill-formed:

[5] In our discussion of Warlpiri in Chapter 4, we discussed example (23) (page 83) in which two separate phrases contributed to the SUBJ function. In that example, unlike (45) above, the two phrases were compatible and could both appear in the same utterance, since one phrase is interpreted as the head and the other is interpreted as a modifier of the head.

(46) Ill-formed f-structure:

$$\begin{bmatrix} \text{PRED} & \text{`RUN}\langle\text{SUBJ}\rangle\text{'} \\ \text{TENSE} & \text{PRES} \\ \text{SUBJ} & g\begin{bmatrix}\text{PRED} & \text{`MAN'/`WOMAN'}\end{bmatrix}\end{bmatrix}$$

The difference between a semantic form and other atomic values is represented notationally by the presence of single quotes around the feature value:

(47) *wati* (g PRED) = 'MAN'
karnta (g PRED) = 'WOMAN'

Abstractly, a semantic form appearing in a lexical entry or phrase structure rule can be thought of as abbreviating an infinite number of distinct forms with different indices (Kaplan and Bresnan 1982). For each use of a word or rule associated with the semantic form, a new and distinct indexed form is chosen. For example, a particular instance of use of the Warlpiri word *wati* gives rise to an instantiated semantic form such as 'MAN$_1$', distinct from the semantic form 'MAN$_2$' for a different instance of use of the word *wati*.

The clitic and full pronouns of Serbo-Croatian, as discussed by Franks and King (2000), provide further evidence for the behavior of semantic forms. For pronominal objects in Serbo-Croatian, the clitic pronoun *ju* 'it' is generally used:

(48) *Mirko ju je čitao.*
Mirko it.ACC.CLITIC aux.3SG read
'Mirko read it.'

Serbo-Croatian also has a full pronominal form *nju*, used for emphasis, which does not appear in the clitic cluster:

(49) *Mirko je čitao nju.*
Mirko aux.3SG read it.ACC
'Mirko read it.'

The f-structures for these examples are largely similar; (50) gives an abbreviated f-structure for both example (48) and example (49):

(50)

$$\begin{bmatrix} \text{PRED} & \text{`READ}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`MIRKO'}\end{bmatrix} \\ \text{OBJ} & g\begin{bmatrix}\text{PRED} & \text{`PRO'}\end{bmatrix}\end{bmatrix}$$

The lexical entries for the clitic and full pronoun contain the information in (51):

(51) *ju* (g PRED) = 'PRO'
nju (g PRED) = 'PRO'

In Serbo-Croatian, the clitic pronoun and the full pronoun cannot be used in the same sentence:

(52) **Mirko ju je čitao nju.*
Mirko it.ACC.CLITIC aux.3SG read it.ACC
'Mirko read it it.'

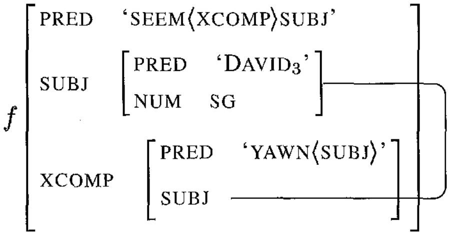
Despite the fact that the PRED value contributed by both pronominal forms is 'PRO', the sentence is ungrammatical; again, as above, a clash is produced by multiple specification of semantic forms with different indices as the value of the OBJ PRED:[6]

(53) Ill-formed f-structure:

$$\begin{bmatrix} \text{PRED} & \text{`READ}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`MIRKO'}\end{bmatrix} \\ \text{OBJ} & g\begin{bmatrix}\text{PRED} & \text{`PRO}_1\text{'/`PRO}_2\text{'}\end{bmatrix} \end{bmatrix}$$

Recall the discussion in Chapter 2, Section 3 of f-structures in which two attributes have the same value:

(54) *David seemed to yawn.*



The verb *seem* requires its SUBJ to be the same as its XCOMP's SUBJ by means of an equation like the following:

(55) $(f\ \text{SUBJ}) = (f\ \text{XCOMP SUBJ})$

The value of the PRED attribute of the SUBJ, 'DAVID', is a semantic form and therefore instantiated to a unique value for this instance of its use. In (54), we have represented this instance of the semantic form as 'DAVID$_3$'. An equally correct but less succinct way of representing the same f-structure is given in (56):

[6] In contrast to Serbo-Croatian, clitic doubling is possible in certain dialects of Spanish as well as other Slavic languages; this is because the PRED value of the clitic pronoun in these languages is *optional*, as we will see in Section 2.4 of this chapter.

(56) *David seemed to yawn.*

$$f\left[\begin{array}{ll} \text{PRED} & \text{`SEEM}\langle\text{XCOMP}\rangle\text{SUBJ'} \\ \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`DAVID3'}\\ \text{NUM} & \text{SG}\end{array}\right] \\ \text{XCOMP} & \left[\begin{array}{ll}\text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`DAVID3'}\\ \text{NUM} & \text{SG}\end{array}\right]\end{array}\right]\end{array}\right]$$

Crucially, the indices on the semantic forms of the SUBJ and the XCOMP SUBJ f-structures are the same. The f-structure in (56) is *not* the same as the one in (57), in which two different semantic forms for *David* have distinct indices:

(57)
$$\left[\begin{array}{ll} \text{PRED} & \text{`SEEM}\langle\text{XCOMP}\rangle\text{SUBJ'} \\ \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`DAVID3'}\\ \text{NUM} & \text{SG}\end{array}\right] \\ \text{XCOMP} & \left[\begin{array}{ll}\text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`DAVID4'}\\ \text{NUM} & \text{SG}\end{array}\right]\end{array}\right]\end{array}\right]$$

In general, following standard LFG practice, we will not display indices on semantic forms unless it is necessary for clarity, and two different semantic forms will be treated as distinct even if they look the same. If we want to indicate that the same semantic form appears in two different places in the f-structure, as in example (54), we will draw a line between the two occurrences.

In some cases, the value of a feature other than the PRED feature might be required to be uniquely contributed; for instance, the value of the TENSE feature is contributed by only a single form, and multiple contributions are disallowed:

(58) a. *Is David yawning?*

b. **Is David is yawning?*

An *instantiated symbol* can be used as the value of the TENSE attribute in such a situation. Like a semantic form, an instantiated symbol takes on a unique value on each occasion of its use. In general, any syntactic uniqueness requirement for a feature can be imposed by the use of an instantiated symbol as the value of that feature. Notationally, instantiated symbols are followed by an underscore; for example, to indicate that the value for the feature TENSE is the instantiated symbol PRES, we write:

(59) (f TENSE) = PRES_

2.2.2. ARGUMENT LISTS

A semantic form, unlike other values, may contain an argument list. In example (50) of this chapter, the PRED value contributed by the verb *čitao* 'read' is the complex semantic form 'READ⟨SUBJ,OBJ⟩'. As discussed in Chapter 2, Section 3.6, this f-structure is *complete* and *coherent* because the requirements specified by the semantic form for *čitao* 'read' are satisfied: the f-structure has a SUBJ and an OBJ, each containing a PRED, and there are no other governable grammatical functions in the f-structure that are not mentioned in the argument list of *čitao* 'read'.

2.3. Disjunction

An f-description can also consist of a *disjunction* of two or more descriptions. When this happens, one of the disjuncts must be satisfied for the f-description to hold.

For instance, the form *met* of the English verb *meet* is either a past tense form or a past participle:

(60) *I met/have met him.*

This is reflected in the following disjunctive f-description in the lexical entry for *met*, which says that the f-structure f for *met* must contain either the attribute-value pair ⟨TENSE, PAST⟩ or the attribute-value pair ⟨VFORM, PASTPART⟩:

(61) *met* (f PRED) = 'MEET⟨SUBJ,OBJ⟩'
{(f TENSE) = PAST | (f VFORM) = PASTPART}

There are two minimal solutions to this f-description:

(62) a.

$$f\begin{bmatrix}\text{PRED} & \text{`MEET}\langle\text{SUBJ,OBJ}\rangle\text{'}\\ \text{TENSE} & \text{PAST}\end{bmatrix}$$

b.

$$f\begin{bmatrix}\text{PRED} & \text{`MEET}\langle\text{SUBJ,OBJ}\rangle\text{'}\\ \text{VFORM} & \text{PASTPART}\end{bmatrix}$$

Each of these minimal solutions satisfies one of the disjuncts of the description.

Formally, a disjunction over descriptions is satisfied when one of the disjuncts is satisfied:

(63) Disjunction:

A disjunction $\{d_1 \mid \ldots \mid d_n\}$ over f-descriptions $d_1 \ldots d_n$ holds of an f-structure f if and only if there is some disjunct d_k, $1 \leq k \leq n$, that holds of f.

2.4. Optionality

An f-description can also be *optional*. When this happens, the f-description may but need not be satisfied.

Bresnan and Mchombo (1987) show that verbs in Chicheŵa optionally carry information about their subjects; in a Chicheŵa sentence, a subject noun phrase may be either present or absent:

(64) a. *njûchi zi-ná-lúm-a alenje*
bees SUBJ-PAST-bite-INDICATIVE hunters
'The bees bit the hunters.'

b. *zi-ná-lúm-a alenje*
SUBJ-PAST-bite-INDICATIVE hunters
'They bit the hunters.'

Bresnan and Mchombo propose that the verb *zi-ná-lúm-a* 'bit' optionally contributes an f-description constraining the value of the PRED attribute of its subject. This optional f-description is enclosed in parentheses:

(65) *zi-ná-lúm-a:* $((f\ \text{SUBJ PRED}) = \text{`PRO'})$

Since the equation $(f\ \text{SUBJ PRED}) = \text{`PRO'}$ is optional, it may but need not contribute to the minimal solution to the f-description for the sentence. If an overt subject noun phrase does not contribute its own PRED value, the f-structure for this sentence is incomplete unless this equation is satisfied, and the wellformed f-structure for the SUBJ contains the pair $\langle \text{PRED}, \text{`PRO'} \rangle$. If an overt subject noun phrase appears, the equation may not be satisfied, since the PRED value of the overt subject would produce a clash; instead, the PRED value for the SUBJ is the one specified by the subject noun phrase:

(66) a. *njûchi zi-ná-lúm-a alenje*
bees SUBJ-PAST-bite-INDICATIVE hunters
'The bees bit the hunters.'

$$f\begin{bmatrix} \text{PRED} & \text{`BITE}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`BEES'} \\ \text{NOUNCLASS} & 10 \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{`HUNTERS'} \\ \text{NOUNCLASS} & 2 \end{bmatrix} \end{bmatrix}$$

b. *zi-ná-lúm-a alenje*
SUBJ-PAST-bite-INDICATIVE hunters
'They bit the hunters.'

$$f\left[\begin{array}{ll} \text{PRED} & \text{`BITE}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`PRO'}\\ \text{NOUNCLASS} & 10\end{array}\right] \\ \text{OBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`HUNTERS'}\\ \text{NOUNCLASS} & 2\end{array}\right]\end{array}\right]$$

A similar analysis is appropriate for languages that allow *clitic doubling*. As discussed in Chapter 4, Section 3, River Plate and Peruvian dialects of Spanish allow either a clitic or a full noun phrase object to appear:

(67) a. *Juan vió a Pedro.*
Juan saw PREP Pedro
'Juan saw Pedro.'

b. *Juan lo vió.*
Juan him.ACC.SG.CLITIC saw
'Juan saw him.'

Unlike many other dialects of Spanish, in these dialects the clitic pronoun can cooccur with a full noun phrase object:

(68) *Juan lo vió a Pedro.*
Juan him.ACC.SG.CLITIC saw PREP Pedro
'Juan saw Pedro.'

We account for these facts by assuming that in the River Plate and Peruvian dialects, the PRED value contributed by the clitic pronoun *lo* is optional:

(69) *Pedro* $(f\ \text{PRED}) = \text{`PEDRO'}$
lo $((f\ \text{PRED}) = \text{`PRO'})$

A skeletal f-structure for (67a) and (68) is:

(70)
$$\left[\begin{array}{ll} \text{PRED} & \text{`SEE}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`JUAN'}\end{array}\right] \\ \text{OBJ} & f\left[\begin{array}{ll}\text{PRED} & \text{`PEDRO'}\end{array}\right]\end{array}\right]$$

When a full noun phrase object is present, the optional equation contributing the PRED value of the clitic pronoun is not satisfied; if two PREDS were present, the example would not satisfy Consistency. When there is no full noun phrase, in order to satisfy Completeness, the PRED contributed by the clitic noun phrase appears. The f-structure for example (67b) is given in (71):

(71)
$$\left[\begin{array}{ll} \mathrm{PRED} & \text{‘}\mathrm{SEE}\langle \mathrm{SUBJ,OBJ}\rangle\text{’} \\ \mathrm{SUBJ} & \left[\begin{array}{ll}\mathrm{PRED} & \text{‘}\mathrm{JUAN}\text{’}\end{array}\right] \\ \mathrm{OBJ} & f\left[\begin{array}{ll}\mathrm{PRED} & \text{‘}\mathrm{PRO}\text{’}\end{array}\right] \end{array}\right]$$

Formally, optionality of an f-description d is treated like a disjunction between d and the f-description *true*, a description that is satisfied by any f-structure.

(72) Optionality:

An f-description d optionally holds of an f-structure f if and only if the disjunction $\{d \mid \mathit{true}\}$ holds of f.

2.5. Negation

An f-description can be *negated*; when this happens, the f-description must not be satisfied.

For example, Quirk et al. (1985, 15.6) claim that it is not possible to use the complementizer *if* in the sentential complement of certain verbs:

(73) a. *I know whether/if David yawned.*

b. *You have to justify whether/*if your journey is really necessary.*

We can analyze the verb *justify* as described by Quirk et al. differently from a verb like *know* by prohibiting the value IF for the attribute COMPFORM in its COMP argument:

(74) *justify* $(f\ \mathrm{COMP}\ \mathrm{COMPFORM}) \neq \mathrm{IF}$

The f-structure in (75) satisfies this constraint:

(75) *Chris justified whether David yawned.*

$$f\left[\begin{array}{ll} \mathrm{PRED} & \text{‘}\mathrm{JUSTIFY}\langle \mathrm{SUBJ,COMP}\rangle\text{’} \\ \mathrm{SUBJ} & \left[\begin{array}{ll}\mathrm{PRED} & \text{‘}\mathrm{CHRIS}\text{’}\end{array}\right] \\ \mathrm{COMP} & \left[\begin{array}{ll} \mathrm{PRED} & \text{‘}\mathrm{YAWN}\langle \mathrm{SUBJ}\rangle\text{’} \\ \mathrm{COMPFORM} & \mathrm{WHETHER} \\ \mathrm{TENSE} & \mathrm{PAST} \\ \mathrm{SUBJ} & \left[\begin{array}{ll}\mathrm{PRED} & \text{‘}\mathrm{DAVID}\text{’}\end{array}\right] \end{array}\right] \end{array}\right]$$

However, the f-structure in (76) does not satisfy the constraint; the offending value is circled:

(76) **Chris justified if David yawned.*

$$f\left[\begin{array}{ll}
\text{PRED} & \text{`JUSTIFY}\langle\text{SUBJ,COMP}\rangle\text{'}\\
\text{SUBJ} & \left[\text{PRED}\quad \text{`CHRIS'}\right]\\
\text{COMP} & \left[\begin{array}{ll}
\text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'}\\
\text{COMPFORM} & \enclose{circle}{\text{IF}}\\
\text{TENSE} & \text{PAST}\\
\text{SUBJ} & \left[\text{PRED}\quad \text{`DAVID'}\right]
\end{array}\right]
\end{array}\right]$$

This example shows that a single equation can be negated, requiring a particular attribute-value pair not to appear. More generally, it is possible to negate not just a single equation, but an entire description. The following expressions are notationally equivalent:

(77) $(f\ a) \neq v \ \equiv\ \neg[(f\ a) = v]$

The negation of a conjunction of descriptions holds just in case at least one of the descriptions does not hold. For example, the base form for verbs in English also serves as the present tense form for all person/number combinations except third person singular. We might analyze this by means of a constraint like the one in (78), which states that f must not contain both $\langle \text{PERS}, 3\rangle$ and $\langle \text{NUM}, \text{SG}\rangle$:

(78) $$\begin{array}{l}\neg[(f\ \text{PERS}) = 3\\ \quad\ (f\ \text{NUM}) = \text{SG}]\end{array}$$

Formally, negation of f-descriptions is defined in the following way:

(79) Negation:

A negated f-description $\neg d$ holds of an f-structure f if and only if the description d does not hold of f.

2.6. Existential Constraints

An f-structure may be required to contain an attribute, but its value may be unconstrained: this kind of constraint is called an *existential constraint*. The f-structural requirement of *Completeness* (Chapter 2, Section 3.6.1) is a kind of existential constraint: Completeness requires the presence of all of the governable grammatical functions specified by a predicate, but does not place any constraints on the particular values of these functions.

Existential constraints can also be used in the analysis of relative clauses. The English relative clause must be tensed, but no particular value for the tense feature is required:

(80) a. *the man who yawned*

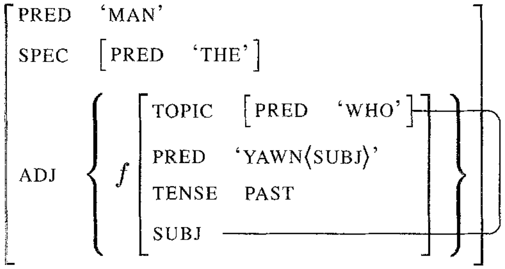
b. *the man who yawns*

c. *the man who will yawn*

d. **the man who yawning*

The f-structure for example (80a) is shown in (81). Note that the f-structure labeled f contains the attribute TENSE with value PAST:

(81) *the man who yawned*

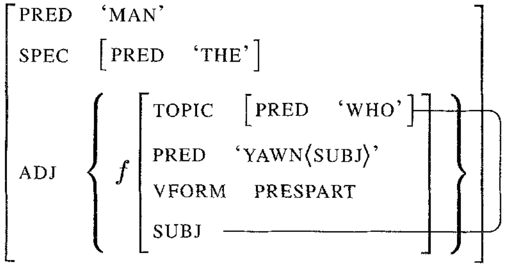


We can enforce the requirement for relative clauses to be tensed by means of a constraint like the following:

(82) (f TENSE)

This constraint requires the f-structure f to contain the attribute TENSE, but it does not constrain the value of the TENSE attribute; any value is acceptable. The f-structure in (81) satisfies this constraint. However, an f-structure like the one in (83) does not satisfy the constraint, since it has no TENSE attribute:

(83) **the man who yawning*



Formally, an existential constraint has the following interpretation:

(84) Existential constraint:

The existential constraint $(f\ a)$ holds of an f-structure f if and only if there is some value v for which the pair $\langle a, v\rangle \in f$.

2.7. Negative Existential Constraints

Just as an f-structure can be required to contain some attribute, it can be proscribed from containing some attribute: this is a *negative existential constraint*. The f-structural requirement of *Coherence* is a constraint of this kind (Chapter 2, Section 3.6.2): a grammatical function that is not mentioned in the argument list of the PRED must not appear in the f-structure.

Another use of a negative existential constraint is in the analysis of participial modifiers, as discussed by Bresnan (1982a). Such modifiers must not be tensed:

(85) a. *Scratching his head, Chris yawned.*

b. *Struck on the head, Chris slumped to the floor.*

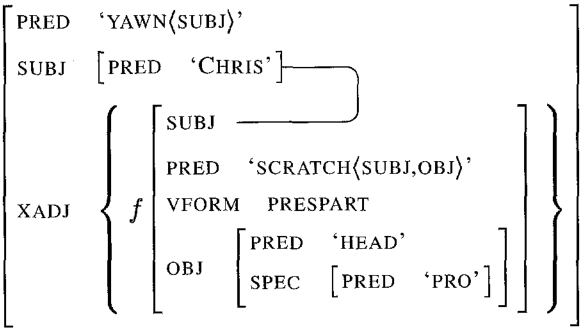
c. **Scratched/Scratches his head, Chris yawned.*

This constraint can be expressed in the following way:

(86) $\neg(f$ TENSE$)$

The constraint in (86) ensures that the f-structure f has no TENSE attribute. The constraint is satisfied in (87):

(87) *Scratching his head, Chris yawned.*



Formally, a negative existential constraint is interpreted in the following way:

(88) Negative existential constraint:

The negative existential constraint $\neg(f\ a)$ holds of an f-structure f if and only if there is no value v for which the pair $\langle a, v\rangle \in f$.

2.8. Defining and Constraining Equations

Besides defining equations like $(f\ a) = v$, LFG allows *constraining equations*, which contribute in a different way to the solution: defining equations determine the minimal solution, and constraining equations check that the minimal solution is wellformed. An example will help to show the difference between the two kinds of equations.

In English, a sentential argument need not contain the complementizer *that* when it bears the grammatical function COMP, but it must contain *that* when it is a SUBJ:

(89) a. *Chris thought that David yawned.*

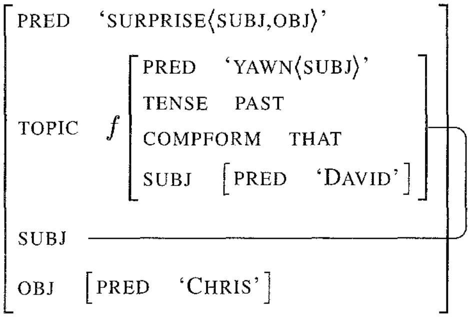
b. *Chris thought David yawned.*

c. *That David yawned surprised Chris.*

d. **David yawned surprised Chris.*

As discussed by Bresnan (1994), in example (89c) the sentential argument *that David yawned* is both a TOPIC and a SUBJ. The TOPIC/SUBJ f contains the attribute-value pair ⟨COMPFORM, THAT⟩:

(90) *That David yawned surprised Chris.*



The pair ⟨COMPFORM, THAT⟩ is not required to belong to the f-structure of COMPS; the following sentence is grammatical, and its f-structure is wellformed:

(91) *Chris thought David yawned.*

$$\begin{bmatrix} \text{PRED} & \text{`THINK}\langle\text{SUBJ,COMP}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`CHRIS'}\end{bmatrix} \\ \text{COMP} & f\begin{bmatrix} \text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`DAVID'}\end{bmatrix} \end{bmatrix} \end{bmatrix}$$

The lexical entry for the complementizer *that* contributes the following defining equation:

(92) $(f\ \text{COMPFORM}) = \text{THAT}$

Further, the phrase structure rules of English impose a requirement for sentential subjects to contain the attribute COMPFORM with value THAT in examples like (90). To accomplish this, we use the following *constraining equation*:

(93) $(f\ \text{COMPFORM}) =_c \text{THAT}$

Notationally, a constraining equation differs from a defining equation by the presence of the subscript c on the equals sign. A constraining equation is not used in determining the minimal solution to an f-description. Instead, it imposes an additional requirement on the minimal solution obtained from the defining equations in the f-description: it requires that the pair $\langle\text{COMPFORM}, \text{THAT}\rangle$ be in the minimal solution for f. Some other defining equation must specify this attribute-value pair for the final solution to be acceptable. If the corresponding defining equation were substituted for the constraining equation in (93), it would not produce the correct result; an equation like $(f\ \text{COMPFORM}) = \text{THAT}$ would just ensure that the attribute COMPFORM with its value THAT is a part of the minimal solution for f, no matter whether the word *that* is present in the sentence or not.

We can propose a formal definition for constraining equations, following Kaplan and Bresnan (1982):

(94) Constraining equation:

$(f\ a) =_c v$ holds if and only if f is an f-structure, a is a symbol, and the pair $\langle a, v\rangle$ is in the minimal solution for the defining equations in the f-description of f.

Kaplan and Bresnan (1982) provide an interesting discussion of the formal role of constraining equations and existential constraints in LFG (see also Johnson 1999; Saraswat 1999). As they note, constraining equations and existential constraints are particularly useful when the presence of a feature is associated with a marked value for the feature, with the absence of the feature indicating the unmarked value. For instance, all marked forms of a particular paradigm may be associated with a certain feature: assume, for instance, that only passive verbs have the feature PASSIVE. In such a case, if a particular voice of a verb is required, a constraining equation must be used to check for the presence or absence of the PASSIVE feature; a defining equation mentioning the PASSIVE feature would be compatible with passive verb forms as well as with active forms having no PASSIVE attribute, the wrong result.

In certain limited situations, the use of a defining equation does not produce a different result from the use of a constraining equation. For instance, suppose that all noun phrases in English are marked for number, so that they are all either singular or plural. Suppose further that the value for the number feature is not an instantiated symbol and can be specified by more than one defining equation. Then, in specifying number agreement with a noun phrase, it does not matter whether the specification involves a defining equation or a constraining equation: we know that the minimal solution to the constraints always contains a number feature with some value, since all noun phrases are specified for number. In this situation, it does not matter whether we include an additional defining specification of the feature or require that the feature and its value be present in the minimal f-structure satisfying the defining equations.

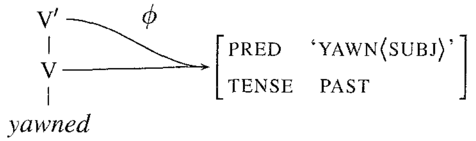
3. THE C-STRUCTURE/F-STRUCTURE CORRESPONDENCE

3.1. Annotated Phrase Structure Rules

Chapter 4 discussed universally valid correspondences between c-structure and f-structure: a c-structure head and the phrases it projects correspond to the same f-structure, for example, and the specifier of the functional categories IP and CP corresponds to a syntacticized discourse function. Here we show how these correspondences are formally stated.

Recall that the ϕ function defines a relationship between c-structure nodes and f-structures:

(95)



To make the following discussion simpler, we assume that the rule expanding V′ is:

(96) $\text{V}' \longrightarrow \text{V}$

As discussed in Chapter 4, Section 2.1, a phrase and its head correspond to the same f-structure. Here, V is the head of V′, and so V′ and V correspond to the same f-structure. We would like a way to express this fact.

We accomplish this by annotating the V node with an expression requiring the f-structure of the V to be the same as the f-structure for the V′. In general, any daughter node — that is, any node on the right-hand side of a phrase structure rule — may be annotated with constraints on the relation between its f-structure and the f-structure of the mother node. If the daughter node is the head, the f-structures must be the same. If the daughter node is a nonhead, its f-structure will bear some relation (say, the OBJ relation) to the mother's f-structure.

In order to do this, we need a notation for the following concepts:

(97) the current c-structure node ("self"): $*$
the immediately dominating node ("mother"): $\hat{*}$
the c-structure to f-structure function: ϕ

The symbol $*$ stands for the node corresponding to the rule element on which the constraint is written; note that this use of $*$ is not related to the Kleene star notation indicating that a category or attribute can be repeated zero or more times, discussed in Section 1.1 of this chapter. The symbol $\hat{*}$ stands for the node immediately dominating the $*$ node. In some LFG literature, the immediately dominating node $\hat{*}$ is represented by means of the mother function on nodes $\mathcal{M}$, as $\mathcal{M}(*)$; the two expressions $\hat{*}$ and $\mathcal{M}(*)$ are equivalent.

The function ϕ applies to a c-structure node to give the f-structure corresponding to that node. Thus, $\phi(*)$ is the f-structure corresponding to the current node, and $\phi(\hat{*})$ is the f-structure corresponding to the mother node in a rule.

To indicate that the f-structure for the V′ and for the V are the same in the rule given in (96), we can write:

(98) $$\text{V}' \longrightarrow \begin{array}{c} \text{V} \\ \phi(\hat{*}) = \phi(*) \\ \text{mother's (V}'\text{'s) f-structure} = \text{self's (V's) f-structure} \end{array}$$

A convenient abbreviation is usually used for $\phi(\hat{*})$ and $\phi(*)$:

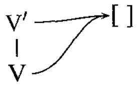
(99) $\phi(\hat{*})$ (mother's f-structure) $= \uparrow$
$\phi(*)$ (self's f-structure) $= \downarrow$

The intuition behind this notation comes from the way trees are usually represented: the up arrow $\uparrow$ points to the mother node, while $\downarrow$ points to the node itself. Using these abbreviations, we can rewrite the rule in (98) in the following more standard way:

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

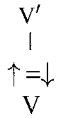
This rule represents the following phrase structure configuration:

(101)



In some LFG literature, f-structure annotations are written above the node labels of a constituent structure tree, making the intuition behind the $\uparrow$ and $\downarrow$ notation clearer; written this way, the arrows point to the appropriate phrase structure nodes:

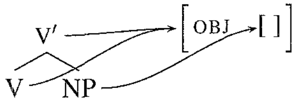
(102)



In the following, we will stick to the more common practice of writing functional annotations beneath the node label of the phrase structure tree, as in (100).

Let us turn to a slightly more complicated rule, one that describes the following c-structure and f-structure:

(103)



Complements of lexical categories like V bear nondiscourse syntactic functions; here, the NP bears the OBJ function. The following rule contains the additional information that the f-structure for the NP daughter of V′ is the value of the OBJ:

(104)
$$\begin{array}{lll} \text{V}' \longrightarrow & \text{V} & \text{NP} \\ & \phi(\hat{*}) = \phi(*) & (\phi(\hat{*})\ \text{OBJ}) = \phi(*) \\ & & \text{mother's f-structure's OBJ} = \text{self's f-structure} \end{array}$$

or, in the abbreviated notation:

(105)
$$\begin{array}{lll} \text{V}' \longrightarrow & \text{V} & \text{NP} \\ & \uparrow = \downarrow & (\uparrow \text{OBJ}) = \downarrow \end{array}$$

Some LFG work follows another abbreviatory convention according to which the annotation ↑ = ↓ is omitted when it is the only annotation on a node. According to this convention, a rule like (105) can be written as:

(106)
$$\begin{array}{lll} \text{V}' \longrightarrow & \text{V} & \text{NP} \\ & & (\uparrow \text{OBJ}) = \downarrow \end{array}$$

In this book we will try to be as explicit as possible, so we will not follow this convention of omitting equations.

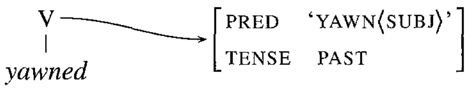
Restricting ourselves to rule annotations referring only to the f-structures corresponding to a daughter node and the mother node in a phrase structure rule makes a strong claim about the local applicability of syntactic constraints. For instance, we cannot refer to the grandmother node in a tree, or to its f-structure. This means that nonlocal syntactic relations are statable only in functional terms, not in terms of constituent structure configuration. Within the tree, only local phrase structure relations can be constrained by phrase structure rules.

In most work within LFG, only the up arrow ↑ and down arrow ↓ annotations are used in rules, which embodies an even stronger locality claim: only the functional relation between a daughter category and its mother can be stated, not the relations among the f-structures of the daughter nodes of an annotated rule. Recently, some proposals have been made for reference to the f-structures of sister nodes in a phrase structure rule. For example, Nordlinger (1998) presents an analysis of morphological composition in which crucial reference is made to the immediate left sister of a node and its f-structure; the left sister of a node can be referred to by using the symbol <∗, and its f-structure is represented as $\phi(<*)$. Similarly, the symbol ∗> is used to refer to the immediately adjacent right sister.[7] This additional expressivity expands the domain of locality slightly, but not beyond the local mother-daughter configuration that is described by the c-structure rule.

3.2. Lexical Entries

We can use the same notation in writing lexical entries that we used in annotations on phrase structure rules. For instance, assume that we would like to describe a c-structure/f-structure pair like the following:

(107)



The following lexical entry for *yawned* provides information about the f-structure corresponding to V, the immediately dominating preterminal node:

(108) *yawned* V (↑ PRED) = 'YAWN⟨SUBJ⟩'
(↑ TENSE) = PAST

This lexical entry asserts that the f-structure ↑ corresponding to the V node immediately dominating *yawned* has an attribute PRED whose value is the semantic form 'YAWN⟨SUBJ⟩', and an attribute TENSE whose value is PAST. The f-structure displayed in (107) is the minimal solution to these constraints, the smallest f-structure that satisfies the constraints.

The use of ↑ and ↓ in a lexical entry is exactly the same as their use in rules: ↑ refers to the node dominating the lexical item, and ↓ refers to the f-structure corresponding to the word itself. This can be seen more easily if we recast the lexical entry in (108) in the equivalent phrase structure rule format shown in (109):

[7] Nordlinger (1998) uses the left arrow ← to refer to the f-structure of the left sister. We prefer the use of <∗ to avoid confusion with the use of the ← symbol in *off-path constraints*: see Chapter 6, Section 1.4.

(109) $$\text{V} \quad \longrightarrow \quad \begin{array}{c} \textit{yawned} \\ (\uparrow\ \text{PRED}) = \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ (\uparrow\ \text{TENSE}) = \text{PAST} \end{array}$$

In most cases, lexical entries specify only information about $\uparrow$, the f-structure of the immediately dominating preterminal. It might sometimes be necessary or desirable to refer to properties of the f-structure $\downarrow$ corresponding to the word itself and to how the f-structure for the word relates to the f-structure for the preterminal node (see, for example, Zaenen and Kaplan 1995), and this is also possible.

We now have the notational equipment to do a complete analysis of a sentence like *David yawned.* For clarity, in the following example the rules and lexical entries have been considerably simplified; for example, the phrase structure rules expanding VP and NP are clearly too impoverished to account for very many constructions in English.

We assume the following annotated phrase structure rules:[8]

(110) $$\begin{array}{lll} \text{IP} & \longrightarrow & \begin{pmatrix} \text{NP} \\ (\uparrow\ \text{SUBJ}) = \downarrow \end{pmatrix} \begin{pmatrix} \text{I}' \\ \uparrow = \downarrow \end{pmatrix} \\ \text{I}' & \longrightarrow & \begin{pmatrix} \text{I} \\ \uparrow = \downarrow \end{pmatrix} \begin{pmatrix} \text{VP} \\ \uparrow = \downarrow \end{pmatrix} \\ \text{VP} & \longrightarrow & \begin{pmatrix} \text{V} \\ \uparrow = \downarrow \end{pmatrix} \\ \text{NP} & \longrightarrow & \begin{pmatrix} \text{N} \\ \uparrow = \downarrow \end{pmatrix} \end{array}$$

In English, the specifier of IP is associated with the syntacticized discourse function SUBJ. Other daughter nodes in the rules in (110) are heads or complements of functional categories, and are associated with the annotation $\uparrow=\downarrow$, requiring that they correspond to the same f-structure as the mother node.

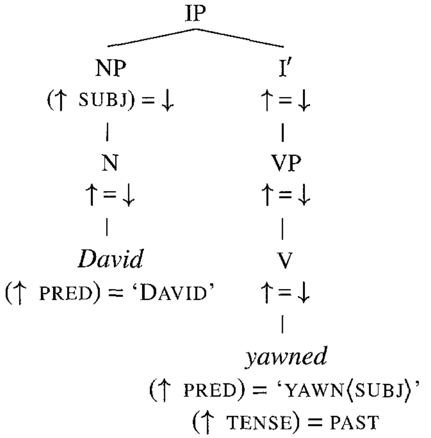
We will also make use of the following lexical entries:

(111) $$\begin{array}{lll} \textit{yawned} & \text{V} & (\uparrow\ \text{PRED}) = \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ & & (\uparrow\ \text{TENSE}) = \text{PAST} \\ \textit{David} & \text{N} & (\uparrow\ \text{PRED}) = \text{`DAVID'} \end{array}$$

These rules and lexical entries admit the following tree, with as yet uninstantiated variables $\uparrow$ and $\downarrow$ over f-structures:

[8] Recall from Chapter 3, Section 4.4 that all rule elements are only optionally present.

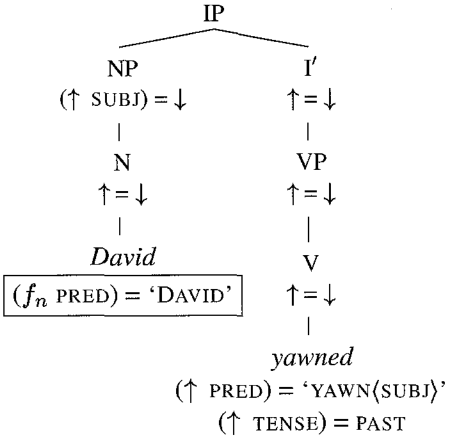
(112) *David yawned.*



The next task is to instantiate the ↑ and ↓ metavariables to the f-structures that they stand for in this case. It will be useful to have names for the f-structures corresponding to each node; we will give the name f_v to the f-structure corresponding to the node labeled V, f_{vp} to the f-structure for the node labeled VP, and so on.

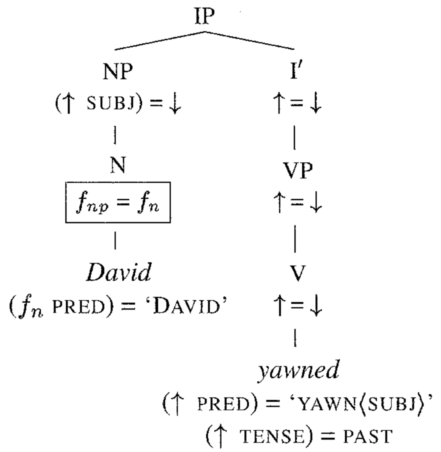
We begin with the information contributed by the lexical entry for *David*. The f-structure variable ↑ in the annotation (↑ PRED) = 'DAVID' for *David* refers to f_n, the f-structure of the N node immediately dominating the leaf node *David*, and so we replace ↑ in that expression with f_n.

(113)



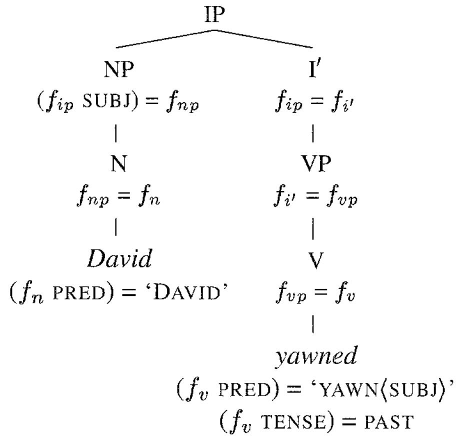
Let us now consider the N node. Its annotation is $\uparrow = \downarrow$, meaning that the f-structure f_{np} corresponding to its mother node NP is the same as f_n, the f-structure for the N node:

(114)



In a similar way, we replace the $\uparrow$ and $\downarrow$ nodes in the rest of the tree with the names of the f-structures they refer to:

(115)



We now have an instantiated f-description of the f-structure for this sentence:

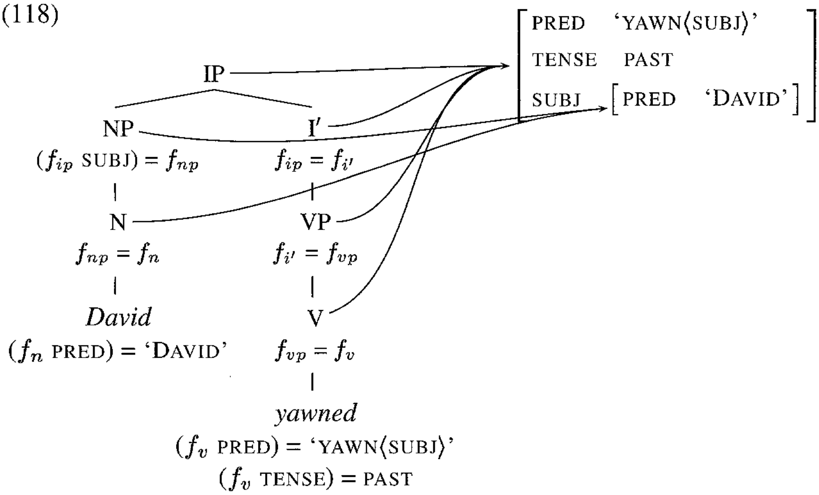
(116) $$\begin{array}{l} (f_{ip}\ \textsc{subj}) = f_{np} \\ f_{np} = f_n \\ (f_n\ \textsc{pred}) = \text{`}\textsc{David}\text{'} \\ f_{ip} = f_{i'} \\ f_{i'} = f_{vp} \\ f_{vp} = f_v \\ (f_v\ \textsc{pred}) = \text{`}\textsc{yawn}\langle\textsc{subj}\rangle\text{'} \\ (f_v\ \textsc{tense}) = \textsc{past} \end{array}$$

According to these constraints, f_{ip}, $f_{i'}$, f_{vp}, and f_v all name the same f-structure, which has three attributes, PRED, TENSE, and SUBJ. The SUBJ of this f-structure is f_{np}, which is also called f_n. The f-structure for this sentence is the minimal solution to these constraints, the f-structure that satisfies all of these constraints and contains no extra structure not mentioned in the constraints:

(117) $$f_{ip}, f_{i'}, f_{vp}, f_v \left[\begin{array}{ll} \textsc{pred} & \text{`}\textsc{yawn}\langle\textsc{subj}\rangle\text{'} \\ \textsc{tense} & \textsc{past} \\ \textsc{subj} & f_{np}, f_n \left[\begin{array}{ll} \textsc{pred} & \text{`}\textsc{David}\text{'} \end{array}\right] \end{array}\right]$$

We have now deduced that the sentence *David yawned* has the following annotated c-structure and f-structure:

(118)



To summarize: an f-structure is admitted in correspondence with a particular constituent structure tree if the annotations on the phrase structure rules and the lexical items admit the pairing of that tree with that f-structure, and if the f-structure is the *minimal solution* — the smallest f-structure — that satisfies the constraints in the annotations and the lexical entries.

4. VARIATION IN GRAMMATICAL FUNCTION ENCODING

Grammatical functions are encoded in different ways in different languages, and languages may employ mixed or multiple strategies for grammatical function encoding. These typological differences are reflected in the constraints associated with lexical items and phrase structure rules.

In some languages, those that are often called "configurational," the grammatical function of a phrase is determined by its constituent structure position. Languages of this type make use of *configurational encoding* (Bresnan 1982a): phrase structure positions are associated with particular grammatical functions by means of annotations on phrase structure rules.

In other languages, grammatical function is encoded by means of morphological marking, and there may be no uniform position where a particular grammatical function must appear. This is what Bresnan (1982a) calls *nonconfigurational encoding*: an association between morphological marking and syntactic function. Languages may tend to employ one of these types of encoding more heavily, but there are many cases in which a single language employs both types. For instance, in English, the OBJ grammatical functions are encoded configurationally, not by means of morphological marking. In contrast, the oblique functions are encoded nonconfigurationally. Below, we will examine languages making use of a combination of these strategies.

In an important typological study, Nichols (1986) shows that some languages are *head marking* and some are *dependent marking*: in other words, the surface indication of grammatical function can appear either on the argument of a predicate (dependent marking) or on the predicate itself (head marking). Often, this surface indication involves nonconfigurational encoding, with morphological marking of grammatical function on either the head or the dependent. In fact, though, configurational languages can be said to exhibit a type of dependent marking, since a surface syntactic property of the dependent — its constituent structure position — indicates its grammatical function. We will see examples of both head-marking and dependent-marking languages below.

The following sections contain brief sketches of several different languages, as an illustration of the kind of variability that LFG predicts. These thumbnail sketches are not intended as complete analyses of these languages; only enough detail is provided so that the broad outline of their typological properties becomes evident. For very interesting discussions of nonconfigurationality, head marking, and dependent marking in a variety of languages, see Nordlinger (1998) and Bresnan (2001b).

4.1. English

English is a language in which the term grammatical functions SUBJ, OBJ, and OBJ_θ are primarily encoded configurationally. This means that phrase structure rules contain specifications of particular grammatical functions: as discussed in Chapter 4, Section 2, the specifier position of IP is filled by the subject, and the object appears as the first nominal complement of V. These principles of mapping between c-structure and f-structure configurations are reflected in the annotations on the rules given in (119): heads of phrases bear the annotation $\uparrow = \downarrow$, ensuring that a phrase and its head correspond to the same f-structure; the specifier of the functional category IP bears the annotation $(\uparrow \text{SUBJ}) = \downarrow$, ensuring that it is associated with the syntacticized discourse function SUBJ; the VP complement of the functional category I is an f-structure co-head, bearing the annotation $\uparrow = \downarrow$; and the complement of the lexical category V bears the annotation $(\uparrow \text{OBJ}) = \downarrow$, ensuring that it is associated with the non-discourse syntactic function OBJ.

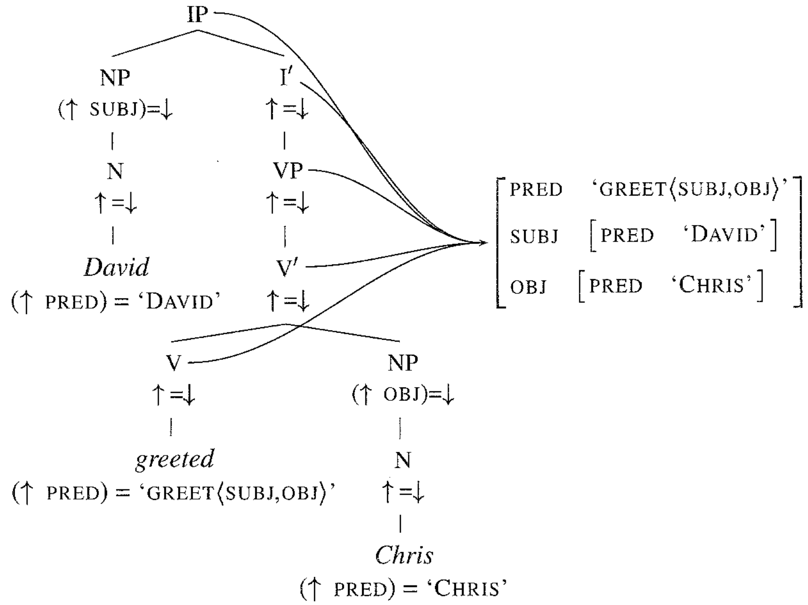
(119)
$$\begin{array}{lll}
\mathrm{IP} & \longrightarrow & \begin{pmatrix} \mathrm{NP} \\ (\uparrow \text{SUBJ}) = \downarrow \end{pmatrix} \begin{pmatrix} \mathrm{I}' \\ \uparrow = \downarrow \end{pmatrix} \\
\mathrm{I}' & \longrightarrow & \begin{pmatrix} \mathrm{I} \\ \uparrow = \downarrow \end{pmatrix} \begin{pmatrix} \mathrm{VP} \\ \uparrow = \downarrow \end{pmatrix} \\
\mathrm{VP} & \longrightarrow & \begin{pmatrix} \mathrm{V}' \\ \uparrow = \downarrow \end{pmatrix} \\
\mathrm{V}' & \longrightarrow & \begin{pmatrix} \mathrm{V} \\ \uparrow = \downarrow \end{pmatrix} \begin{pmatrix} \mathrm{NP} \\ (\uparrow \text{OBJ}) = \downarrow \end{pmatrix}
\end{array}$$

Predicates specify a list of the governable grammatical functions that they require:

(120) *greeted* $(\uparrow \text{PRED}) = \text{`GREET}\langle\text{SUBJ,OBJ}\rangle\text{'}$

The c-structure and f-structure for the sentence *David greeted Chris* are given in (121), with the relation between the clausal head c-structure nodes and the main f-structure indicated by arrows:

(121) *David greeted Chris.*



The requirement for the presence of a SUBJ and an OBJ is lexically specified by the verb, and the grammatical function of each argument is determined by its phrase structure position.

4.2. Warlpiri

Warlpiri is typologically quite different from English. As discussed in Chapter 4, Section 2.5, Warlpiri makes use of the exocentric category S; like many languages with S, phrase structure configuration does not determine the grammatical function of a Warlpiri phrase. Instead, grammatical function is determined by morphological casemarking on the argument phrase (Simpson 1991; Austin and Bresnan 1996). The phrase structure rules of Warlpiri[9] make use of the abbreviation GF, which represents a disjunction of all grammatical functions:

(122) $\text{GF} \equiv \{\text{SUBJ} \mid \text{OBJ} \mid \text{OBJ}_\theta \mid \text{COMP} \mid \text{XCOMP} \mid \text{OBL} \mid \text{ADJ} \mid \text{XADJ}\}$

[9] As for English, we have provided simplified phrase structure rules for Warlpiri for the purposes of this discussion.

(123)
$$\begin{array}{lll} \text{IP} & \longrightarrow & \begin{pmatrix} \text{NP} \\ (\uparrow \text{ FOCUS}) = \downarrow \\ (\uparrow \text{ GF}) = \downarrow \end{pmatrix} \begin{pmatrix} \text{I}' \\ \uparrow = \downarrow \end{pmatrix} \\ \text{I}' & \longrightarrow & \begin{pmatrix} \text{I} \\ \uparrow = \downarrow \end{pmatrix} \begin{pmatrix} \text{S} \\ \uparrow = \downarrow \end{pmatrix} \\ \text{S} & \longrightarrow & \left\{ \begin{array}{c} \text{NP} \\ (\uparrow \text{ GF}) = \downarrow \end{array} \middle| \begin{array}{c} \text{V} \\ \uparrow = \downarrow \end{array} \right\}^{*} \end{array}$$

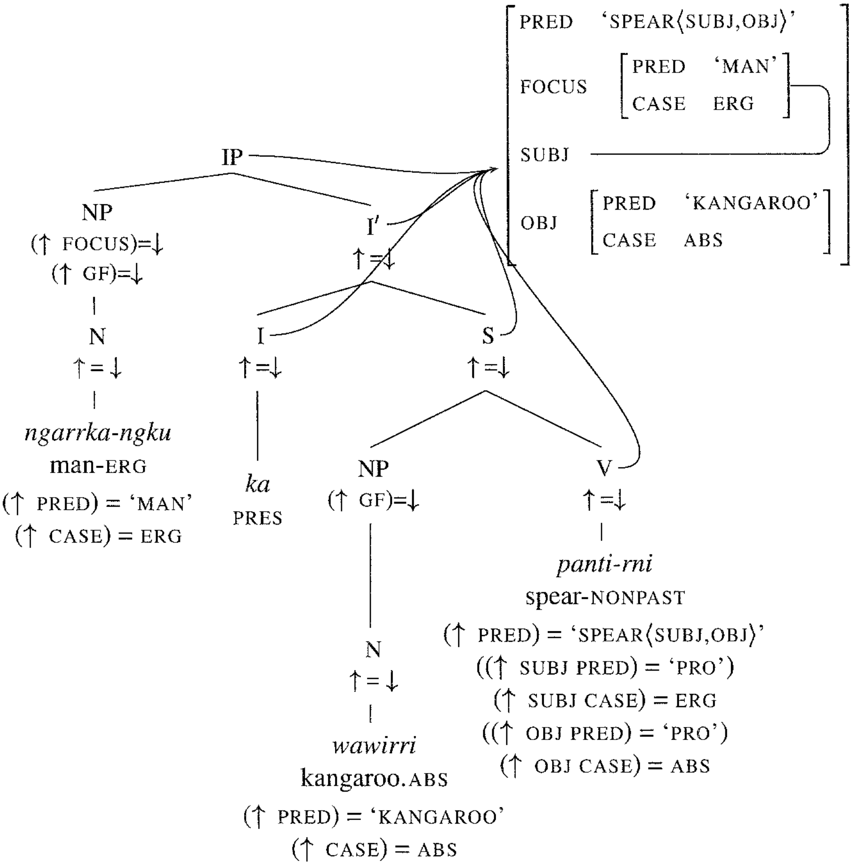
In Warlpiri, the specifier position of IP is associated with the syntacticized discourse function FOCUS (Austin and Bresnan 1996). The FOCUS phrase also plays a grammatical function in the sentence, as required by the annotation $(\uparrow \text{ GF}){=}\downarrow$. The NP daughters of S are also annotated with the equation $(\uparrow \text{ GF}){=}\downarrow$, indicating that a noun phrase with any grammatical function can appear there. See Chapter 6, Section 1.1 for more discussion of the use of the symbol GF.

In contrast to English, the Warlpiri verb specifies a great deal of information about its arguments. The case of each argument is specified, and additionally an optional PRED value for each argument is provided. As described in Section 2.4 of this chapter, this allows for the absence of overt phrasal arguments ("pro-drop"): a Warlpiri sentence may consist simply of a verb, with no overt subject or object phrases present at c-structure. In this case, the PRED values of the arguments of the verb come from the verb's specifications.

(124) *panti-rni* V $(\uparrow \text{ PRED}) = \text{'SPEAR}\langle\text{SUBJ,OBJ}\rangle\text{'}$
$((\uparrow \text{ SUBJ PRED}) = \text{'PRO'})$
$(\uparrow \text{ SUBJ CASE}) = \text{ERG}$
$((\uparrow \text{ OBJ PRED}) = \text{'PRO'})$
$(\uparrow \text{ OBJ CASE}) = \text{ABS}$

The c-structure and f-structure for the sentence *ngarrka-ngku ka wawirri panti-rni* 'the man is spearing the kangaroo' are:

(125) *ngarrka-ngku ka wawirri panti-rni*
man-ERG PRES kangaroo.ABS spear-NONPAST
'The man is spearing the kangaroo.'



The verb requires its subject to be in ergative case; phrase structural annotations allow the phrase *ngarrka-ngku* 'man' to bear any grammatical function GF, but only the SUBJ grammatical function is compatible with the ergative casemarking requirements imposed by the verb. Similarly, absolutive casemarking requires the phrase *wawirri* 'kangaroo' to bear the OBJ function.

The verb also provides optional 'PRO' values, enclosed in parentheses, for the PRED of its subject and object. These values do not appear in the final f-structure, since the overt subject and object noun phrases *ngarrka-ngku* 'man' and *wawirri* 'kangaroo' are present and contribute their PRED values to the final f-structure.

If these phrases did not appear, the 'PRO' value optionally provided by the verb would appear as the value of the PRED of these arguments.

4.3. Chicheŵa

Chicheŵa is typologically different from both English and Warlpiri, and illustrates an interesting combination of configurational and nonconfigurational characteristics (Bresnan and Mchombo 1987; Bresnan 2001b). The relevant phrase structure rules for Chicheŵa are:

(126)
$$\begin{array}{lll}
\mathrm{S} & \longrightarrow & \begin{pmatrix} \mathrm{NP} \\ (\uparrow \textsc{subj}) = \downarrow \end{pmatrix}, \begin{pmatrix} \mathrm{NP} \\ (\uparrow \textsc{topic}) = \downarrow \end{pmatrix}, \begin{pmatrix} \mathrm{VP} \\ \uparrow = \downarrow \end{pmatrix} \\
\mathrm{VP} & \longrightarrow & \begin{pmatrix} \mathrm{V}' \\ \uparrow = \downarrow \end{pmatrix} \\
\mathrm{V}' & \longrightarrow & \begin{pmatrix} \mathrm{V} \\ \uparrow = \downarrow \end{pmatrix} \begin{pmatrix} \mathrm{NP} \\ (\uparrow \textsc{obj}) = \downarrow \end{pmatrix}
\end{array}$$

These rules show that grammatical functions in Chicheŵa are specified configurationally to some extent, though not in the same way as in English. Chicheŵa makes use of the exocentric category S, and the subject, the topic, or both may appear as daughters of S, before or after the VP (the first rule in (126) is an unordered ID rule, as described in Section 1.3 of this chapter). The complement of the lexical category V is the nondiscourse syntactic function OBJ, as indicated by the equation (↑ OBJ) = ↓.

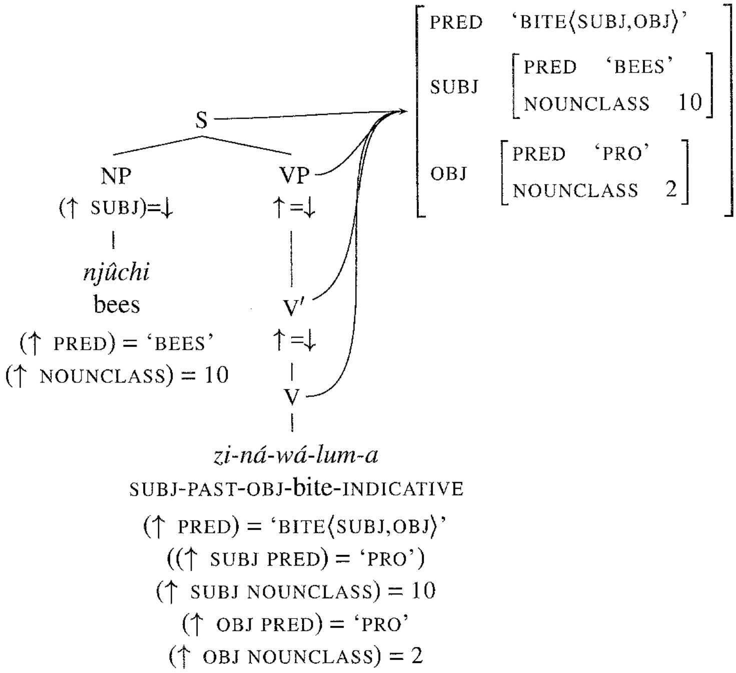
The lexical entry for the Chicheŵa transitive verb *zi-ná-wá-lum-a* 'bite' is given in (127):

(127) *zi-ná-wá-lum-a* V (↑ PRED) = 'BITE⟨SUBJ,OBJ⟩'
((↑ SUBJ PRED) = 'PRO')
(↑ SUBJ NOUNCLASS) = 10
(↑ OBJ PRED) = 'PRO'
(↑ OBJ NOUNCLASS) = 2

Unlike the English verb, and like the Warlpiri verb, this verb contains an optional 'PRO' value for the PRED of its subject; this means that an overt SUBJ phrase may but need not appear. The verb also carries information about the noun class of its arguments: Chicheŵa, like many Bantu languages, has a complex noun class system, and the prefix *zi-* indicates that the SUBJ belongs to noun class 10.

The OBJ is treated differently from the SUBJ. As Bresnan and Mchombo (1987) show, this verb contains an incorporated pronominal object *wá*. This means that the equation specifying the OBJ PRED is not optional. The c-structure and f-structure for the sentence *njûchi zi-ná-wá-lum-a* 'the bees bit them' are displayed in (128):

(128) *njûchi zi-ná-wá-lum-a*
bees SUBJ-PAST-OBJ-bite-INDICATIVE
'The bees bit them.'

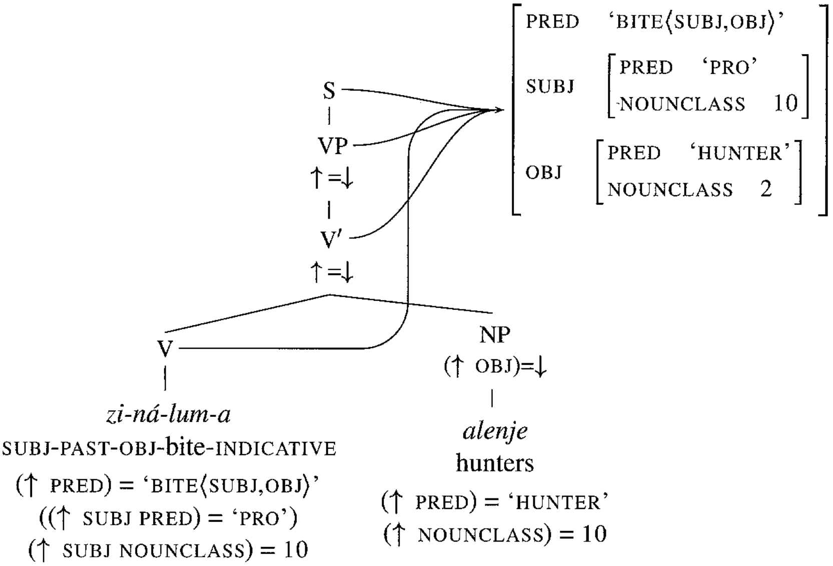


When the incorporated object pronoun *wá* does not appear, the sentence is incomplete unless an overt noun phrase is present. The lexical entry for the verb *zi-ná-lum-a* 'bite', with no incorporated OBJ pronoun, is:

(129) *zi-ná-lum-a* V (↑ PRED) = 'BITE⟨SUBJ,OBJ⟩'
((↑ SUBJ PRED) = 'PRO')
(↑ SUBJ NOUNCLASS) = 10

In the following Chicheŵa sentence, there is no overt SUBJ noun phrase, and the 'PRO' value of the PRED of the subject noun phrase is provided by the verb. An overt OBJ noun phrase, *alenje* 'hunters', also appears; if there were no overt OBJ noun phrase, the sentence would be incomplete and therefore ungrammatical. The grammatical function of *alenje* is determined by the phrase structure configuration in which it appears:

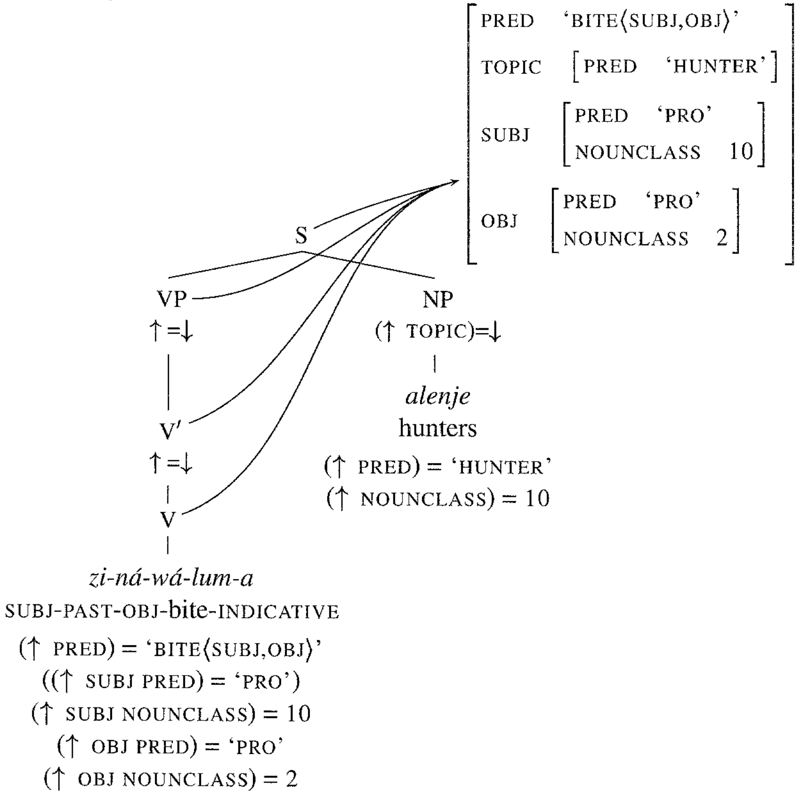
(130) *zi-ná-lúm-a* *alenje*
SUBJ-PAST-bite-INDICATIVE hunters
'They bit the hunters.'



It is also possible for the incorporated object pronoun to be anaphorically linked to an overt TOPIC noun phrase, a semantic relationship that is not indicated in the functional structure; see Bresnan (2001b, Chapter 4) for discussion. In this case, the incorporated OBJ pronoun *wá* appears, as shown in example (131) (page 133). Example (131) is different from (130) in that the phrase *alenje* 'hunters' appears not in canonical OBJ position, but in the c-structure position associated with the topic (Bresnan and Mchombo 1987). This TOPIC phrase is anaphorically linked to the incorporated OBJ pronoun, as indicated by the subscript *i* indexes in the gloss.

As these examples illustrate, the pronominal typology predicted by LFG is richer than the one proposed by Jelinek (1984), who hypothesizes that all nonconfigurational languages should be analyzed as pronominal-incorporating, as we have analyzed the Chicheŵa incorporated object pronoun. Dahlstrom (1986a) shows that this simple proposal does not provide an adequate account of the facts in Fox; Bresnan and Mchombo (1987), Austin and Bresnan (1996), Nordlinger (1998), and Toivonen (2000) also show that an adequate analysis of the phrasal and functional structure of many languages requires a distinction between incorporated pronouns (like the Chicheŵa incorporated object pronoun *wá*), agreement markers, and forms which, like the Chicheŵa subject marker, are ambiguous between the two.

(131) *zi-ná-wá-lúm-a* *alenje*
SUBJ-PAST-OBJ-bite-INDICATIVE hunters
'They bit them$_i$, the hunters$_i$.'



4.4. Bulgarian

Bulgarian is unusual in combining relatively free word order with a lack of nominal inflection (Rudin 1985): only pronominal forms show casemarking. In some cases, the subject can be identified as the argument that agrees with the verb; additionally, Bulgarian allows clitic doubling, so that the case, gender, and number of clitic-doubled arguments are specified. In other cases, however, these clues do not serve to disambiguate the grammatical functions of the argument phrases, and a phrase may be associated with any of the grammatical functions selected by the predicate. In such cases, only contextual information and world knowledge help in determining the intended structure.

As noted in Chapter 4, the specifier of IP is the FOCUS position in Bulgarian. The relevant phrase structure rules for Bulgarian are:

(132)
$$\begin{array}{lll}
\mathrm{IP} & \longrightarrow & \begin{pmatrix} \mathrm{NP} \\ (\uparrow \textsc{focus}) = \downarrow \\ (\uparrow \textsc{gf}) = \downarrow \end{pmatrix} \begin{pmatrix} \mathrm{I'} \\ \uparrow = \downarrow \end{pmatrix} \\
\mathrm{I'} & \longrightarrow & \begin{pmatrix} \mathrm{I} \\ \uparrow = \downarrow \end{pmatrix} \begin{pmatrix} \mathrm{S} \\ \uparrow = \downarrow \end{pmatrix} \\
\mathrm{I} & \longrightarrow & \begin{pmatrix} \mathrm{Cl} \\ (\uparrow \textsc{obj}) = \downarrow \end{pmatrix} \begin{pmatrix} \mathrm{I} \\ \uparrow = \downarrow \end{pmatrix} \\
\mathrm{S} & \longrightarrow & \left\{ \begin{matrix} \mathrm{NP} \\ (\uparrow \textsc{gf}) = \downarrow \end{matrix} \;\middle|\; \begin{matrix} \mathrm{V} \\ \uparrow = \downarrow \end{matrix} \right\}^{*}
\end{array}$$

At this rough level of detail, these rules are remarkably similar to the rules for Warlpiri, which also allows relatively free word order.

A verb such as *vidja* 'saw' has a lexical entry like the following:[10]

(133) *vidja* V (↑ PRED) = 'SEE⟨SUBJ,OBJ⟩'
((↑ SUBJ PRED) = 'PRO')
(↑ SUBJ PERS) = 3
(↑ SUBJ NUM) = SG
(↑ SUBJ CASE) = NOM
(↑ OBJ CASE) = ACC

As Rudin (1985) shows, a Bulgarian clause can appear without an overt subject, as in Chicheŵa and Warlpiri: the verb contains an optional equation specifying a pronominal value for the PRED of its subject. If the Bulgarian verb is transitive, either an overt object phrase or the object clitic pronoun must appear, since (unlike Warlpiri) the verb does not specify a PRED value for its object.

The noun *knigata* 'the book' and the proper noun *Georgi* are unmarked for case; each of them is compatible with either nominative or accusative case. In example (135) (page 136), the metavariable GF, which represents any grammatical function, is arbitrarily instantiated to SUBJ for *Georgi* and OBJ for *knigata* 'the book'. As Rudin (1985) notes, it is only world knowledge that requires the interpretation of *knigata* 'the book' as the object of the verb *vidja* 'saw', and *Georgi* as its subject. Neither phrase structure position, casemarking, nor agreement requirements serve to disambiguate the syntactic role of these arguments.

Example (136) (page 137) differs from (135) in that the FOCUS noun phrase *decata* 'the children' is plural. Thus, since the verb shows third person singular agreement with its SUBJ, the only available analysis is the one in which the third person singular phrase *Georgi* is the subject.

[10] The representations of the values for the featuresPERS, GEND, and CASE have been simplified for the purposes of this example; see Chapter 13 for more discussion of these features.

The presence of a doubled clitic object pronoun can also help to disambiguate a potentially ambiguous sentence; example (137) (page 138) is unambiguous. The lexical entry for the feminine singular accusative clitic pronoun *ja* is

(134) *ja* ((↑ PRED) = 'PRO')
(↑ PERS) = 3
(↑ NUM) = SG
(↑ GEND) = FEM
(↑ CASE) = ACC

If there is no full object noun phrase, the PRED value for the OBJ function is given by the object clitic pronoun phrase. Since the PRED of the clitic pronoun is optional (as in Spanish; see Section 2.4 of this chapter) the presence of *ja* is also compatible with the OBJ being filled by the feminine phrase *Marija* (but not the masculine phrase *Georgi*, since that would produce a clash in GEND values).

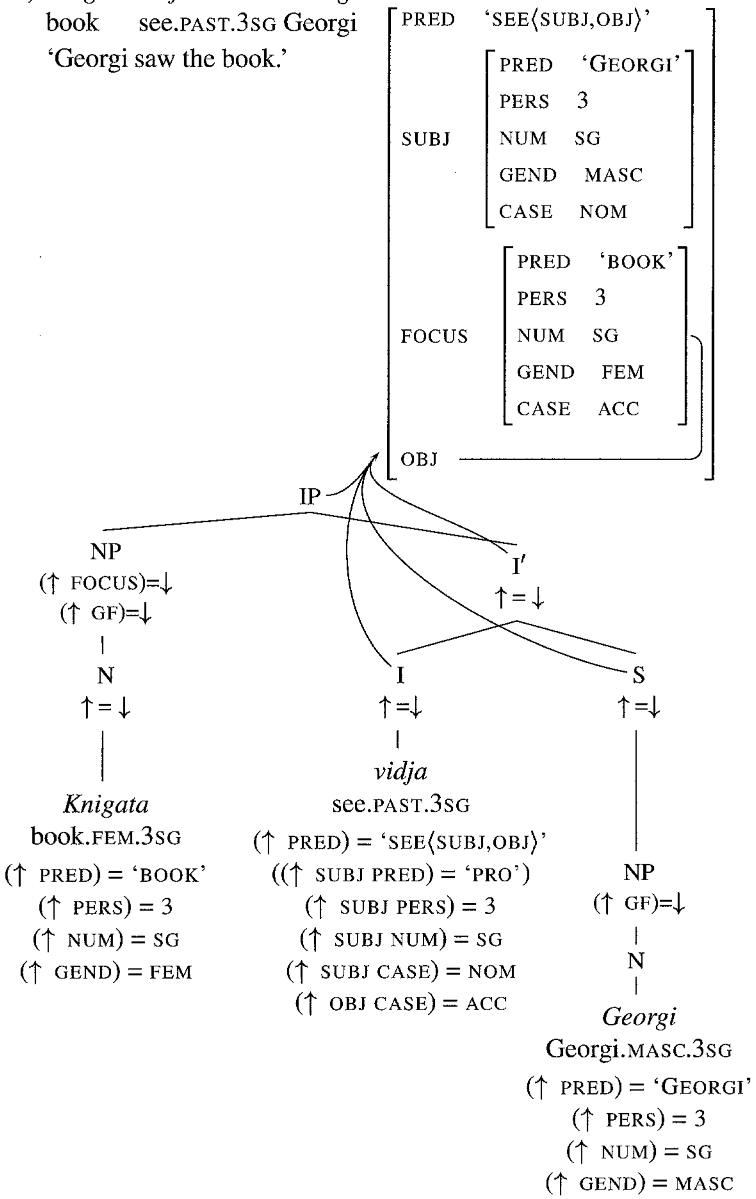
The English, Warlpiri, Chicheŵa, and Bulgarian examples presented in this section attest both to the diversity of expression found crosslinguistically and to the basic underlying unity of structure at a more abstract syntactic level.

5. FURTHER READING AND RELATED ISSUES

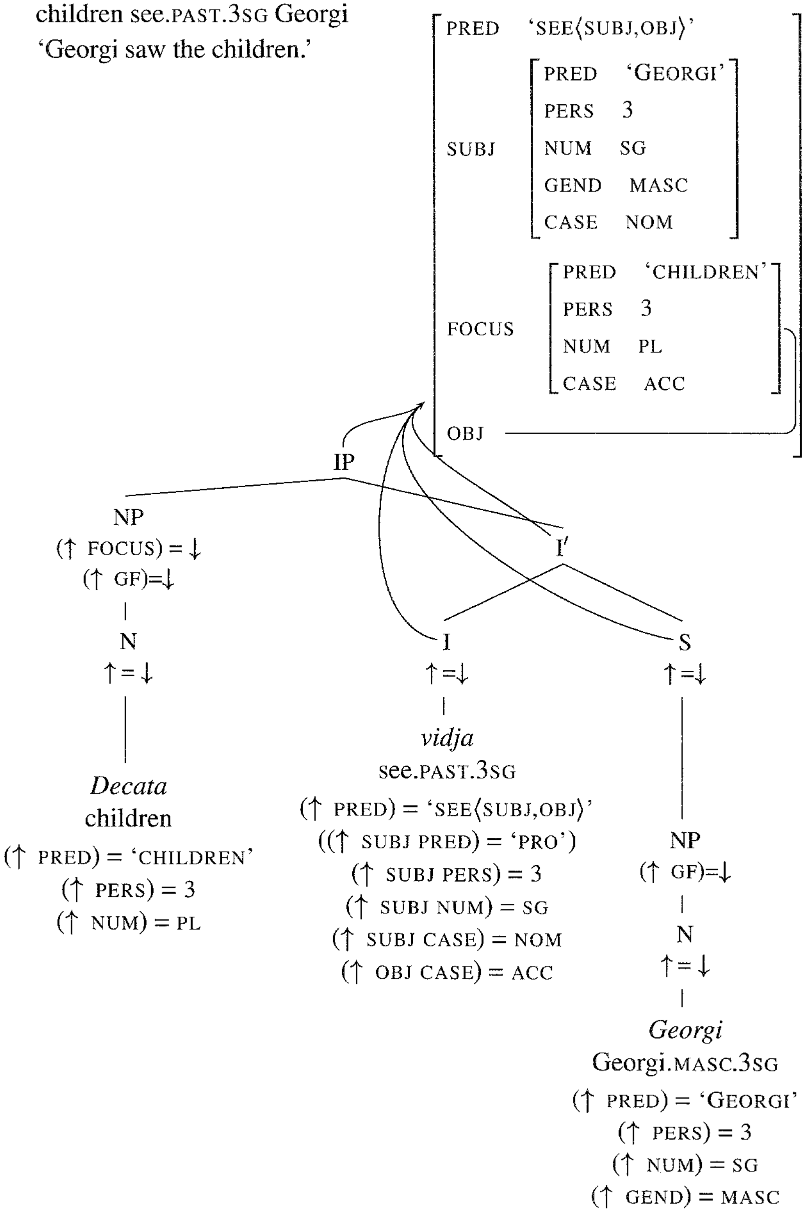
The formal tools and notational conventions of LFG presented in this and the next chapter are discussed in detail by Kaplan and Maxwell (1996).

In the preceding, we have assumed that f-structures and lexical entries are fully specified, and we have not appealed to any markedness or blocking principles that would lead us to choose the least marked form or the most specific compatible form in any given instance. This issue was discussed in detail by Andrews (1990b), who proposes the *Morphological Blocking Principle*, requiring that the most specific compatible lexical item must be chosen. Much recent LFG research discusses and relies on the Morphological Blocking Principle, and Bresnan (2000) proposes a recasting of the principle in Optimality-theoretic terms; Optimality-theoretic LFG analyses are briefly discussed in Chapter 15, Section 3.

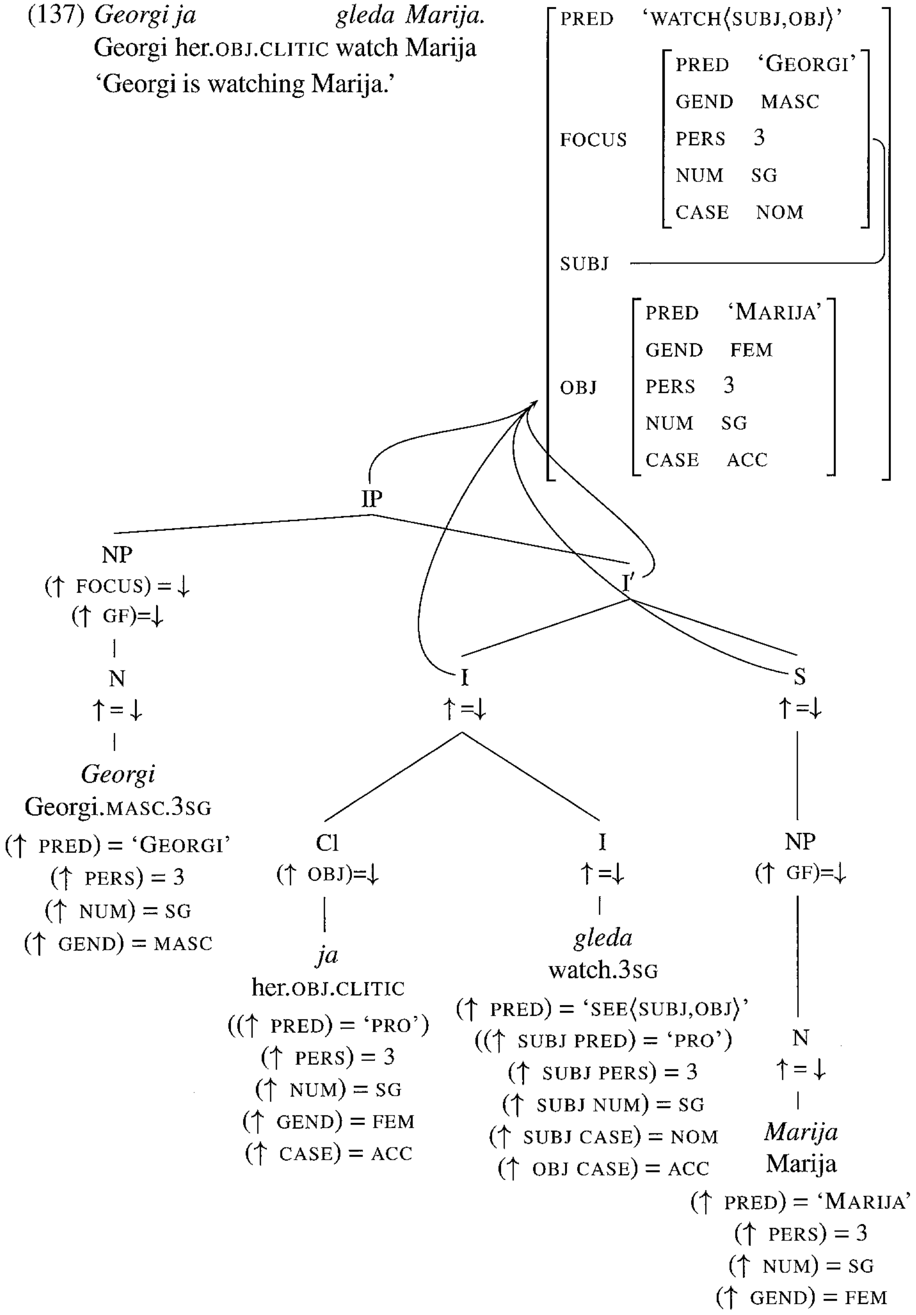
(135) *Knigata vidja Georgi.*
book see.PAST.3SG Georgi
'Georgi saw the book.'



(136) *Decata vidja Georgi.*
children see.PAST.3SG Georgi
'Georgi saw the children.'



(137) *Georgi ja gleda Marija.*
Georgi her.OBJ.CLITIC watch Marija
'Georgi is watching Marija.'



6

SYNTACTIC RELATIONS AND SYNTACTIC CONSTRAINTS

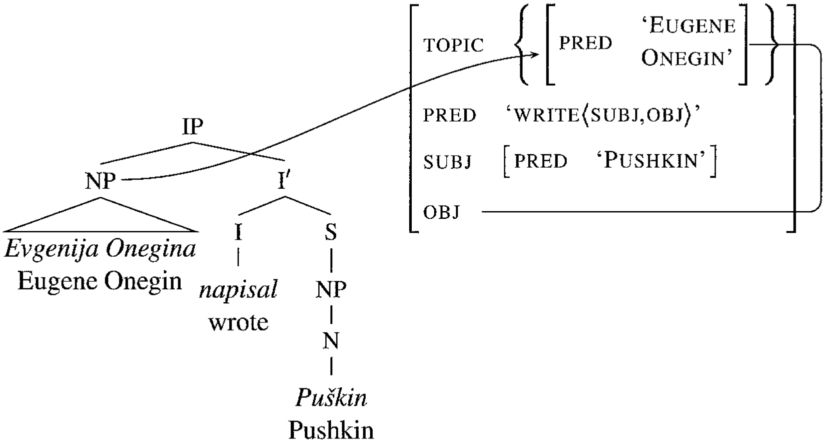
Chapter 5 discussed ways of talking about and constraining constituent structures and functional structures. This chapter continues that thread, introducing additional relations and constraints on structures. For most readers, this chapter will best serve as a reference to be consulted for definition and discussion of constraints and relations that are used in the syntactic analyses presented in the remainder of the book.

1. ATTRIBUTES AND VALUES

1.1. Functional Uncertainty

Recall that the TOPIC of a sentence in Russian appears in the specifier position of IP and also bears a grammatical function inside the same sentence (King 1995, page 206):

(1) *'Evgenija Onegina' napisal Puškin.*
Eugene Onegin wrote Pushkin
'Pushkin wrote 'Eugene Onegin'.'



Here the TOPIC also bears the OBJ function; in other examples, the TOPIC might be the SUBJ or an oblique function. This *functional uncertainty* about the grammatical function of the TOPIC can be represented by defining a special abbreviatory symbol GF representing a disjunction of all grammatical functions, as discussed in Chapter 5, Section 4.2:

(2) $\textsc{gf} \equiv \{\textsc{subj}\,|\,\textsc{obj}\,|\,\textsc{obj}_\theta\,|\,\textsc{comp}\,|\,\textsc{xcomp}\,|\,\textsc{obl}\,|\,\textsc{adj}\,|\,\textsc{xadj}\}$

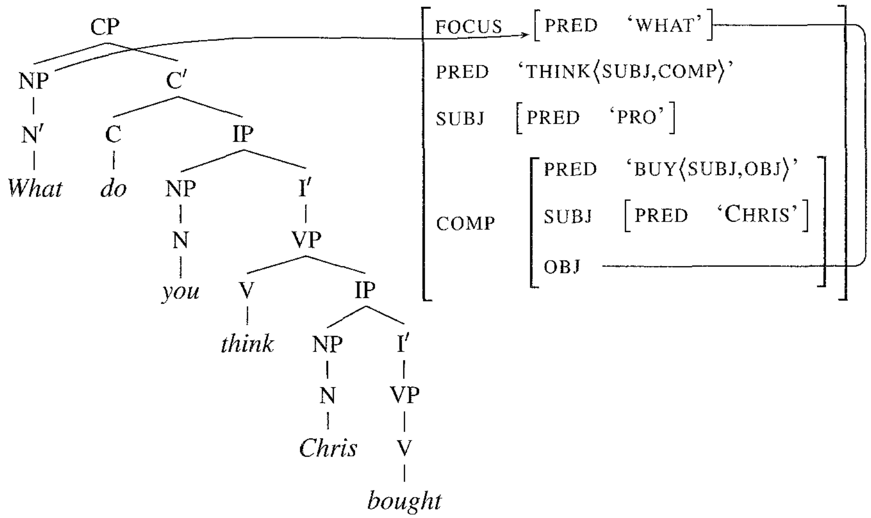
This symbol appears in the phrase structure rule for IP (King 1995, page 204):

(3) $$\text{IP} \longrightarrow \begin{pmatrix} \text{XP} \\ \downarrow \in (\uparrow \textsc{topic}) \\ (\uparrow \textsc{gf}) = \downarrow \end{pmatrix} \begin{pmatrix} \text{I}' \\ \uparrow = \downarrow \end{pmatrix}$$

An equation such as $(\uparrow \textsc{gf}) = \downarrow$ is satisfied if there is some value of GF for which the equation is true. Here, the equation is true if the value of GF is OBJ, since the TOPIC is also the OBJ of the sentence.

In this instance, the uncertainty about the grammatical function of the TOPIC was limited: one member of a disjunction of grammatical functions was chosen. In other cases, there might be more uncertainty; the TOPIC phrase might bear a grammatical function more deeply embedded inside the sentence. This is true for wh-questions in English. Example (4) shows that the question phrase *what* can also fill the role of OBJ in the complement clause COMP, appearing as the value of the path COMP OBJ in the f-structure. In example (7) (page 142) it is the value of the path COMP COMP OBJ.

(4) *What do you think Chris bought?*



A simplified version of the annotated phrase structure rule for English wh-questions is given in (5):

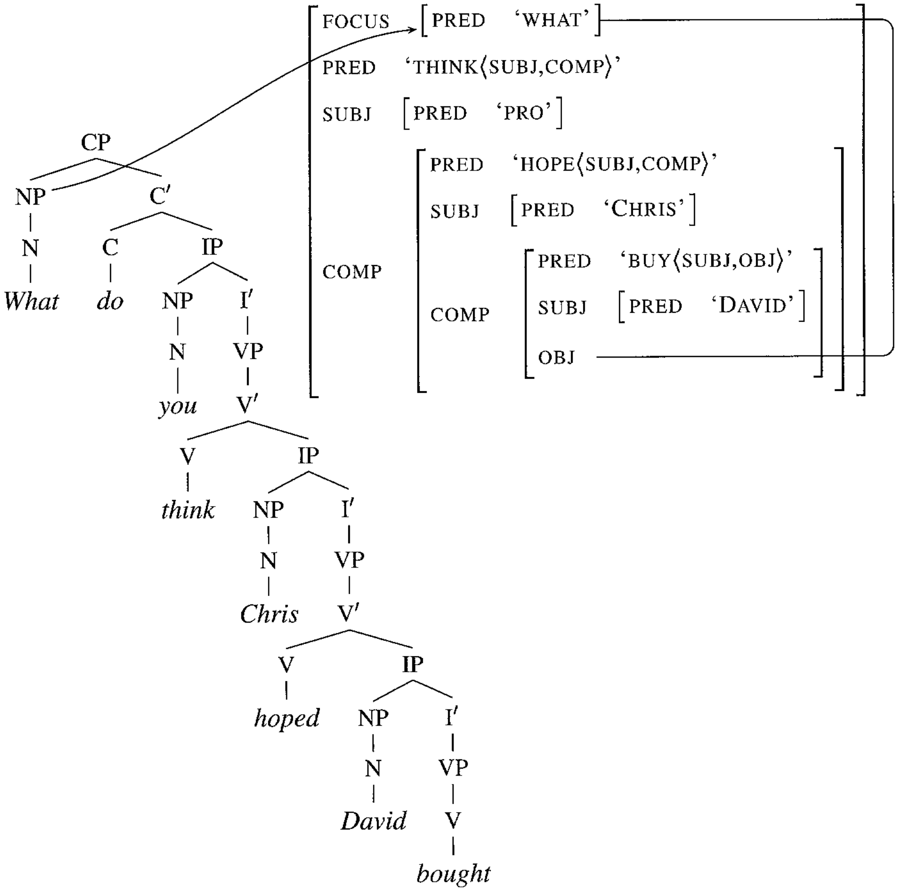
$$(5)\quad \text{CP} \longrightarrow \left(\begin{array}{c}\text{XP}\\ (\uparrow \textsc{focus}) = \downarrow\\ (\uparrow \textsc{focus}) = (\uparrow \textsc{comp}^{*}\ \textsc{gf})\end{array}\right) \left(\begin{array}{c}\text{C}'\\ \uparrow = \downarrow\end{array}\right)$$

The annotation (↑ FOCUS) = (↑ COMP* GF) on the XP daughter of CP contains a new sort of expression. As discussed in Chapter 5, Section 1.1, the *Kleene star operator* * indicates than an expression may be repeated zero or more times. In particular, COMP* represents paths containing any number of COMPs: the empty path, COMP, COMP COMP, and so on. Thus, the equation (↑ FOCUS) = (↑ COMP* GF) indicates that the FOCUS f-structure also fills some grammatical function GF within the f-structure ↑ which lies at the end of some path in the set of paths COMP*GF: that is, some GF that may be embedded inside any number of COMPs. The constraint holds if there is some path in the set of paths COMP* GF for which the equation is true. In example (4), the path is COMP OBJ. In example (7), the path is COMP COMP OBJ. In some other example, a different path might be chosen.

More complicated paths can also be characterized. A slightly more complete version of the rule for question formation in English is:

$$(6)\quad \text{CP} \longrightarrow \left(\begin{array}{c}\text{XP}\\ (\uparrow \textsc{focus}) = \downarrow\\ (\uparrow \textsc{focus}) = (\uparrow \{\textsc{xcomp}|\textsc{comp}\}^{*}\ \textsc{gf})\end{array}\right) \left(\begin{array}{c}\text{C}'\\ \uparrow = \downarrow\end{array}\right)$$

(7) *What do you think Chris hoped David bought?*



The regular expression {XCOMP|COMP}* denotes paths containing any number of XCOMPS or COMPS in any order: COMP XCOMP, COMP COMP, XCOMP COMP XCOMP, for example.[1]

Equations of this sort, involving abbreviatory symbols over grammatical functions or more complex regular expressions denoting paths through an f-structure, exemplify *functional uncertainty*. Functional uncertainty was first introduced by Kaplan et al. (1987) and Kaplan and Zaenen (1989) in the treatment of *long-distance dependencies* such as topicalization, question formation, and relative

[1] Note that this expression is not the same as the regular expression {XCOMP* | COMP*} which denotes paths containing *either* any number of XCOMPS *or* any number of COMPS: XCOMP XCOMP XCOMP or COMP COMP, but not XCOMP COMP.

clause formation in English. The expression in (7) more adequately captures constraints on question formation in English than the one in (6), but still does not completely characterize the possible grammatical functions of the sentence-initial FOCUS constituent in English questions; a detailed discussion of the syntax and semantics of long-distance dependencies can be found in Chapter 14.

Kaplan and Zaenen (1989) provide the following interpretation for constraints involving regular expressions over paths:[2]

(8) Functional uncertainty:

$(f\ \alpha) = v$ holds if and only if f is an f-structure, α is a set of strings, and for some s in the set of strings α, $(f\ s) = v$.

Definition (35) in Chapter 5, repeated in (9), tells us how to interpret instances where the string s is of length greater than one:

(9) $(f\ as) \equiv ((f\ a)\ s)$ for a symbol a and a (possibly empty) string of symbols s.

$(f\ \epsilon) \equiv f$, where ϵ is the empty string.

Much work has been done on the formal properties of systems using functional uncertainty. For an overview discussion, see Dalrymple et al. (1995d). The issue of decidability and functional uncertainty is treated in detail by Baader et al. (1991), Bakhofen (1993), and Keller (1993).

1.2. Inside-Out Functional Uncertainty

By using functional uncertainty, we can specify an f-structure embedded at an arbitrary depth inside another f-structure. We can also talk about f-structures that *enclose* an f-structure at an arbitrary level of distance. This is referred to as *inside-out functional uncertainty*, first introduced by Kaplan (1988). The two types of functional uncertainty are closely related, but they are used in different contexts: "regular" or outside-in functional uncertainty is used to define constraints on more deeply embedded structures, while inside-out functional uncertainty is used to define constraints on enclosing structures.

Inside-out functional uncertainty is used by Nordlinger (1998) in her analysis of *stacked casemarking* and *constructive case*. Consider the following Warlpiri example (Nordlinger 1998, page 136, attributed to Simpson 1991):

[2] The definition in (8) is stated in terms of the set of strings described by a regular expression. For example, XCOMP XCOMP is a member of the set of strings characterized by the regular expression XCOMP*. See Chapter 14, Section 1.1.4 for more discussion of functional uncertainty and an alternative formal definition.

(10) *Japanangka-rlu luwa-rnu marlu pirli-ngka-rlu*
Japanangka-ERG shoot-PAST kangaroo rock-LOC-ERG
'Japanangka shot the kangaroo on the rock.'

The noun *pirli-ngka-rlu* 'rock-LOC-ERG' contains two casemarkers, a LOCative marker *-ngka-* and an ERGative marker *-rlu*; the use of multiple casemarkers in this way is called stacked casemarking. Stacked casemarking specifies the syntactic environment within which the phrase must appear. Here, *pirli-* 'rock' is specified as being marked with LOCative case by the first casemarker *-ngka-*. The second casemarker, the ERGative marker *-rlu*, specifies that *pirli-ngka-* 'rock-LOC' must appear as a modifier of a phrase with ERGative case. In Warlpiri, an ERGative phrase is always a SUBJ, so specification of the case of the modified phrase also fixes its grammatical function. Thus, the use of multiple casemarkers specifies not only the features of the word itself, but also features of the syntactic environment in which the phrase must appear.

The f-structure corresponding to the noun dominating *pirli-ngka-rlu* is:

(11)
$$g\left[\begin{array}{ll}\text{PRED} & \text{`ROCK'}\\ \text{CASE} & \text{LOC}\end{array}\right]$$

The PRED value contributed by *pirli-ngka-rlu* is 'ROCK', and its CASE value is LOC. According to Nordlinger's analysis, the rules for stacked casemarking in Warlpiri require this f-structure to appear in the following f-structure environment:

(12)
$$\left[\begin{array}{ll}\text{SUBJ} & \left[\begin{array}{ll}\text{CASE} & \text{ERG}\\ \text{OBL}_{\text{LOC}} & g\left[\begin{array}{ll}\text{PRED} & \text{`ROCK'}\\ \text{CASE} & \text{LOC}\end{array}\right]\end{array}\right]\end{array}\right]$$

In (12), the f-structure for *pirli-ngka-rlu* is the one labeled *g*. It is required to appear in the f-structural environment shown: it must bear the OBL_{LOC} relation within the ERG-marked SUBJ.

Nordlinger enforces these requirements by means of the following lexical entry for *pirli-ngka-rlu*:

(13) *pirli-ngka-rlu*
$$\begin{array}{l}(\uparrow \text{PRED}) = \text{`ROCK'}\\ (\uparrow \text{CASE}) = \text{LOC}\\ ((\text{OBL}_{\text{LOC}} \uparrow)\ \text{CASE}) = \text{ERG}\\ (\text{SUBJ OBL}_{\text{LOC}} \uparrow)\end{array}$$

The first two equations state that the f-structure for *pirli-ngka-rlu* must have a PRED with value 'ROCK', and the value LOC for its CASE feature, as shown in (11).

The expression $(\text{OBL}_{\text{LOC}} \uparrow)$ in the third line of the lexical entry in (13) uses inside-out functional uncertainty to refer to an f-structure whose value for the attribute OBL$_{\text{LOC}}$ is the f-structure $\uparrow$. If we assume that $\uparrow$ is instantiated to the f-structure named g in the diagram in (14), then (OBL$_{\text{LOC}}$ $\uparrow$) is labeled f:

(14) $f = (\text{OBL}_{\text{LOC}}\ g)$

$$\left[\begin{array}{ll}\text{SUBJ} & f\left[\begin{array}{ll}\text{CASE} & \text{ERG}\\ \text{OBL}_{\text{LOC}} & g\begin{bmatrix}\text{PRED} & \text{`ROCK'}\\ \text{CASE} & \text{LOC}\end{bmatrix}\end{array}\right]\end{array}\right]$$

The expression ((OBL$_{\text{LOC}}$ $\uparrow$) CASE) = ERG requires (OBL$_{\text{LOC}}$ $\uparrow$), labeled f in (14), to contain the attribute CASE with value ERG.

Similarly, the expression (SUBJ OBL$_{\text{LOC}}$ $\uparrow$) in the fourth line of the lexical entry in (13) refers to an f-structure from which the path SUBJ OBL$_{\text{LOC}}$ leads to the f-structure $\uparrow$. Here, the expression represents an *existential constraint* (see Chapter 5, Section 2.6) requiring such an f-structure to exist.

Formally, $(a\ f)$ is the f-structure whose value for the attribute a is f:

(15) Inside-out expression:

$(af) = g$ holds if and only if g is an f-structure, a is a symbol, and the pair $\langle a, f\rangle \in g$.

Longer paths in an inside-out equation are interpreted incrementally, as with outside-in expressions (see Chapter 5, Section 2.1).

(16) $(\epsilon\ f) \equiv f$, where ϵ is the empty string.

$(saf) \equiv (s\ (a\ f))$ for a symbol a and a (possibly empty) string of symbols s.

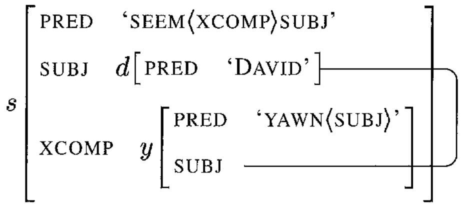
As with outside-in functional uncertainty, it is possible to use a regular expression to characterize a set of paths through the f-structure. This will be useful in our analysis of anaphora, to be presented in Chapter 11. The use of regular expressions in inside-out functional uncertainty is similar to its use in outside-in functional uncertainty: the expression is true if there is some string in the set picked out by the regular expression for which the expression holds.

(17) Inside-out functional uncertainty:

$(\alpha\ f) \equiv g$ if and only if g is an f-structure, α is a set of strings, and for some s in the set of strings α, $(s\ f) \equiv g$.

Notice that even when the inside-out path is fixed and the expression containing it appears to be determinate, it may denote one of several f-structures. Consider a structure with a verb like *seem*, whose subject is shared with the subject of its infinitival complement:

(18) *David seemed to yawn.*



The f-structure for *David*, labeled *d*, is the SUBJ of two f-structures: the f-structure for *seem*, labeled *s*, and the f-structure for *yawn*, labeled *y*. In this case, (SUBJ *d*), the f-structure of which *d* is the SUBJ, is either *s* or *y*.

1.3. Local Names for F-Structures

In expressing constraints on f-structures, a *local name* can be used in a lexical entry or annotated phrase structure rule to refer to an f-structure (Kaplan and Maxwell 1996). The reference of a particular local name is restricted to the lexical item or rule element within which it occurs; that is, a local name cannot be used in more than one daughter in a rule or more than one lexical item to refer to the same f-structure. A local name begins with the percent sign %.

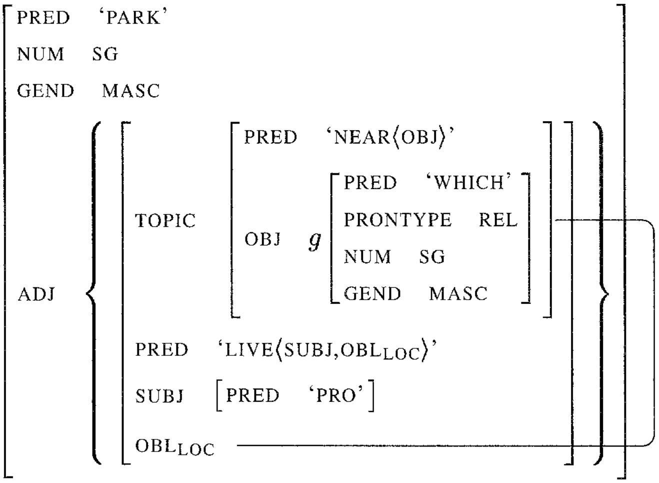
A local name is particularly useful in expressions involving functional uncertainty: it makes it possible to name a particular f-structure that participates in the uncertainty and to place constraints on it. For example, the relative pronoun in Russian agrees in number and gender with the head noun of the noun phrase. Lipson (1981) discusses the following Russian example, in which the masculine singular relative pronoun *kotorogo* must be used with a masculine singular noun like *park*:

(19) *park, okolo kotorogo ja živu*
park.MASC.SG near which.MASC.SG I live
'the park near which I live'

As example (19) shows, the relative pronoun can appear as a subconstituent of a displaced phrase such as *okolo kotorogo* 'near which'. The f-structure for this example is:[3]

[3] In Chapter 14, we augment the f-structure for relative clauses with an attributeRELPRO, whose value is the f-structure of the relative pronoun within the fronted phrase. Here we omit this attribute in order to provide a clear illustration of how a local name is used.

(20) *park, okolo kotorogo ja živu*
park.MASC.SG near which.MASC.SG I live



In the analysis of example (19), we would like to impose an agreement requirement that allows us to refer to an arbitrarily deeply embedded constituent, the relative pronoun, and to constrain its NUM and GEND features. The following phrase structure rule accomplishes this:

(21)
$$\text{NP} \longrightarrow \begin{pmatrix} \text{N} \\ \uparrow = \downarrow \end{pmatrix} \begin{pmatrix} \text{CP} \\ \downarrow \in (\uparrow \text{ ADJ}) \\ (\downarrow \text{ TOPIC GF}^*) = \%\text{RELPRON} \\ (\%\text{RELPRON PRONTYPE}) =_c \text{REL} \\ (\%\text{RELPRON NUM}) = (\uparrow \text{ NUM}) \\ (\%\text{RELPRON GEND}) = (\uparrow \text{ GEND}) \end{pmatrix}$$

This simplified rule states that a Russian noun phrase consists of a head noun and a CP relative clause, and that the CP's f-structure is a member of the set of modifiers of the phrase:

(22) $\downarrow \in (\uparrow \text{ ADJ})$

The relative clause CP contains a relative phrase, the phrase *okolo kotorogo* in example (19). The CP rule (not displayed here) ensures that this relative phrase bears the TOPIC function within the relative clause. According to the rule in (21), this TOPIC f-structure must contain a relative pronoun at some level of embedding GF* inside the TOPIC. This f-structure is referred to by a name local to this rule as %RELPRON:

(23) $(\downarrow \text{ TOPIC GF}^*) = \%\text{RELPRON}$

An f-structure name such as %RELPRON may be used either in a lexical item or in annotations on a category on the right-hand side of a phrase structure rule. The final three annotations in (21) place further constraints on the f-structure %RELPRON: it must have a PRONTYPE of REL, and its NUM and GEND must match the NUM and GEND of the mother NP:

(24) (%RELPRON PRONTYPE) $=_c$ REL
(%RELPRON NUM) = (↑ NUM)
(%RELPRON GEND) = (↑ GEND)

These constraints are satisfied if %RELPRON names the f-structure labeled g in example (20). Using a local name like %RELPRON is essential in this instance: the use of a local name ensures that all of the constraints in (24) refer to the *same* f-structure. In particular, a set of expressions like the following are *not* equivalent to those in (24):

(25) (↓ TOPIC GF* PRONTYPE) $=_c$ REL
(↓ TOPIC GF* NUM) = (↑ NUM)
(↓ TOPIC GF* GEND) = (↑ GEND)

The equations in (25) require some f-structure inside the topic to have a PRONTYPE of REL, some f-structure to have the same NUM as the full noun phrase, and some f-structure to have the same GEND as the full noun phrase; crucially, these constraints impose no requirement for the same f-structure to satisfy all of these constraints. It is the use of a local name that enforces the proper requirement.

1.4. Off-Path Constraints

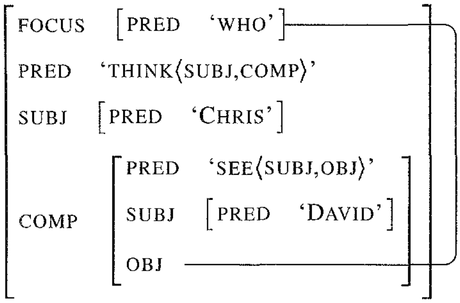
There are cases in which a long-distance dependency is constrained not in terms of the grammatical functions that appear on the path, but in terms of other properties of the f-structures on the path. For instance, some English verbs allow extraction from their sentential complements, while others do not:

(26) *Who did Chris think/*whisper that David saw?*

Verbs allowing extraction are often called *bridge verbs*, while those disallowing extraction are called *nonbridge verbs*.

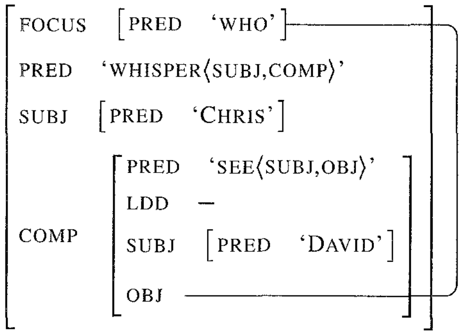
There is no reason to assume that the grammatical function of the sentential complements of these two verbs differs; other than this difference between them, they behave the same syntactically, and both bear the grammatical function COMP. A sentence with a bridge verb allowing extraction has an f-structure like the following:

(27) *Who did Chris think that David saw?*



We propose that the sentential complement COMP of a nonbridge verb bears a feature that bridge verbs lack, which we will call LDD, with value −. The path in a long-distance dependency may not pass through an f-structure with this feature:

(28) * *Who did Chris whisper that David saw?*



In example (28), the FOCUS constituent is related to its within-clause function OBJ by means of an equation such as the following on the phrase structure rule dominating the focused phrase:

(29) $(\uparrow \text{FOCUS}) = (\uparrow \text{COMP OBJ})$

The attributes COMP and OBJ do not reflect the prohibition against extraction. Instead, this requirement must be stated "off the path" characterizing the dependency, as an additional condition on the f-structures along the path. In this case, we would like to express the following constraint:

(30) A COMP in the extraction path must not contain the pair ⟨LDD, −⟩.

We can use *off-path constraints* to express this requirement in the following way:

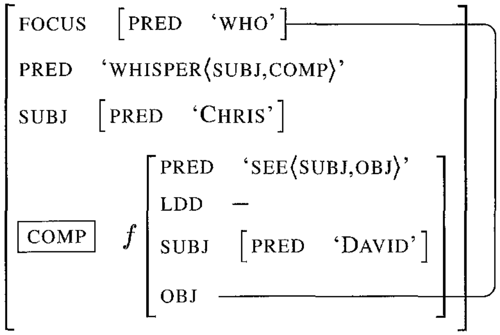
(31) $(\uparrow \text{FOCUS}) = (\uparrow \underset{(\rightarrow \text{LDD}) \neq -}{\text{COMP}} \quad \text{OBJ})$

In the expression in (32), the right arrow → stands for the value of the attribute COMP:

(32) $\underset{(\rightarrow \text{LDD}) \neq -}{\text{COMP}}$

In (33), the COMP attribute is boxed, and the f-structure denoted by → is labeled *f*:

(33) * *Who did Chris whisper that David saw?*



The f-structure *f* contains the attribute LDD with value −. This is forbidden by the negative constraint (→ LDD) ≠ −, accounting for the ungrammaticality of example (33).

Slightly more generally, we can use an expression like the following to constrain long distance dependencies in English:[4]

(34) $(\uparrow \text{FOCUS}) = (\uparrow \underset{(\rightarrow \text{LDD}) \neq -}{\text{COMP}^*} \quad \text{GF})$

This expression indicates that any number of occurrences of the annotated COMP attribute $\underset{(\rightarrow \text{LDD}) \neq -}{\text{COMP}}$ displayed in (32) can occur in the long-distance path; in other words, the FOCUS value of the f-structure ↑ also bears some grammatical function GF embedded inside any number of COMPs, as long as none of the COMP f-structures contain the pair ⟨LDD, −⟩.

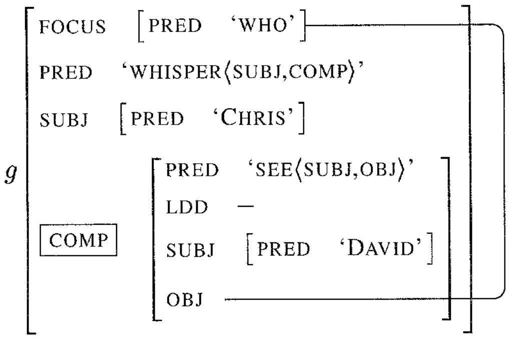
We can also use the left arrow ← in off-path constraints to denote the f-structure which contains an attribute. The following equation imposes a different requirement, not the one we want for English bridge verbs:

(35) $(\uparrow \text{FOCUS}) = (\uparrow \underset{(\leftarrow \text{LDD}) \neq -}{\text{COMP}} \quad \text{OBJ})$

[4] As in Section 1.1 of this chapter, this provisional characterization of constraints on question formation in English is incomplete; we provide a more complete treatment of the syntax and semantics of questions in Chapter 14.

This requires the f-structure of which the COMP is an attribute not to contain the feature LDD with value −. The left arrow ← in this equation refers to the f-structure labeled g in (36), since COMP is an attribute of g:

(36)



This is not the requirement that we want in this case; this constraint prevents the outermost f-structure g from having the attribute LDD with value −. However, it is not the outermost f-structure that must be constrained, but the value of its COMP.

Formally, we define the expressions ← and → as they are used in off-path constraints in the following way:

(37) Off-path constraints:

In an expression like $\underset{(\leftarrow s)}{a}$, ← refers to the f-structure of which a is an attribute.

In an expression like $\underset{(\rightarrow s)}{a}$, → refers to the value of the attribute a.

Using the f-structure variables ← and →, any kind of constraint can be written as an off-path constraint; defining equations, constraining equations, existential constraints, and other kinds of f-descriptions may be specified. We return to a discussion of off-path constraints and long-distance dependencies in Chapter 14.

1.5. The PCASE Attribute

The particular grammatical function of an oblique argument is determined in English by the preposition that is used. For example, the goal phrase *to Chris* in a sentence like *David gave the book to Chris* bears the grammatical function OBL_{GOAL}. Kaplan and Bresnan (1982) propose that the constraint specifying the grammatical function of an oblique phrase is given by the preposition; in this case, the information that *to Chris* is an OBL_{GOAL} is specified as the value of the PCASE attribute in the lexical entry of the preposition *to*:[5]

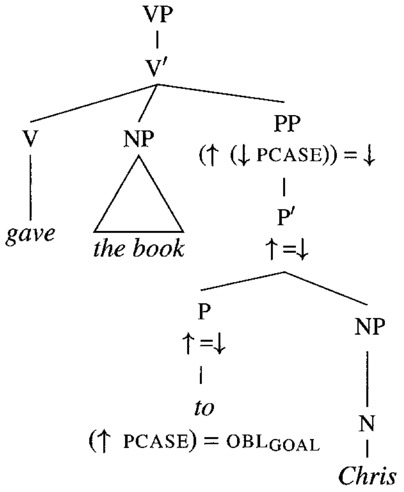
(38) *to* $(\uparrow \text{PCASE}) = \text{OBL}_{\text{GOAL}}$

Kaplan and Bresnan further propose that the value of the PCASE attribute is also the attribute whose value is the oblique phrase, so that OBL_{GOAL} is an attribute name as well as a value. The following annotation on the PP phrase structure node accomplishes this:

(39) PP
$(\uparrow (\downarrow \text{PCASE})) = \downarrow$

This annotation appears as a part of the rule expanding VP. The full expansion of the VP node is as follows:

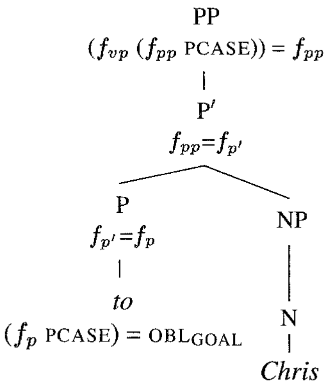
(40) *gave the book to Chris*



Using mnemonic names for f-structures such as f_{pp} for the f-structure corresponding to the PP node in the c-structure tree, the equations in (40) give rise to the instantiated equations given in (41) for the PP and the nodes it dominates:

[5] Kaplan and Bresnan's analysis prefigures the theory of *Constructive Case*, developed by Nordlinger (1998) and discussed briefly in Section 1.2 of this chapter, according to which the grammatical function of an argument is specified by the case morpheme with which it appears.

(41)


PP
$(f_{vp}\ (f_{pp}\ \text{PCASE})) = f_{pp}$
P′
$f_{pp}{=}f_{p'}$
P
$f_{p'}{=}f_p$
to
$(f_p\ \text{PCASE}) = \text{OBL}_{\text{GOAL}}$
NP
N
Chris

The relevant equations are the following:

(42)
$$\begin{array}{l}(f_{vp}\ (f_{pp}\ \text{PCASE})) = f_{pp}\\ f_{pp}{=}f_{p'}\\ f_{p'}{=}f_p\\ (f_p\ \text{PCASE}) = \text{OBL}_{\text{GOAL}}\end{array}$$

These equations tell us that the f-structure f_{pp} corresponding to the PP node is the same as the f-structures $f_{p'}$ and f_p corresponding to the P′ and P nodes, and that f_p's PCASE is OBL_{GOAL}. Thus, we have the following equivalences:

(43) $(f_{pp}\ \text{PCASE}) = (f_{p'}\ \text{PCASE}) = (f_p\ \text{PCASE}) = \text{OBL}_{\text{GOAL}}$

Substituting OBL_{GOAL} for $(f_{pp}\ \text{PCASE})$ in the first equation in (42), we have:

(44) $(f_{vp}\ \text{OBL}_{\text{GOAL}}) = f_{pp}$

The equality induced by the constraint in (44) is explicitly indicated in the f-structure in (45):

(45)
$$f_{vp}\left[\begin{array}{ll}\text{PRED} & \text{`GIVE}\langle\text{SUBJ,OBJ,OBL}_{\text{GOAL}}\rangle\text{'}\\ \text{OBL}_{\text{GOAL}} & f_{pp,p',p}\left[\begin{array}{ll}\text{PRED} & \text{`CHRIS'}\\ \text{PCASE} & \text{OBL}_{\text{GOAL}}\end{array}\right]\end{array}\right]$$

2. TALKING ABOUT SETS

Sets are used to represent several different types of objects in LFG. In general, sets are used where an unbounded number of elements is allowed: for coordinate structures, for example, where there is no fixed limit to the number of conjuncts; or for the modifiers of a phrase, where any number of modifiers may appear. More recently, Dalrymple and Kaplan (2000) have proposed the use of sets of atomic values as values of features like CASE, PERS, and GEND to account for feature indeterminacy and feature resolution. Here we discuss ways of describing sets and constraining their members.

2.1. Open Set Descriptions

An *open set description* is given by separately specifying the individual elements of a set; the constraints specifying the elements may be given in different parts of the grammar, by different phrase structure rules or lexical items. For example, consider the following simplified rule for the English verb and its complements:

(46)
$$\text{V}' \longrightarrow \left(\begin{array}{c}\text{V}\\ \uparrow = \downarrow\end{array}\right) \left(\begin{array}{c}\text{NP}\\ (\uparrow \textsc{obj}) = \downarrow\end{array}\right) \begin{array}{c}\text{PP*}\\ \downarrow\in (\uparrow \textsc{adj})\end{array}$$

The expression PP* represents a sequence of zero or more PPs. What about the annotation $\downarrow\in (\uparrow \textsc{adj})$? This annotation means that the f-structure of each PP that appears is a member ($\in$) of the ADJ set of the mother's f-structure $\uparrow$. That is, there may be zero or more occurrences of the following annotated node:

(47)
$$\begin{array}{c}\text{PP}\\ \downarrow\in (\uparrow \textsc{adj})\end{array}$$

The expression in (48) represents an alternative way of specifying set membership:

(48) $(\uparrow \textsc{adj} \in) = \downarrow$

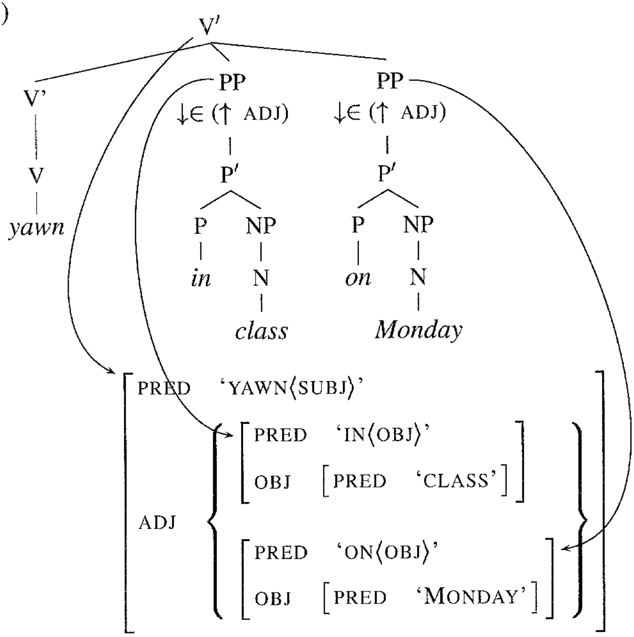
This expression uses the set membership symbol $\in$ as an *attribute* and states that $\downarrow$ is a member of the set $(\uparrow \textsc{adj})$. Expressions such as these are sometimes useful in writing constraints on set members, particularly in expressions involving *inside-out functional uncertainty*, discussed in Section 1.2 of this chapter. The two expressions in (49) are equivalent; each states that the f-structure $\downarrow$ is a member of the set of f-structures $(\uparrow \textsc{adj})$:

(49) $\downarrow\in (\uparrow \textsc{adj})$

$(\uparrow \textsc{adj} \in) = \downarrow$

The c-structure and f-structure for a V′ like *yawn in class on Monday* are given in example (50):

(50)



As the annotations on the rule require, the f-structure of each modifying adjunct PP is a member of the ADJ set of the f-structure for the mother V′ node.

Formally, an expression involving set membership is defined as we would expect:

(51) Open set description:

$g \in f$ holds if and only if f is a set and g is a member of f.

It is also possible to write a constraining expression for set membership:

(52) Constraining statement of set membership:

$g \in_c f$ holds if and only if f is a set and g is a member of f in the minimal solution for the defining equations in the f-description of f.

Rounds (1988) provides more discussion of the description and representation of sets in LFG.

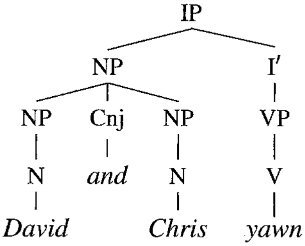
2.2. Distributive and Nondistributive Features

Sets are also used in the representation of coordinate structures, but in that case there is a difference: following a proposal by John Maxwell (p.c.), Dalrymple and Kaplan (2000) treat coordinate structures as *hybrid objects*, sets with both elements and properties.[6] This captures the fact that a coordinate structure such as *David and Chris* in an example like (53) has properties that the individual conjuncts do not have:

(53) *David and Chris yawn/*yawns.*

Although both *David* and *Chris* are singular phrases, the coordinate structure as a whole is plural. The c-structure and f-structure for such an example are:

(54) *David and Chris yawn.*



$$\left[\begin{array}{ll} \text{PRED} & \text{‘YAWN}\langle\text{SUBJ}\rangle\text{’} \\ \text{SUBJ} & \left[\begin{array}{l} \text{NUM} \quad \text{PL} \\ \left\{ \begin{array}{l} \left[\begin{array}{ll} \text{PRED} & \text{‘DAVID’} \\ \text{PERS} & 3 \\ \text{NUM} & \text{SG} \end{array}\right] \\ \\ \left[\begin{array}{ll} \text{PRED} & \text{‘CHRIS’} \\ \text{PERS} & 3 \\ \text{NUM} & \text{SG} \end{array}\right] \end{array}\right\} \end{array}\right] \end{array}\right]$$

We present here a simplified, preliminary phrase structure rule for NP coordination; a more detailed discussion of coordination can be found in Chapter 13:

(55) $$\begin{array}{lllll} \text{NP} & \longrightarrow & \text{NP} & \text{Cnj} & \text{NP} \\ & & \downarrow \in \uparrow & (\uparrow \text{NUM}) = \text{PL} & \downarrow \in \uparrow \end{array}$$

The annotations on the NP daughters require the f-structure for each conjunct NP to be a member of the f-structure for the coordinate NP. That is, the f-structure for the NP as a whole is a set, with the NP conjuncts as its members.

The annotation on the Cnj daughter requires the coordinate structure to have a NUM feature whose value is PL: the coordinate structure is a plural phrase. In other words, the set representing the coordinate structure is given the attribute NUM with value PL. What does it mean to specify a property of a set in this way?

2.2.1. Nondistributive Features

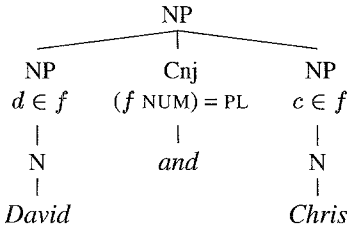
In specifying a property of a set, the property may or may not *distribute* to the members of the set, depending on whether the feature involved is a *distributive* or a *nondistributive* feature. For the present, we assume that the following features are nondistributive:

[6] This proposal is foreshadowed in work on coordination in LFG by Peterson (1982) and Andrews (1983a).

(56) Nondistributive features:
PERS, NUM, GEND

If a feature is nondistributive, it and its value become a property of the set as a whole. Thus, the NUM feature and its value specified in the rule in (55) are a property of the coordinate structure as a whole, not the individual conjuncts:

(57) *David and Chris*

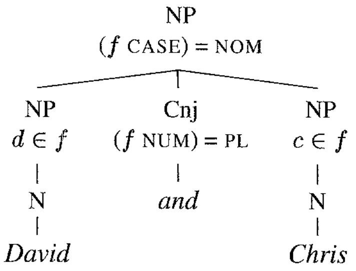


$$f\left[\begin{array}{ll} \text{NUM} & \text{PL} \\ \multicolumn{2}{l}{\left\{\begin{array}{l} d\left[\begin{array}{ll}\text{PRED} & \text{`DAVID'}\\ \text{PERS} & 3\\ \text{NUM} & \text{SG}\end{array}\right] \\ c\left[\begin{array}{ll}\text{PRED} & \text{`CHRIS'}\\ \text{PERS} & 3\\ \text{NUM} & \text{SG}\end{array}\right]\end{array}\right\}}\end{array}\right]$$

2.2.2. Distributive Features

In contrast, a distributive feature is an attribute of each member of the set, not of the set as a whole. Suppose, for example, that we want to specify the CASE of a coordinate phrase. Since CASE is a distributive feature, requiring the set f to have the attribute CASE with value NOM means that each member of the set f — here, d and c — must contain the pair ⟨CASE, NOM⟩:

(58) *David and Chris*



$$f\left[\begin{array}{ll} \text{NUM} & \text{PL} \\ \multicolumn{2}{l}{\left\{\begin{array}{l} d\left[\begin{array}{ll}\text{PRED} & \text{`DAVID'}\\ \text{PERS} & 3\\ \text{NUM} & \text{SG}\\ \text{CASE} & \text{NOM}\end{array}\right] \\ c\left[\begin{array}{ll}\text{PRED} & \text{`CHRIS'}\\ \text{PERS} & 3\\ \text{NUM} & \text{SG}\\ \text{CASE} & \text{NOM}\end{array}\right]\end{array}\right\}}\end{array}\right]$$

Formally, distributive and nondistributive features are treated in the following way (Dalrymple and Kaplan 2000):

(59) Distributive and nondistributive features:

If a is a *distributive* feature and s is a set of f-structures, then $(s\ a) = v$ holds if and only if $(f\ a) = v$ for all f-structures f that are members of the set s.

If a is a *nondistributive* feature, then $(f\ a) = v$ holds if and only if the pair $\langle a, v\rangle \in f$.

2.3. Closed Set Descriptions

A *closed set description* exhaustively enumerates the elements of the set instead of specifying the elements of the set by means of separate constraints mentioning each element. Dalrymple and Kaplan (2000) use closed set descriptions to represent the values of features like PERS:

(60) *We yawned.*

$$\begin{bmatrix} \text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & \begin{bmatrix} \text{PERS} & \{\text{S,H}\}\ (\,=\text{FIRST PERSON}) \\ \text{NUM} & \text{PL} \\ \text{PRED} & \text{`PRO'} \end{bmatrix} \end{bmatrix}$$

The first person feature is defined as the set {S,H}, mnemonic for speaker and Hearer, explained in more detail in Chapter 13. This representation enables a simple and intuitive treatment of feature resolution in coordination. The closed set description characterizing the value of the PERS feature for the pronoun *we* is:

(61) *we* $(\uparrow \text{PERS}) = \{\text{S,H}\}$

This description differs from the constraints in (62), which are consistent with the presence of other members of the set; the constraint in (61) is not:

(62) *we* $\text{S} \in (\uparrow \text{PERS})$
$\text{H} \in (\uparrow \text{PERS})$

For example, the additional constraint $\text{O} \in (\uparrow \text{PERS})$ is compatible with the constraints in (62) but not with the constraint in (61).

3. RELATIONS BETWEEN F-STRUCTURES

3.1. F-Command

F-command is a relation between f-structures analogous to the c-command relation defined on trees (Reinhart 1976). F-command was originally defined by Bresnan (1982a) in the following way:

(63) F-command:

f f-commands g if and only if f does not contain g, and all f-structures that contain f also contain g.

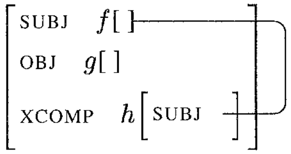
In examples (64a) and (64b), the f-structure labeled f f-commands the f-structure labeled g. In (64a), but not in (64b), g also f-commands f:

(64) f f-commands g:

a. $\begin{bmatrix} \text{SUBJ} & f[\,] \\ \text{OBJ} & g[\,] \end{bmatrix}$ b. $\begin{bmatrix} \text{SUBJ} & f[\,] \\ \text{COMP} & \begin{bmatrix} \text{SUBJ} & g[\,] \end{bmatrix} \end{bmatrix}$

The definition of f-command given in (63) is correct for cases like (64). However, as pointed out by Ron Kaplan (p.c.), this definition may not make the right predictions in cases in which two attributes share the same value. Consider the f-structure in (65), where the f-structure labeled f is the value of the SUBJ as well as the XCOMP SUBJ:

(65)



The f-structure labeled f in (65) does not f-command the f-structure labeled g, because there is an f-structure (namely h) that contains f but does not contain g. For the f-command relation to hold between f and g, we can formulate a new definition of f-command using inside-out functional uncertainty (Section 1.2 of this chapter):

(66) F-command, definition 2:

f f-commands g if and only if $\neg(f\ \text{GF}^*) = g$ (f does not contain g) and $((\text{GF}\ f)\ \text{GF}^+) = g$ (all f-structures whose value for some grammatical function GF is f also contain g).

The notion of f-command is important in the characterization of binding relations between pronouns and their antecedents: in many cases, the antecedent of a reflexive pronoun like *himself* must f-command the pronoun. The contrast in acceptability between examples (67a) and (67b) is due to the fact that in example (67a), the antecedent f of the reflexive pronoun *himself* f-commands the f-structure g of the pronoun, while the f-command relation does not hold in (67b):

(67) a. *David$_i$ saw himself$_i$.*

$$\begin{bmatrix} \text{PRED} & \text{`SEE}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & f\begin{bmatrix}\text{PRED} & \text{`DAVID'}\end{bmatrix} \\ \text{OBJ} & g\begin{bmatrix}\text{PRED} & \text{`PRO'} \\ \text{PRONTYPE} & \text{REFL}\end{bmatrix}\end{bmatrix}$$

b. **David$_i$'s mother saw himself$_i$.*

$$\begin{bmatrix} \text{PRED} & \text{`SEE}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`MOTHER'} \\ \text{SPEC} & f\begin{bmatrix}\text{PRED} & \text{`DAVID'}\end{bmatrix}\end{bmatrix} \\ \text{OBJ} & g\begin{bmatrix}\text{PRED} & \text{`PRO'} \\ \text{PRONTYPE} & \text{REFL}\end{bmatrix}\end{bmatrix}$$

Chapter 11 provides a fuller discussion of constraints on anaphoric binding; there, we will see that the f-command condition for antecedents of reflexive pronouns follows as a corollary from the binding requirements for reflexives, along the lines of the definition in (66).

3.2. Subsumption

Subsumption is a relation that holds between two f-structures f and g if g is compatible with but perhaps has more structure than f. In other words, f subsumes g if f and g are the same, or if g is the same as f except that it contains some additional structure that does not appear in f. For example, the f-structure labeled f in (68) subsumes the f-structure labeled g:

(68) f subsumes g:

$$f\begin{bmatrix} \text{PRED} & \text{`GO}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix}\text{NUM} & \text{SG}\end{bmatrix}\end{bmatrix} \qquad g\begin{bmatrix} \text{PRED} & \text{`GO}\langle\text{SUBJ}\rangle\text{'} \\ \text{TENSE} & \text{FUTURE} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`PRO'} \\ \text{CASE} & \text{NOM} \\ \text{NUM} & \text{SG}\end{bmatrix}\end{bmatrix}$$

Dalrymple and Kaplan (2000) use subsumption in their analysis of feature resolution in coordination (see Chapter 13). The subsumption relation can be formally defined as follows:

(69) Subsumption:

An f-structure f subsumes an f-structure g ($f \sqsubseteq g$) if and only if:

$f = g$; or

f and g are sets, and for each member f_1 of f there is a member g_1 of g such that $f_1 \sqsubseteq g_1$; or

f and g are f-structures, and for each attribute-value pair $\langle a, v_1 \rangle \in f$, there is a pair $\langle a, v_2 \rangle \in g$ such that $v_1 \sqsubseteq v_2$.

3.3. Generalization

Intuitively, the *generalization* of two f-structures is the structure that they have in common. For example, in (70) the f-structure labeled f is the generalization of the f-structures g and h:

(70) f is the generalization of g and h:

$$f\begin{bmatrix}\text{PRED} & \text{`GO}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix}\text{CASE} & \text{NOM}\end{bmatrix}\end{bmatrix}$$

$$g\begin{bmatrix}\text{PRED} & \text{`GO}\langle\text{SUBJ}\rangle\text{'} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`CHRIS'} \\ \text{CASE} & \text{NOM}\end{bmatrix}\end{bmatrix} \qquad h\begin{bmatrix}\text{PRED} & \text{`GO}\langle\text{SUBJ}\rangle\text{'} \\ \text{TENSE} & \text{FUTURE} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`DAVID'} \\ \text{CASE} & \text{NOM} \\ \text{NUM} & \text{SG}\end{bmatrix}\end{bmatrix}$$

Kaplan and Maxwell (1988) use generalization in their analysis of coordination, proposing that the value of an attribute of a set is the generalization of the values of the attributes of the elements of the set. We propose a different analysis of coordination in Chapter 13.

Formally, the generalization $f_1 \prod f_2$ of two f-structures f_1 and f_2 is defined recursively as follows (see also Kaplan and Maxwell 1988):

(71) Generalization:

An f-structure f is the generalization $f_1 \prod f_2$ of two f-structures f_1 and f_2 if and only if:

$f_1 = f_2 = f$; or

f_1 and f_2 are sets, and each member of f is the generalization of some member of f_1 and some member of f_2; or,

f_1 and f_2 are f-structures, and for each pair $\langle a, v_1 \rangle \in f_1$, if there is a pair $\langle a, v_2 \rangle \in f_2$, then $\langle a, v_1 \prod v_2 \rangle \in f$.

Unlike many previous definitions of generalization, (71) defines the generalization of two sets. This definition has an interesting consequence: the generalization of two sets may not be unique. For instance, consider the two sets given in (72a) and (72b):

(72) a. $\left\{ \begin{bmatrix} \text{F1} & \text{V1} \\ \text{F2} & \text{V2} \end{bmatrix} \begin{bmatrix} \text{F3} & \text{V3} \\ \text{F4} & \text{V4} \end{bmatrix} \right\}$

b. $\left\{ \begin{bmatrix} \text{F1} & \text{V1} \\ \text{F3} & \text{V3} \end{bmatrix} \begin{bmatrix} \text{F2} & \text{V2} \\ \text{F4} & \text{V4} \end{bmatrix} \right\}$

According to the definition in (71), both of the following two sets constitute a generalization of the sets in (72):

(73) a. $\left\{ \begin{bmatrix} \text{F1} & \text{V1} \end{bmatrix} \begin{bmatrix} \text{F4} & \text{V4} \end{bmatrix} \right\}$

b. $\left\{ \begin{bmatrix} \text{F2} & \text{V2} \end{bmatrix} \begin{bmatrix} \text{F3} & \text{V3} \end{bmatrix} \right\}$

3.4. Restriction

The *restriction* of an f-structure with respect to an attribute can be intuitively defined as the f-structure that results from removing the attribute and its value (Kaplan and Wedekind 1993). The f-structure labeled $g|_{\text{TENSE}}$ in (74) is the restriction with respect to TENSE of the f-structure labeled g:

(74) $g|_{\text{TENSE}}$ is the restriction of g with respect to TENSE:

$$g \begin{bmatrix} \text{PRED} & \text{`GO}\langle\text{SUBJ}\rangle\text{'} \\ \text{TENSE} & \text{PAST} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`CHRIS'} \end{bmatrix} \end{bmatrix}$$

$$g|_{\text{TENSE}} \begin{bmatrix} \text{PRED} & \text{`GO}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`CHRIS'} \end{bmatrix} \end{bmatrix}$$

More formally, Kaplan and Wedekind (1993) define the restriction of an f-structure f with respect to an attribute a as follows:

(75) Restriction:

$$f|_a \equiv \{\langle s, v \rangle \in f \mid s \neq a\}$$

If a is a set-valued attribute:

$$f|_{\langle av \rangle} \equiv \begin{cases} f|_a & \text{if } (f\ a) = \{v\} \\ f|_a \cup \{\langle a, (f\ a) - \{v\} \rangle\} & \text{otherwise} \end{cases}$$

Restriction is useful if an f-structure plays two roles, with some different syntactic feature associated with each role. For example, assume that some f-structure f is shared as the SUBJ value of two different f-structures g_1 and g_2, but that f must take on a different CASE value in each structure. The equation in (76a) requires all of the attribute-value pairs of f other than CASE to be same as the attribute-value pairs of g_1's SUBJ other than CASE, and the equation in (76b) imposes the same requirement for g_2:

(76) a. $f|_{\text{CASE}} = (g_1\ \text{SUBJ})|_{\text{CASE}}$

b. $f|_{\text{CASE}} = (g_2\ \text{SUBJ})|_{\text{CASE}}$

We can then specify different CASE values for the subjects of g_1 and g_2; the constraints in (77) are consistent with the requirements in (76):

(77) a. $(g_1\ \text{SUBJ CASE}) = \text{NOM}$

b. $(g_2\ \text{SUBJ CASE}) = \text{ACC}$

Kaplan and Wedekind (1993) also use restriction in their analysis of the semantics of modification. We will present a different analysis of modification in Chapter 10.

3.5. Priority Union

Kaplan (1987) first proposed the operation of *priority union*, defined in (78):

(78) Priority union, definition 1:

An f-structure f/g is the priority union of f with g (or "f given g"), if f/g is the set of pairs $\langle a, v \rangle$ such that v is equal to the value of the attribute a in f if it exists, otherwise the value of a in g.

Intuitively, the priority union of two f-structures contains all the structure that each f-structure has, with one of the f-structures "winning" if there is a conflict. For example, in (79) f/g is the priority union of f with g:

(79) f/g is the priority union of f with g:

$$f/g\begin{bmatrix}\text{SUBJ} & s\\ \text{OBJ} & o1\\ \text{COMP} & c\\ \text{OBJ}_\theta & t\end{bmatrix}$$

$$f\begin{bmatrix}\text{SUBJ} & s\\ \text{OBJ} & o1\\ \text{COMP} & c\end{bmatrix}\qquad g\begin{bmatrix}\text{SUBJ} & s\\ \text{OBJ} & o2\\ \text{OBJ}_\theta & t\end{bmatrix}$$

The priority union f/g has all the structure in f as well as all the structure in g that does not conflict with f.

Kaplan's original definition of priority union, given in (78), was intended as a proposal for the analysis of elliptical constructions. For example, we might assume the following incomplete f-structures for a coordinate sentence with gapping:

(80) *David saw Chris, and Matty Ken.*

$$\left\{ g\begin{bmatrix}\text{PRED} & \text{`SEE}\langle\text{SUBJ,OBJ}\rangle\text{'}\\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`DAVID'}\end{bmatrix}\\ \text{OBJ} & \begin{bmatrix}\text{PRED} & \text{`CHRIS'}\end{bmatrix}\end{bmatrix}\; f\begin{bmatrix}\text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`MATTY'}\end{bmatrix}\\ \text{OBJ} & \begin{bmatrix}\text{PRED} & \text{`KEN'}\end{bmatrix}\end{bmatrix}\right\}$$

An analysis of gapping that appeals to priority union might propose that the final f-structure f for the second conjunct *Matty Ken* is obtained by taking the priority union f/g: in effect, the f-structure for *Matty* would replace the f-structure for *David*, and similarly for the f-structures for *Chris* and *Ken*:

(81) f/g is the priority union of f with g:

$$f/g\begin{bmatrix}\text{PRED} & \text{`SEE}\langle\text{SUBJ,OBJ}\rangle\text{'}\\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`MATTY'}\end{bmatrix}\\ \text{OBJ} & \begin{bmatrix}\text{PRED} & \text{`KEN'}\end{bmatrix}\end{bmatrix}$$

$$f\begin{bmatrix}\text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`MATTY'}\end{bmatrix}\\ \text{OBJ} & \begin{bmatrix}\text{PRED} & \text{`KEN'}\end{bmatrix}\end{bmatrix}\qquad g\begin{bmatrix}\text{PRED} & \text{`SEE}\langle\text{SUBJ,OBJ}\rangle\text{'}\\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`DAVID'}\end{bmatrix}\\ \text{OBJ} & \begin{bmatrix}\text{PRED} & \text{`CHRIS'}\end{bmatrix}\end{bmatrix}$$

Priority union produces for the second conjunct an f-structure like the one that would be associated with a sentence like *Matty saw Ken.*

Kaplan (1987) purposely formulated the definition of priority union to refer only to the top-level attributes of f and g; the definition given in (79) is not recursive. However, later work (for instance, Brun 1996b) assumes a recursive definition for priority union like the following:

(82) Priority union, definition 2:

An f-structure f/g is the priority union of f with g (or "f given g") if and only if:

f is atomic, and $f = f/g$; or,

f and g are sets, and $f/g = f \cup g$; or,

f is an f-structure, and:

if $\langle a, v_1 \rangle \in f$ and $\langle a, v_2 \rangle \in g$, then $\langle a, v_1/v_2 \rangle \in f/g$.
if $\langle a, v_1 \rangle \in f$ and there is no pair $\langle a, v_2 \rangle \in g$, then $\langle a, v_1 \rangle \in f/g$.
if $\langle a, v_2 \rangle \in g$ and there is no pair $\langle a, v_1 \rangle \in f$, then $\langle a, v_2 \rangle \in f/g$.

Future work will show which of these definitions is the most useful one.

4. C-STRUCTURE/F-STRUCTURE CONSTRAINTS

4.1. Wellformedness Conditions on C-Structures

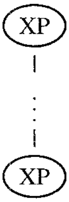
Constituent structure representations are governed by a constraint on valid derivations originally proposed by Kaplan and Bresnan (1982):

(83) Nonbranching Dominance Constraint, preliminary version:

A c-structure derivation is valid if and only if no category appears twice in a nonbranching dominance chain.

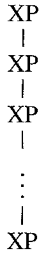
Intuitively, this requirement prevents a sentence from having an infinite number of c-structures by preventing a c-structure node from dominating another node with the same label in a nonbranching chain:

(84) Disallowed:



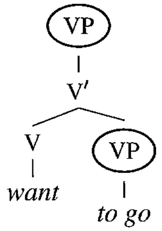
A tree like this one is not permitted, since an XP cannot dominate another XP without also dominating some other material as well. If this were permitted, there could be a chain of XPs of unbounded length dominating any XP, giving rise to an infinite number of possible constituent structure trees for that XP:

(85) Disallowed:



Of course, a phrase may dominate a phrase of the same type if other material is present as well:

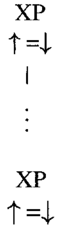
(86) Permitted:



This constraint is also discussed by Pereira and Warren (1983), who refer to the constraint as "off-line parsability"; this is because in their formulation it depends on the application of the nonbranching dominance constraint "off-line," after the parsing algorithm has applied to derive a set of trees for a string. Johnson (1988) also provides a definition of off-line parsability that is very similar to the definition in (83).

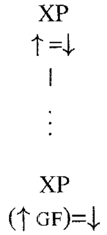
In more recent work (Kaplan and Maxwell 1996), the nonbranching dominance constraint has been revised to allow nonbranching dominance chains with nodes of the same category if the two nodes have different *functional annotations*. Under this view, the following dominance chain is ill-formed:

(87) Disallowed, revised nonbranching dominance constraint:



However, the following configuration is permitted:

(88) Permitted, revised nonbranching dominance constraint:



(89) Nonbranching Dominance Constraint, final version:

A c-structure derivation is valid if and only if there are no categories with the same functional annotation appearing twice in a nonbranching dominance chain.

Kaplan and Bresnan (1982) show that the nonbranching dominance constraint is important in proving that the membership problem for lexical functional grammars is *decidable* — that it is always possible to determine whether a string is acceptable according to a given grammar:

(90) Decidability Theorem [Kaplan and Bresnan 1982, (181)]:

For any lexical functional grammar G and for any string s, it is decidable whether s belongs to the language of G.

4.2. Category-Function Correlations

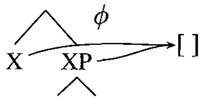
Certain lexical categories may be associated only with certain grammatical functions. For example, Bresnan and Moshi (1990) propose that in general, only verbs and prepositions can subcategorize for the OBJ function, though exceptions to this tendency have often been noted: for instance, Maling (1983) discusses transitive adjectives, and Iida (1987) discusses casemarking nominals.

There are also correlations between phrase structure positions and functional annotations associated with those positions. For example, we have seen that in English the specifier of IP is associated with the grammatical function SUBJ. English also allows sentential subjects, subjects with the phrase structure category CP; however, Bresnan (1994) presents evidence that the categories CP and PP cannot appear in the specifier of IP position, the canonical position for subjects (see also Bresnan 2001b). Of course, as Bresnan shows, this does not prevent phrases of those categories from bearing the SUBJ function: CP and PP can appear in a TOPIC or FOCUS position and may be associated with the SUBJ function by means of a functional uncertainty equation.

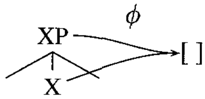
Zaenen and Kaplan (1995) propose to determine phrase structure category labels on the basis of the c-structure/f-structure relation. In particular, they propose that the constituent structure category of a phrase is determined on the basis of its relation to the lexical head of its f-structure:

(91) A maximal (nonlexical) category is of type XP if it corresponds to an f-structure that also corresponds to a lexical category of type X (Zaenen and Kaplan 1995).

a.



b.



This proposal is an interesting one, but does not allow for the existence of functional categories; as discussed in Chapter 4, Section 2.3.1, the same f-structure can be associated both with a lexical category and with a functional category, both with maximal (but different) phrasal projections. Since we assume the existence of functional categories, we do not adopt Zaenen and Kaplan's reinterpretation of constituent structure category determination.

4.3. Inverse Correspondences

In our discussion of subcategorization in Chapter 2, Section 2, we noted that LFG defines subcategorization requirements in functional terms: predicates subcategorize for a particular set of grammatical functions rather than phrasal categories or configurations. In many instances, as shown by Maling (1983), what appears to be evidence for selection for a particular phrase structure category is often better explained in semantic terms. In some cases, however, constraints on syntactic category do seem to be at issue.

Some predicates are exceptional in that they impose a categorial requirement on their arguments, restricting the constituent structure category of the argument to be only a subset of the categories that may be associated with a particular grammatical function. For instance, Pollard and Sag (1994, Chapter 3) claim that the verb *grow* cannot be used with a noun phrase complement, though complements of other phrase structure categories are acceptable:

(92) a. *Kim grew successful.*

b. *Kim grew to love Chris.*

c. **Kim grew a success.*

A very few English verbs subcategorize for phrases of one particular c-structure category; the verb *wax* is a marginal example of one such verb:

(93) a. *Kim waxed poetical.*

b. **Kim waxed a success.*

c. **Kim waxed sent more and more leaflets.*

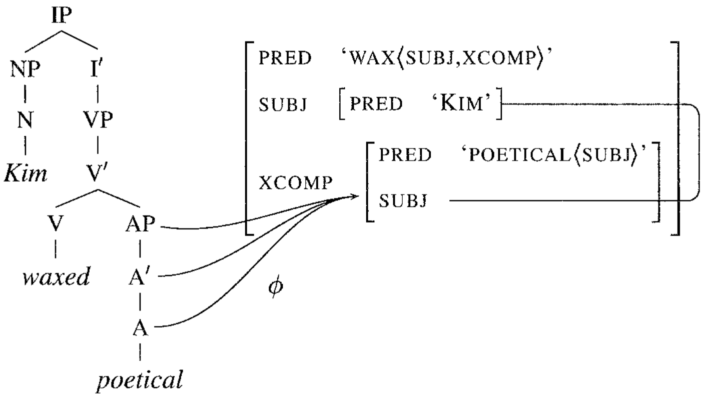
d. **Kim waxed doing all the work.*

e. **Kim waxed to like anchovies.*

As Pollard and Sag point out, *wax* is a relatively uncommon verb, used mostly with a very few adjectives like *poetical* and *lyrical*. In a theory where constituent structure information is available as readily as functional information in defining subcategorization requirements, the scarcity of such verbs is somewhat surprising. Our theory of subcategorization allows for these exceptional cases of categorial subcategorization, while reflecting the fact that in the normal case functional information is all that is relevant.

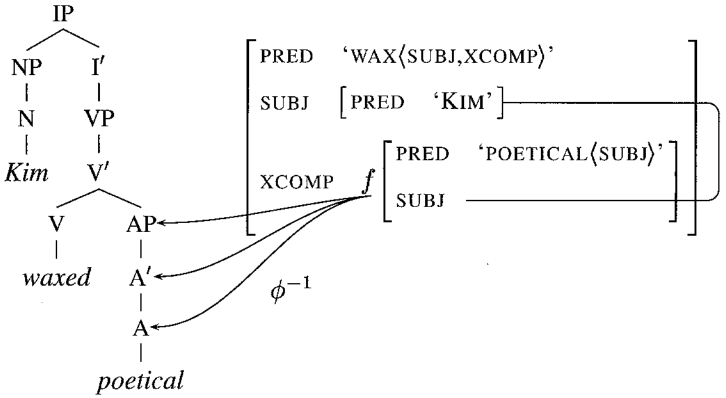
The c-structure/f-structure correspondence for an example like *Kim waxed poetical* is:

(94) *Kim waxed poetical.*



The ϕ function relating c-structure nodes to their f-structures is indicated by arrows. The *inverse* of this function, the ϕ^{-1} relation, is indicated by arrows pointing the opposite direction in (95):

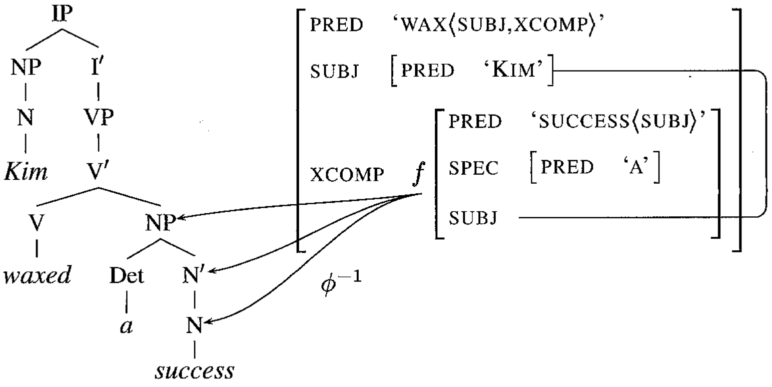
(95) *Kim waxed poetical.*



For each f-structure f, the *inverse correspondence* relation $\phi^{-1}(f)$ gives the c-structure nodes that are associated with that f-structure; the relation between f-structures and their corresponding c-structure nodes is therefore not a function, because more than one c-structure node can correspond to a single f-structure. For the f-structure labeled f in (95), $\phi^{-1}(f)$ yields the nodes labeled AP, A′, and A.

Example (96) gives the c-structure and f-structure for the ill-formed sentence *Kim waxed a success*:

(96) **Kim waxed a success.*



For this example, the f-structure labeled f is associated via the inverse ϕ correspondence with different nodes, those labeled NP, N′, and N.

We can now state the categorial requirement imposed on the complement of the verb *wax* by using the predicate CAT (Kaplan and Maxwell 1996), defined in terms of the inverse ϕ correspondence. The CAT predicate associates f-structures with the set of category labels of the nodes that correspond to them. Formally, CAT is defined in the following way:

(97) Definition of CAT:

$$\mathrm{CAT}(f) = \{c \mid \exists n \in \phi^{-1}(f).\mathit{category}(n) = c\}$$

This definition states that if f is an f-structure, CAT(f) is the set of category labels c such that for some node $n \in \phi^{-1}(f)$, the category label of n is c. In the wellformed example (95), the following is true:

(98) $\mathrm{CAT}(f) = \{\mathrm{AP}, \mathrm{A}', \mathrm{A}\}$

Thus, we posit the following categorial requirement, lexically associated with the verb *wax*:

(99) *wax*: AP $\in$ CAT(($\uparrow$ XCOMP))

This requirement ensures that one of the c-structure nodes associated with the XCOMP has the category AP, as required.

4.4. Functional Precedence

Functional precedence is a relation between two f-structures based on the c-structure precedence relation holding between the c-structure nodes corresponding to the two f-structures. Although it is based on the c-structural relation of precedence, it is different in interesting ways; differences between the two relations show up most clearly when an f-structure is related to discontinuous c-structure elements and when an f-structure does not correspond to any c-structure nodes.

Kameyama (1989) presents an analysis of Japanese pronominals that accounts for the distribution of overt pronominals as well as "null" pronouns, pronouns that appear at f-structure but not c-structure:

(100) a. ??*kare-no imooto-o Taroo-ga sewasiteiru (koto...)*
his-GEN sister-ACC Taro-NOM be.taking.care.of that
'...(that) Taro$_i$ was taking care of his$_i$ sister'

b. *Taroo-ga kare-no imooto-o sewasiteiru (koto...)*
Taro-NOM his-GEN sister-ACC be.taking.care.of that

In the unacceptable example in (100a), the pronoun *kare* precedes its antecedent *Taroo*, while example (100b), in which *Taroo* precedes the pronoun, is acceptable. In contrast, there are no restrictions on the relation between the null pronominal and the definite antecedent:

(101) a. *imooto-o* *Taroo-ga* *sewasiteiru* *(koto...)*
[$\emptyset$'s] sister-ACC Taro-NOM be.taking.care.of that
'...(that) Taro$_i$ was taking care of $\emptyset_i$ sister'

b. *Taroo-ga* *imooto-o* *sewasiteiru* *(koto...)*
Taro-NOM [$\emptyset$'s] sister-ACC be.taking.care.of that

Simplifying Kameyama's analysis somewhat, these and other examples show that the antecedent of an overt pronoun must precede the pronoun, while this constraint does not hold for null pronouns. The facts about pronominal binding in Japanese can be given a uniform explanation in terms of f-precedence:

(102) The antecedent of a pronoun must *f-precede* the pronoun.

This generalization about anaphoric binding in Japanese holds under the following definition of f-precedence (Kaplan and Zaenen 1989; Kameyama 1989):

(103) F-precedence, definition 1 (Kaplan and Zaenen 1989):

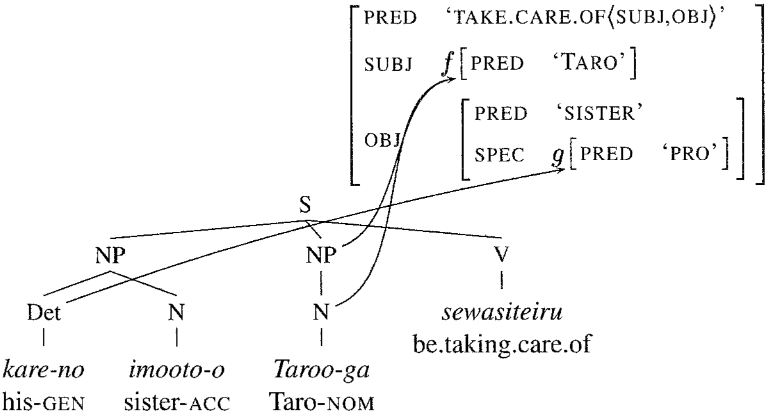
f *f-precedes* g ($f <_f g$) if and only if for all $n_1 \in \phi^{-1}(f)$ and for all $n_2 \in \phi^{-1}(g)$, n_1 c-precedes n_2.

This definition appeals to the *inverse relation* ϕ^{-1}, defined in Section 4.3 of this chapter, which associates f-structures with the c-structure nodes they correspond to. The definition of f-precedence states that an f-structure f f-precedes an f-structure g if and only if all of the nodes corresponding to f c-precede all of the nodes corresponding to g in the c-structure. The notion of c-precedence for c-structure nodes is the intuitively familiar notion of linear precedence, definable in the following terms (see also Partee et al. 1993, section 16.3.2):

(104) A c-structure node n_1 c-precedes a node n_2 if and only if n_1 does not dominate n_2, n_2 does not dominate n_1, and all nodes that n_1 dominates precede all nodes that n_2 dominates.

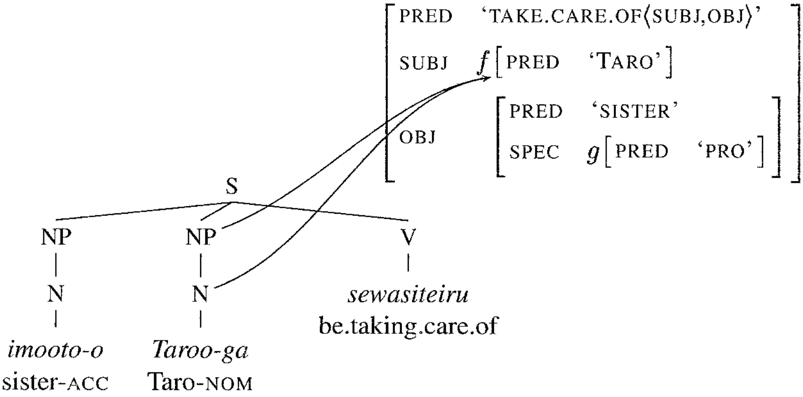
In the unacceptable example in (105), the f-structure g of the possessive pronoun *kare-no* f-precedes the f-structure f of the antecedent *Taro*, since all of the nodes corresponding to g (the Det node) precede the nodes corresponding to f (the NP and N nodes):

(105) ??*kare-no imooto-o Taroo-ga sewasiteiru (koto...)*
his-GEN sister-ACC Taro-NOM be.taking.care.of that
'...(that) Taro$_i$ was taking care of his$_i$ sister'



What is the situation with null pronominals, pronouns that do not appear at c-structure? Consider the c-structure and f-structure for example (106), which is like (105) except for having a null pronoun (realized only at f-structure) in place of the overt pronoun *kare-no*:

(106) *imooto-o Taroo-ga sewasiteiru (koto...)*
[∅'s] sister-ACC Taro-NOM be.taking.care.of that
'...(that) Taro$_i$ was taking care of ∅$_i$ sister'



Crucially, the f-structure of the null pronoun does not correspond to any c-structure node. According to the definition of f-precedence, null elements vacuously f-precede and are f-preceded by all other elements in the sentence; in particular, null pronouns vacuously satisfy the constraint of being preceded by their antecedents. Thus, a uniform statement of antecedent requirements for Japanese null and overt pronominals, together with the definition of f-precedence given in (103), makes the correct predictions.

The definition of f-precedence in (103) is the one originally proposed by Bresnan (1984) and used by Kameyama (1989) and by Kaplan and Zaenen (1989) in their analysis of word order in Dutch and German. A different definition has been proposed by Bresnan (2001b) in her analysis of weak crossover:

(107) F-precedence, definition 2 (Bresnan 2001b):

f f-precedes g if the rightmost node in $\phi^{-1}(f)$ precedes the rightmost node in $\phi^{-1}(g)$.

Clearly, this definition gives a different result for f-structures that do not correspond to any c-structure nodes, since such f-structures do not have a corresponding rightmost node. We will return to a discussion of these formulations of f-precedence in Chapter 14, Section 3.

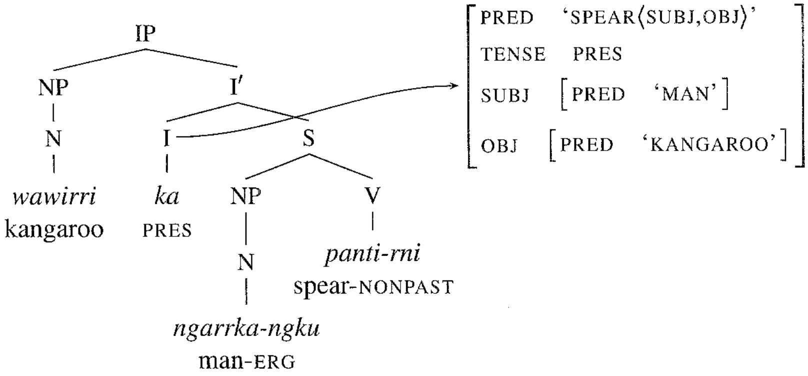
4.5. Empty Rule Nodes

In her analysis of Warlpiri, Simpson (1991) discusses gaps in morphological paradigms, showing that the Warlpiri AUX does not appear in what she calls the "null perfect aspect":

(108) *Japanangka-rlu* $\emptyset$ *pantu-rnu* *marlu*
Japanangka-ERG PERFECT spear-PAST kangaroo
'Japanangka speared the kangaroo.'

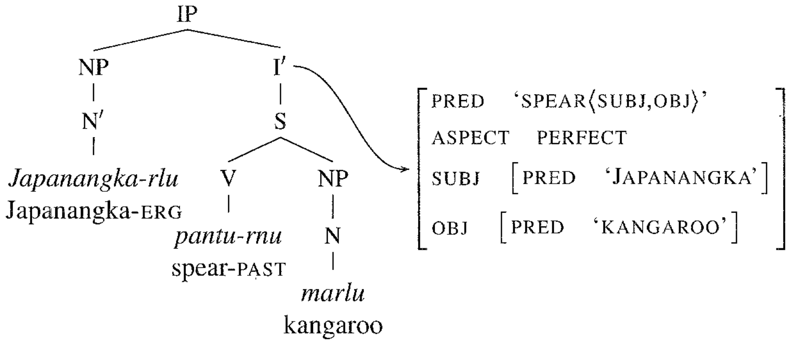
According to the generalization that Simpson formulates, there are two possibilities for phrasal expansion of I′. In example (109), with a "non-null" auxiliary, the first daughter of I′ is I:

(109) *wawirri ka ngarrka-ngku panti-rni*
kangaroo PRES man-ERG spear-NONPAST
'The man is spearing the kangaroo.'



In example (110), no auxiliary appears, and the sentence is interpreted as having perfect aspect:

(110) *Japanangka-rlu ∅ pantu-rnu marlu*
Japanangka-ERG PERFECT spear-PAST kangaroo
'Japanangka speared the kangaroo.'



The rule in (111) expresses the possibilities Simpson outlines for the phrase structure expansion for the category I′ in Warlpiri:

(111)
$$\text{I}' \longrightarrow \left\{ \begin{array}{c|c} \text{I} & \epsilon \\ \uparrow=\downarrow & (\uparrow \text{ ASPECT}) = \text{PERFECT} \end{array} \right\} \begin{array}{c} \text{S} \\ \uparrow=\downarrow \end{array}$$

In this rule, the symbol ϵ corresponds to the empty string and represents the absence of a phrase structure constituent. Importantly, the rule does not license the presence of an empty category or node in the c-structure tree: it simply constitutes an instruction to introduce some functional constraints in the absence of some overt word or phrase. No empty node is introduced into the tree, as example (110) shows.

The rule in (111) contains a disjunction: in one case, an auxiliary appears in I, while in the other case no auxiliary appears, and the sentence is interpreted as having perfect aspect. In fact, the rule in (111) is exactly equivalent to the one in (112):

(112)
$$\text{I}' \longrightarrow \left\{ \begin{array}{cc|c} \text{I} & \text{S} & \text{S} \\ \uparrow=\downarrow & \uparrow=\downarrow & \uparrow=\downarrow \\ & & (\uparrow \text{ ASPECT}) = \text{PERFECT} \end{array} \right\}$$

However, although the rule in (112) also represents the two possible phrase structure expansions of I$'$, it is more cumbersome and fails to express Simpson's generalization. By using the symbol ϵ, the generalization can be expressed concisely.

With the formal tools that have now been introduced, we are ready to begin a full-scale excursion into new linguistic realms. In the following chapters, we will explore other levels of linguistic structure and how they are best represented. We will then illustrate the formal architecture and linguistic theory in a discussion of the treatment of a representative set of linguistic phenomena.

7

BEYOND SYNTAX: NONSYNTACTIC STRUCTURES

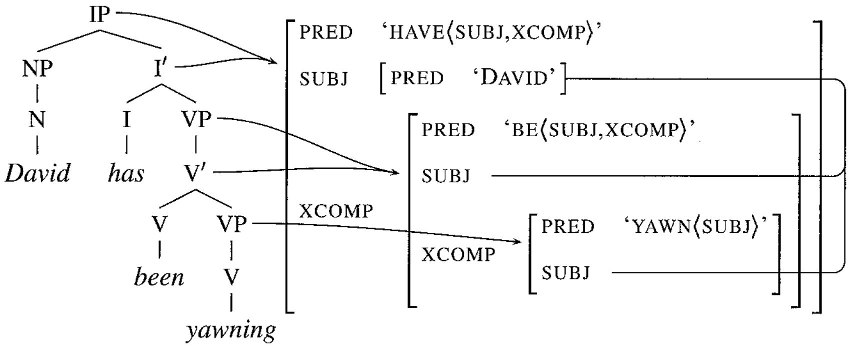
Thus far, our discussion has centered on two linguistic structures: the constituent structure represents phrasal groupings and precedence relations, and the functional structure represents more abstract functional syntactic predicate-argument relations. We now turn to the relation between these syntactic structures and other linguistic structures.

To illustrate these additional structures and how they relate to the structures we are already familiar with, Section 1 of this chapter introduces *morphosyntactic structure* and discusses its relation to c-structure and f-structure. In Section 2, we present the *projection architecture*, the theoretical architecture that allows the definition and expression of relations between linguistic structures. We explore *information structure* and its relation to other structures in Section 3. Finally, Section 4 examines cross-structural constraints and how they can be defined, and Section 5 discusses diagnostics and methods for positing new structures and defining their relations to other structures.

1. MORPHOSYNTACTIC STRUCTURE

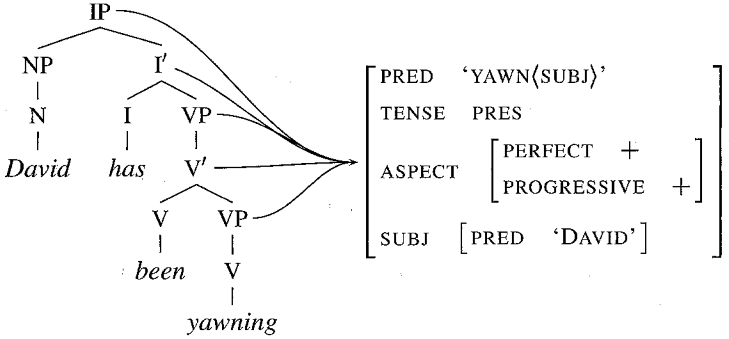
In some LFG work on the structure of the auxiliary system of English (Kaplan and Bresnan 1982; Falk 1984), English auxiliaries introduce their own f-structure and require an XCOMP complement verb phrase (see the discussion of *raising verbs* in Chapter 12). On this analysis, the c-structure and f-structure for a sentence like *David has been yawning* are:

(1) *David has been yawning.* (Kaplan and Bresnan 1982; Falk 1984)



However, although a multiclausal structure may be appropriate for English modals, there is no compelling evidence in English for a multiclausal structure for non-modal auxiliaries, and indeed it has often been argued that auxiliary verbs and their complements in other languages correspond to a single f-structure (Mohanan 1982; Ackerman 1987; Simpson 1991). We might instead propose a monoclausal functional structure for this example:

(2) *David has been yawning.*



However, this analysis does not ensure that the complement of each auxiliary has the correct verb form, a requirement enforced in the transformational literature by the "affix hopping" transformation (Chomsky 1957). Some means is necessary for ensuring that sentences like the following are ruled out:

(3) a. **David has be yawned.*

b. **David has being yawning.*

Thus, this approach must be supplemented with a theory of morphosyntactic features and the structure of the auxiliary system that enforces this requirement.

One method for encoding the relations between different auxiliary forms might be to define different c-structure categories for each auxiliary form and to constrain the order of the different auxiliary forms by specialized c-structure rules. For example, we might assume specialized constituent structure subcategories like $\mathrm{V}_{\textsc{progressive}}$ for the progressive auxiliary *is* in an example like *David is yawning*, and $\mathrm{V}_{\textsc{prespart}}$ for present participle forms. Under these assumptions, a c-structure rule like the following locally ensures that the progressive auxiliary *is* is followed by a verb in present participial form:

(4) $\mathrm{V}' \longrightarrow \mathrm{V}_{\textsc{progressive}}\ \mathrm{V}_{\textsc{prespart}}$

However, this approach is not sufficiently general. Although a c-structure rule can only constrain a local configuration, a mother node and its daughters, constraints on the form of the verbal complement of the progressive auxiliary must be enforced even when a nonlocal relation obtains between them:[1]

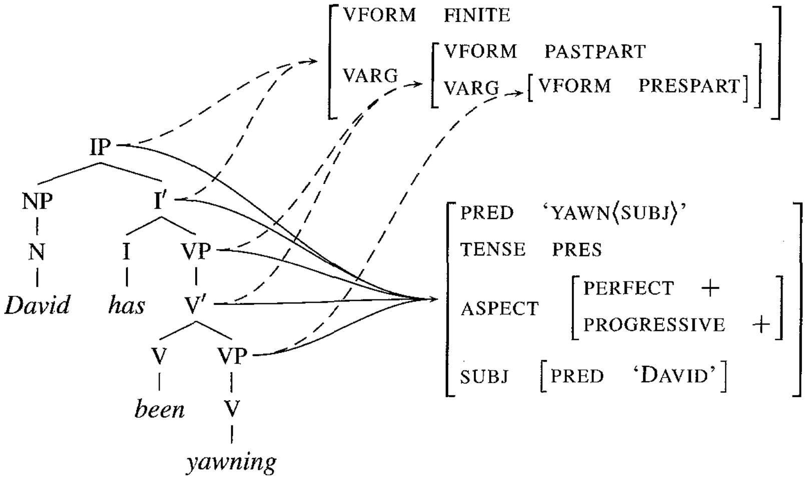
(5) *... there is no greater deed than to die for Iran. And dying/*died/*die they are, ...* (from Ward 1990)

In their analysis of the English auxiliary system, Butt et al. (1996) propose that an English sentence with multiple auxiliary verbs is in fact monoclausal, as in (2). They also propose that a separate morphosyntactic structure keeps track of the requirements on the form of a sequence of auxiliaries. The relation they propose between c-structure nodes and morphosyntactic structures is indicated by dotted lines in example (6) (page 180). Morphosyntactic structure records morphosyntactic dependencies of the auxiliary structure without requiring these distinctions to be reflected in functional structure. Butt et al. (1999) provide further discussion of morphosyntactic structure and the English auxiliary system.

Other work on the auxiliary systems of English and other languages makes different assumptions about the properties of morphosyntactic structure and its relation to c-structure and f-structure. In particular, Frank and Zaenen (2002) propose a treatment of the English auxiliary system that assumes a different set of relations between structures and a larger role for the c-structure component,

[1] In a different but related treatment of morphosyntactic structure, Frank and Zaenen (2002) discuss this issue and propose a treatment that makes use of complex phrase structure categories.

(6) *David has been yawning.* (Butt et al. 1996)



and Dyvik (1999) presents evidence that the Norwegian auxiliary system differs in interesting ways from the English system and should be treated differently.

In the following, we will see how additional linguistic structures like morphosyntactic structure can be defined and how their relation to the other structures can be constrained.

2. THE PROJECTION ARCHITECTURE

Just as the f-structure is related to the c-structure by a function from nodes of the c-structure tree to f-structures, in the same way other functions can be defined to relate parts of one structure to parts of another. As example (6) shows, the morphosyntactic structure proposed by Butt et al. (1996) is an attribute-value structure that is related to the c-structure by means of a function from c-structure nodes to morphosyntactic structures.

Kaplan (1987) originally proposed the *projection architecture* to define piecewise correspondences between linguistic structures that may be of very different formal types (see also Halvorsen and Kaplan 1988). The different structures are referred to as *projections*, and the functions relating the structures are referred to as *correspondence functions*. Chapter 5, Section 3.1 presented a notation for referring to the current c-structure node and its mother:

(7) the current c-structure node ("self"): $*$
the immediately dominating node ("mother"): $\widehat{*}$

The correspondence function ϕ relates nodes of the c-structure to their corresponding f-structures. In a similar way, we can define other functions relating other aspects of linguistic organization.

The correspondence function relating nodes of the c-structure to morphosyntactic structures represented by the dotted lines in (6) is called μ, following Butt et al. (1996). $\mu(*)$ is the morphosyntactic structure corresponding to the relevant daughter node, and $\mu(\widehat{*})$ is the morphosyntactic structure corresponding to the mother node. There is no standard abbreviation for these expressions, unlike the situation with f-structures, where $\uparrow$ is a standard abbreviation for $\phi(\widehat{*})$ and $\downarrow$ is a standard abbreviation for $\phi(*)$. Alternative notations for correspondence functions have been proposed, however; in some work, correspondence functions are represented as a subscript, with $*_\mu$ used as an alternative notation for $\mu(*)$. This notation is particularly common in discussions of structures projected from the f-structure, and we will adopt this variant notation in Chapter 9, in our discussion of semantic structure. For the remainder of this chapter, we will represent correspondence functions in the mathematically more standard way shown in (8) below.

In the phrase structure rule expanding V′, we can indicate that the morphosyntactic structure for the V′ and for the V are the same by writing:

(8)
$$\text{V}' \longrightarrow \begin{array}{c} \text{V} \\ \mu(\widehat{*}) = \mu(*) \end{array}$$

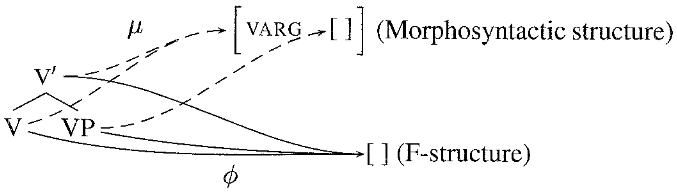
mother's morphosyntactic structure = self's morphosyntactic structure

In example (6) the V′ node dominating *been* expands in the following way:

(9)
$$\text{V}' \longrightarrow \begin{array}{c} \text{V} \\ \mu(\widehat{*}) = \mu(*) \\ \uparrow=\downarrow \end{array} \quad \begin{array}{c} \text{VP} \\ (\mu(\widehat{*})\ \textsc{varg}) = \mu(*) \\ \uparrow=\downarrow \end{array}$$

At f-structure, the V and VP daughters are both heads: both are annotated with $\uparrow=\downarrow$. However, at morphosyntactic structure the V is distinguished as the head of the structure, and the VP plays the VARG role:

(10)



By proposing new functions such as μ, we can define a relation between structures in terms of a function between parts of any linguistic structure and parts of another. For example, a function σ relating f-structure to semantic structure can also be defined, as we will see in Chapter 9.

Although morphosyntactic structure is represented here as an attribute-value structure like the f-structure, this is not a necessary property of the projection architecture. As discussed in Chapter 4, Section 5, LFG does not assume that every structure must be represented as a tree or as an attribute-value structure. Rather, LFG assumes a fine-grained relation between representations in which subparts of one representation are related to subparts of another one, just as the c-structure is related to the f-structure.

There is a clear relation between the projection architecture and other formal linguistic architectures. For example, Head-Driven Phrase Structure Grammar (Pollard and Sag 1994) and related theories represent linguistic information by a complexly structured attribute-value matrix, with subparts corresponding to different aspects of linguistic structure. A linguistic structure that is represented as an attribute-value structure and defined as a projection of the f-structure can be represented in an equivalent but less revealing way as a subpart of the f-structure it is associated with: the function defining the relation between the f-structure and the new projection can be reinterpreted as an attribute of the f-structure, with the new structure as its value. The significance of the projection architecture lies not in the additional formal power that it brings, but in its expressiveness and modularity; it allows for the relation between different linguistic components to be expressed while also retaining the identity of these components as separate structures representing different kinds of linguistic information.

3. INFORMATION STRUCTURE

Much linguistic research has centered on the nature of *information structure* and its relation to syntactic structure (Givón 1979; Andrews 1985; Sgall et al. 1986; Vallduví 1992; Kroeger 1993; King 1995). Some previous LFG literature has represented information structure by means of special attributes in the f-structure. In fact, in the preceding chapters this was done for consistency with the materials being cited. However, it is more satisfactory to distinguish between syntacticized discourse roles appearing in the f-structure (Bresnan and Mchombo 1987; Bresnan 2001b) and real discourse functions, which are not syntactic notions and which should therefore appear as a part of a separate structure.

3.1. Grammaticized Discourse Functions TOPIC and FOCUS

Bresnan and Mchombo (1987) propose that the phrase appearing in sentence-initial position in interrogative clauses in English and many other languages bears what they call a *grammaticized discourse function* in the f-structure, the FOCUS function, and that the relativized constituent in a relative clause bears a grammaticized TOPIC function. In a cleft construction, the clefted constituent bears both functions: as shown in (11b), it is the FOCUS in the matrix clause and the TOPIC in the embedded clause. Bresnan and Mchombo further propose that the same constituent cannot bear both TOPIC and FOCUS functions at the same level and derive a variety of very interesting predictions from this. In particular, this accounts for the pattern of ungrammaticality illustrated in (11):

(11) a. *I know [what you want].*
FOCUS

b. *It is my car [that you want].*
FOCUS TOPIC

c. *??I bought the car [that it was ___ [that you want]].*
TOPIC FOCUS

Example (11c) is unacceptable because the same constituent is both the TOPIC and the FOCUS of the subordinate clause.

The grammaticized discourse functions discussed by Bresnan and Mchombo have clear syntactic roles and should be represented syntactically, in the f-structure. However, other researchers have also represented information-structural categories such as discourse topic and focus in the functional structure. For instance, King (1995) examines the interaction of word order and the encoding of topic/focus relations in Russian, showing that different discourse functions are associated with different phrase structure positions; she represents discourse TOPIC and FOCUS as f-structure attributes. Similarly, Choi (1999) uses the features ±NEW and ±PROM to represent the information-structural status of the different parts of the f-structure and to explain the semantic and discourse effects of scrambling. In explicating her theory, Choi represents these features as a part of the f-structure, as if they were syntactic features; Choi notes, however, that a preferable theoretical architecture would represent features such as these separately, in a different structure representing discourse information.

3.2. Representing Information Structure

Butt and King (2000) make an explicit proposal to represent information structure within the projection architecture (see also Butt and King 1996). In an analysis of Hindi-Urdu word order and information structure, they propose four distinct discourse functions:

(12) Discourse functions (Butt and King 2000):

TOPIC, FOCUS, BACKGROUND, COMPLETIVE INFORMATION

Butt and King make the following proposals about these discourse functions and their phrase structure realization in Hindi-Urdu:

TOPIC is old or known information that is relevant in the current context. In Hindi-Urdu, the TOPIC appears in clause-initial position, in the specifier position of IP.

FOCUS is new and prominent information. It appears in preverbal position in Hindi-Urdu if there is only one focused element; additionally, phrases may be intonationally marked as focused when they are in their canonical positions.

BACKGROUND INFORMATION is like TOPIC in consisting of old or known information; it provides information as to how new information fits in with old information in an utterance. It appears postverbally in Hindi-Urdu.

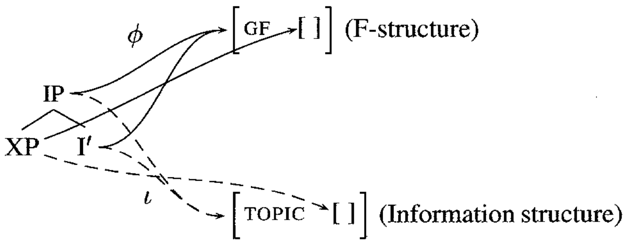
COMPLETIVE INFORMATION is new information that is not prominent in the discourse. It is not associated with a particular Hindi-Urdu phrase structure position, but occurs preverbally.

Butt and King propose that discourse structure, like morphosyntactic structure, is a projection of the c-structure, defined by the function ι from c-structure nodes to information structures. They propose c-structure rules for Hindi-Urdu like the following:

(13)
$$\text{IP} \quad \longrightarrow \quad \begin{array}{cc} \text{XP} & \text{I}' \\ (\iota(\hat{*})\ \text{TOPIC}) = \iota(*) & \iota(\hat{*}) = \iota(*) \\ (\uparrow\ \text{GF}) = \downarrow & \uparrow = \downarrow \end{array}$$

According to this rule, an IP consists of a phrase of any phrase structure category (XP) that plays some syntactic role GF in the f-structure and a discourse-theoretic role TOPIC in the information structure. The head daughter I′ corresponds to the same f-structure and information structure as the IP. This rule gives rise to the c-structure, f-structure, and information structure configuration shown in (14):

(14)



To exemplify their proposal, Butt and King (2000) propose the c-structure, f-structure, and information structure in (17) (page 186), in which the attribute COMP.INF represents completive information as defined earlier. The values of the information-structural functions TOPIC and FOCUS are associated with syntactic functions in the f-structure in accordance with the *extended coherence condition*, first proposed by Zaenen (1980) and discussed in detail by Fassi-Fehri (1988). Bresnan and Mchombo (1987) state the condition as follows:

(15) Extended Coherence Condition:

> FOCUS and TOPIC must be linked to the semantic predicate argument structure of the sentence in which they occur, either by functionally or by anaphorically binding an argument.

The structures in example (14) obey the extended coherence condition, since the topic *Naadyaa* also bears the grammatical function SUBJ, and the focus *bazaar-mẽ* is an adjunct phrase.

4. DEFINING RELATIONS BETWEEN STRUCTURES

Kaplan (1989) notes that there are two ways in which relations between structures can be defined: *codescription* and *description by analysis*. The formal difference between these two methods of description has not yet been fully explored. Ron Kaplan (p.c.) hypothesizes that description by analysis is the more powerful of the two, but a complete formal analysis and proof awaits further research.

4.1. Codescription

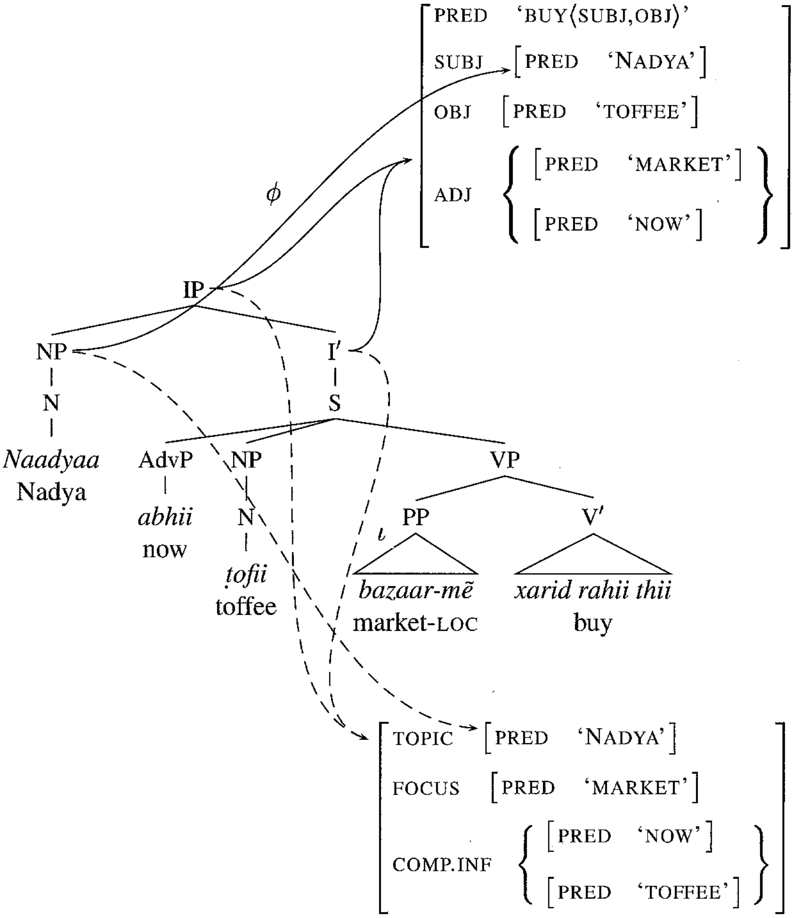
The analyses of morphosyntactic structure and information structure presented earlier exemplify *codescription*, where multiple structures are simultaneously described:

(16)
$$\begin{array}{lccc} \mathrm{V}' & \longrightarrow & \mathrm{V} & \mathrm{VP} \\ & & \mu(\hat{*}) = \mu(*) & (\mu(\hat{*})\ \textsc{varg}) = \mu(*) \\ & & \uparrow = \downarrow & \uparrow = \downarrow \end{array}$$

This rule simultaneously describes, or *codescribes*, the f-structure and the morphosyntactic structure.

Other linguistic architectures also make use of codescription. In Head-Driven Phrase Structure Grammar (Pollard and Sag 1994), for example, all linguistic structures are subparts of a sign, represented as an attribute-value matrix, and the various substructures are codescribed in lexical entries and rules and are built up

(17) *Naadyaa abhii ṭofii bazaar-mẽ xarid rahii thii*
Nadya now toffee market-LOC buy stative.FEM.SG be.PAST.FEM.SG
'Nadya was buying toffee at the market just now.'



simultaneously. Fenstad et al. (1987) provide a proposal for the architecture of LFG that involves codescription and that differs significantly in its formal architecture from standard LFG approaches; Fenstad et al.'s approach was independently developed at about the same time as the early development of HPSG, and it formally resembles HPSG much more closely than current standard versions of LFG. Besides the c-structure tree, Fenstad et al. propose a bipartite structure consisting of a syntactic representation like the f-structure and a semantic representation in attribute-value format. Additionally, they propose that other levels of structure, such as phonological structure, are also represented as subparts of the overall structure. The following is their representation of the semantic and syntactic structure for the sentence *John walks*:

(18)

$$\left[\begin{array}{ll}
\text{SITSCHEMA} & \left[\begin{array}{ll}
\text{REL} & \textit{walk} \\
\text{ARG.1} & \left[\begin{array}{ll}\text{IND} & \textit{John}\end{array}\right] \\
\text{LOC} & \left[\begin{array}{ll}
\text{IND} & \text{IND.1} \\
\text{COND} & \left[\begin{array}{ll}
\text{REL} & \circ \\
\text{ARG.1} & [\ ]_1 \\
\text{ARG.2} & l_d
\end{array}\right]
\end{array}\right] \\
\text{POL} & 1
\end{array}\right] \\
\text{FSTRUCT} & \left[\begin{array}{ll}
\text{SUBJ} & \left[\begin{array}{ll}
\text{PRED} & \text{'JOHN'} \\
\text{NUM} & \text{SG}
\end{array}\right] \\
\text{TENSE} & \text{PRES} \\
\text{PRED} & \text{'WALK}\langle\text{SUBJ}\rangle\text{'}
\end{array}\right]
\end{array}\right]$$

4.2. Description by Analysis

Relations between structures can also be defined by *description by analysis*, in which the description of one structure is obtained by analysis of another structure. A number of LFG proposals for semantic analysis and representation involve description by analysis: for instance, Halvorsen (1983) defines a semantic structure for an utterance on the basis of properties of its f-structure. Chapter 9 presents a theory of the syntax-semantics interface that differs from Halvorsen's proposals in a number of respects, and the theory presented there will be used in semantic analyses of the phenomena treated in the remainder of the book. Here we present Halvorsen's theory of the syntax-semantics interface as a particularly clear example of description by analysis.

Description by analysis involves the definition of properties of one structure based on the properties of another. Informally, a rule using description by analysis says something like: "whenever there is a structure with a certain property, it corresponds to another structure of a certain type." Halvorsen presents the following table of correspondences between semantic forms in the f-structure and meanings, represented as logical formulas:[2]

(19)

Semantic form in f-structure:	Meaning:
'EVERY'	$\lambda R.\lambda S.\mathit{every}(R, S)$
'HORSE'	$\lambda x.\mathit{horse}(x)$

Halvorsen also presents a set of rules for determining the meaning of an f-structure based on its semantic forms. The following is a simplified version of his Rule III, SPEC-PRED configuration:

(20) SPEC-PRED configuration:

If f_k is an f-structure of the form $\begin{bmatrix} \text{SPEC} & v_1 \\ \text{PRED} & v_n \\ \vdots & \vdots \end{bmatrix}$, then

$(\mathrm{M}_k\ \text{PREDICATE}) = \lambda P.\lambda Q.Q(P)$
$(\mathrm{M}_k\ \text{ARG1}) = \mathrm{M}_1$ (semantic structure of the SPEC)
$(\mathrm{M}_k\ \text{ARG2}) = \mathrm{M}_2$ (semantic structure of the PRED)

Let us see how this rule operates in the analysis of a particular f-structure for the noun phrase *every horse*. For this simple example, we will leave out most of the detail in the f-structure:

(21) $f_1 \begin{bmatrix} \text{SPEC} & \text{'EVERY'} \\ \text{PRED} & \text{'HORSE'} \end{bmatrix}$

We apply the rule in (20) in the following way. First, we inspect the f-structure in (21), noting that it is of the form required by the rule in (20). Therefore, we introduce a constraint on the structure M_1, the semantic structure corresponding to f_1: it must contain the attribute PREDICATE with value $\lambda P.\lambda Q.Q(P)$. We also add the additional constraint that M_1 must contain the attribute ARG1, and we determine the value of that attribute by consulting the table in (19); in a similar way, we add a constraint on the value of ARG2. This yields the following set of constraints on M_1:

(22) $(\mathrm{M}_1\ \text{PREDICATE}) = \lambda P.\lambda Q.Q(P)$
$(\mathrm{M}_1\ \text{ARG1}) = \lambda R.\lambda S.\mathit{every}(R, S)$
$(\mathrm{M}_1\ \text{ARG2}) = \lambda x.\mathit{horse}(x)$

[2] Halvorsen's analysis has been considerably simplified for presentation here, leaving aside many of the details of the meanings and rules that he proposes. Further, we have modified his analysis by assuming that a word like *every* is treated as a *generalized quantifier* (Barwise and Cooper 1981) specifying that the *every* relation holds between two properties R and S. We will discuss quantification in more detail in Chapter 9. The lambda operator λ is explained in Chapter 9, Section 4.1.

These equations describe the following semantic structure M_1 for the f-structure f_1:

(23)
$$\mathrm{M}_1 \begin{bmatrix} \textsc{predicate} & \lambda P.\lambda Q.Q(P) \\ \textsc{arg1} & \lambda R.\lambda S.\mathit{every}(R,S) \\ \textsc{arg2} & \lambda x.\mathit{horse}(x) \end{bmatrix}$$

We also need a set of rules for interpreting this structure: Halvorsen (1983) presents a set of rules that produce formulas of intensional logic from these semantic structures. Among them are rules like the following:

(24) PREDICATE and $\textsc{arg}_i$

Apply the PREDICATE to $\textsc{arg}_n$, where n is the polyadicity of the predicate. Apply the result to $\textsc{arg}_{n-1}$ and so on, until $n = 0$.

Applying this rule to the semantic structure for *every horse* in (23) produces the following result, which is just what is required for the phrase *every horse*:

(25) $\lambda P.\lambda Q.[Q(P)](\lambda x.\mathit{horse}(x))(\lambda R.\lambda S.\mathit{every}(R,S))$
$\equiv \lambda S.\mathit{every}(\mathit{horse},S)$

In sum, linguistic analyses formulated according to the description-by-analysis paradigm operate by imposing constraints on one structure on the basis of an inspection of one or more other structures.

5. DEFINING NEW STRUCTURES AND RELATIONS

LFG research has explored a number of different linguistic levels and their representations. We have seen that it is possible to define new linguistic levels or projections by specifying the properties of a structure and its functional relation to other structures. How can it be determined when it is necessary to postulate the existence of a new linguistic level, distinct from those already assumed within the theory? And how can we define new constraints within a level, or constraints that hold across levels, within this architecture?

5.1. Postulating a New Structure

When confronted with new and unfamiliar linguistic phenomena, it is sometimes tempting to assume that they constitute a new and independent linguistic system, governed by its own rules, rather than simply hitherto undiscovered manifestations of familiar phenomena and familiar rules. Whether or not to add a new representation for a new type of linguistic information must be considered carefully. Sadock (1991, page 214-215) states that:

> Any postulated component should be of a kind that one might want in a full description of the basic facts of language, regardless of the way in which the modules are related to one another.

It is difficult to provide strict criteria for when it is appropriate to postulate the existence of a new level. Sadock rightly cautions against assuming "levels that are entirely abstract," encoding relations like coindexing or traces; of course, any new level that is assumed should deal with a cohesive set of linguistic phenomena that are found in multiple languages and are demonstrably and cohesively related to one another.

5.2. Defining a New Relation on an Existing Structure

In some cases, constraints on grammatical phenomena may involve more than one kind of linguistic information, and thus constraints must be stated by reference to more than one aspect of linguistic structure. This in itself does not constitute motivation for introducing yet another linguistic level incorporating information from other levels.

Indeed, many phenomena that we have already examined are of this nature. For example, the correlation between phrase structure positions and grammatical functions involves reference both to c-structure and to f-structure, and is captured by annotations on c-structure rules making reference to f-structure properties. In fact, any phenomenon that is treatable by codescription — for example, the relation between phrase structure position, morphosyntactic form, and syntactic function that we explored in our discussion of morphosyntactic structure — has this character.

One strategy for the formal treatment of linguistic phenomena involving relations and entities at more than one level of representation is to define a new relation on an existing structure in terms of relations that are native to another structure. An example of this is *f-precedence*, a relation between f-structures that is defined in terms of the precedence relation that holds between c-structure nodes.

As discussed in Chapter 6, Section 4.4, defining a precedence relation derivative of c-structure properties on the f-structure, a level at which precedence is not native, has very interesting results that are not available when the standard c-structural precedence relation is considered. In particular, f-structures that do not correspond to any c-structure nodes also take part in f-precedence. Also, constituents in different parts of the phrase structure tree may correspond to the same f-structure, and thus the c-structure nodes corresponding to one f-structure may be interleaved with the nodes of another; in this case, no f-precedence relation may hold between the two f-structures. Thus, constraining precedence relations in terms of f-precedence instead of c-structure precedence gives rise to a richer set of predictions for how the linear order of arguments can be constrained.

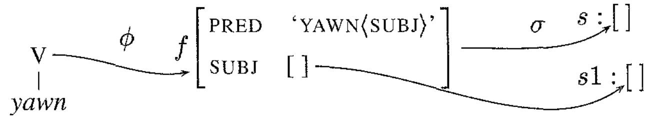
5.3. Defining Interstructural Constraints

It is often useful to refer to the relation between one representation and another: to speak, for example, of the semantic structure that corresponds to a particular c-structure node, or the morphosyntactic structure corresponding to a particular f-structure. How can we talk about the relation between two structures related by functional projections?

5.3.1. Structural Correspondence by Composition

In Chapter 9, we will introduce *semantic structure* as a level of linguistic structure. We assume that semantic structure is directly related to f-structure; the correspondence function σ relates f-structures to semantic structures in the following way:

(26)



In the configuration depicted in (26), the correspondence function ϕ relates the c-structure node labeled V to the f-structure labeled f. The σ function defines a direct relation between f-structures and semantic structures: in (26), the semantic structure corresponding to f is labeled s, and the semantic structure corresponding to f's subject is $s1$. The following facts hold of the configuration in (26):[3]

(27) $\phi(\mathrm{V}) = f \qquad \sigma(f) = s$

Given the functions ϕ and σ, we can define a function between c-structure nodes and semantic structures as the *composition* of the two correspondence functions, $\sigma \circ \phi$.[4] In the case at hand, we can apply the composition function $\sigma \circ \phi$ to the c-structure node V to get the semantic structure s:

(28) $\sigma \circ \phi(\mathrm{V}) = \sigma(\phi(\mathrm{V})) = s$

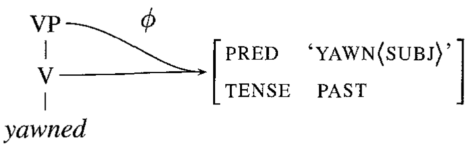
[3] As mentioned in Section 2 of this chapter, an alternative subscript notationf_{σ}, exactly equivalent to $\sigma(f)$, is often used for semantic structures.

[4] The *composition* of two functions is obtained by taking the result of applying the first function to its argument and then applying the second function to that result (see Partee et al. 1993, page 33). The composition of two functions is also a function.

In this way, we can define a function between c-structure nodes and their corresponding semantic structures that is mediated by the f-structure, in terms of the function ϕ from c-structure nodes to f-structures and the function σ from f-structures to semantic structures. More generally, we can exploit the projection architecture to define relations between structures that are not directly related via correspondence functions by defining new composite functions to relate the two levels.

Subtle issues arise in the design of the optimal projection architecture for linguistic description. In particular, care must be taken in arranging the various linguistic structures properly, since distinctions that are relevant at one level but collapsed at another cannot be reintroduced in further projections. Consider the configuration in (29):

(29)

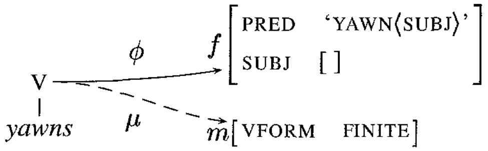


At c-structure, the distinction between the nodes labeled VP and V is clearly represented: two different c-structure nodes are involved. At f-structure, the distinction between the two nodes is collapsed, since the two nodes correspond to the same f-structure. Thus, any level of linguistic representation that must make reference to a distinction between the V and the VP node cannot be a projection of the f-structure, since the distinction between VP and V is collapsed at that level and cannot be reintroduced.

5.3.2. Structural Correspondence by Inverse Correspondence

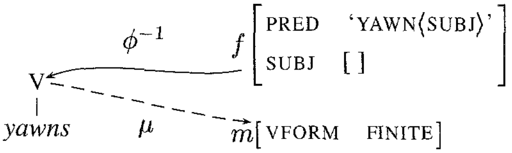
The diagram in (30) indicates a relation between f-structure and morphosyntactic structure that can be defined in terms of nodes of the c-structure:

(30)



In Chapter 6, Section 4.3, we discussed *inverse correspondences* between structures. The inverse of the ϕ function, ϕ^{-1}, relates f-structures to the set of c-structure nodes that correspond to them:

(31)



In the configuration depicted in (31), the f-structure labeled f is related via the ϕ^{-1} correspondence to a single c-structure node, the one labeled V. In turn, V is related via the μ correspondence function to the morphosyntactic structure labeled m:

(32) $\phi^{-1}(f) = \{\mathrm{V}\}$
$\mu(\mathrm{V}) = m$

Given these two facts, we can define a relation between f-structures and the morphosyntactic structures that indirectly correspond to them. We do this by extending the definition of the μ projection so that it holds of *sets* of nodes as well as of individual nodes. The μ function from a set of c-structure nodes gives the set of morphosyntactic structures that the nodes are related to:

(33) Correspondence function from a set of nodes C:

$$\mu(C) \equiv \{m \mid \exists c \in C.\mu(c) = m\}$$

This definition states that the morphosyntactic projection of a set of c-structure nodes C is the set of all morphosyntactic structures m that are related to some c-structure node in C. We extend the other correspondence functions in a similar manner: any correspondence function from one structure to another can be applied to a set of structures to give the set of corresponding structures.

Given this definition and the configuration displayed in (31), we can now apply the μ correspondence function to the nodes in the inverse ϕ correspondence of the f-structure f:

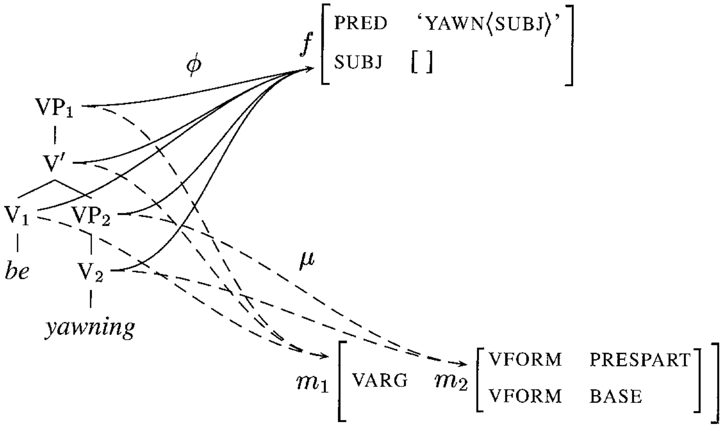
(34) $\mu(\phi^{-1}(f)) = \{m\}$

Since the inverse correspondence is not in general a function, as noted in Chapter 6, Section 4.3, there are often several c-structure nodes that correspond to a particular f-structure. In example (37), for example, all five nodes of the c-structure tree are related to the same f-structure f; the nodes of the tree have been labeled VP$_1$, VP$_2$, V$'$, V$_1$, and V$_2$ for ease of reference. The inverse correspondence ϕ^{-1} of the f-structure labeled f in (37) is given in (35). Additionally, in the diagram in (37), the equations given in (36) hold:

(35) $\phi^{-1}(f) = \{\mathrm{VP}_1, \mathrm{V}_1, \mathrm{V}', \mathrm{VP}_2, \mathrm{V}_2\}$

(36) $\mu(\mathrm{VP}_1) = m_1 \quad \mu(\mathrm{V}') = m_1 \quad \mu(\mathrm{V}_1) = m_1$
$\mu(\mathrm{VP}_2) = m_2 \quad \mu(\mathrm{V}_2) = m_2$

(37)



Therefore, exactly as we would expect, the morphosyntactic structures that are related to f in (37) are:

(38) $\mu(\phi^{-1}(f)) = \{m_1, m_2\}$

Thus, the projection architecture allows for the statement of complex relations among linguistic structures, and for constraining these relations in appropriate ways.

6. FURTHER READING AND RELATED ISSUES

Besides the work cited above, there has been much interesting work on nonsyntactic structures and their relation to c-structure and f-structure. Butt and King (1998) present an LFG analysis of the phonology-syntax interface. Research on argument structure and its relation to syntax has been done by Arka and Manning (1998) and Manning (1996a,b); argument structure will be discussed at greater length in Chapter 8. Hong (1991) and King (1997) discuss information structure and its place in the overall architecture of LFG. Work on semantic structure and meaning will be discussed in detail in Chapter 9. Finally, Andrews and Manning (1999) present a new and very different view of the relations between linguistic structures within the LFG framework, an approach that formally resembles the work of Fenstad et al. (1987) as well as more recent work in Head-Driven Phrase Structure Grammar (Pollard and Sag 1994).

8

ARGUMENT STRUCTURE AND MAPPING THEORY

Chapter 7 explored the relations between LFG's syntactic structures and other linguistic structures such as morphosyntactic structure and information structure. In this chapter, we focus on *argument structure* and its relation to syntax, particularly concentrating on the role of argument structure in determining the syntactic functions of the semantic arguments of a predicate. We will examine different views of the representation and content of argument structure, and outline the theory of the relation between semantic roles and grammatical functions.

1. SYNTAX, SEMANTICS, AND ARGUMENT STRUCTURE

That syntax and semantics are separate structures, separately constrained, can be easily demonstrated by a consideration of verbs like *eat* or *rain* in examples like:

(1) a. *Chris ate.*

b. *It rained.*

Semantically, the verb *eat* refers to a two-place relation between an AGENT, or individual who eats, and a PATIENT or entity that is eaten. Syntactically, however, the verb *eat* has an intransitive use, illustrated in (1a), where its only argument is the AGENT; the PATIENT argument is understood but syntactically unexpressed. Evidence that the verb *eat* is intransitive in this case is given by the possibility of *out*-prefixation, as discussed by Bresnan (1980). Bresnan shows that only intransitive verbs can participate in *out*-prefixation:

(2) a. *The lamp shines./The lamp outshines the candle.*

b. *The Brownies found the treasure./*The Brownies outfound the Girl Scouts in the treasure hunt.*

The verb *eat* can participate in *out*-prefixation, indicating that it has an intransitive use:

(3) *Chris outate David.*

Thus, the verb *eat* can be syntactically monovalent, requiring only a SUBJ argument, whereas it is semantically bivalent, denoting a relation between two entities.

Conversely, a verb like *rain* in example (1b) requires a SUBJ argument, but denotes a predicate that does not take a semantic argument; it does not make sense to ask **Who/what rained*, or to replace the SUBJ argument of *rain* by any other argument:

(4) **He/David/Something rained.*

Here too, syntactic and semantic valence are different: *rain* requires a syntactic argument that does not play a semantic role. These simple examples make it clear that syntactic and semantic argument structures are different and must be represented separately.

The influence of semantic structure on syntactic form is shown by the predictable nature of the syntactic realization of arguments of newly coined verbs. Alsina (1996, Chapter 1) illustrates this point by considering the nonce word *obliquate*, which he defines as meaning "to build or place in an oblique orientation." As he notes, the possibilities for syntactic expression of the arguments of this verb are limited:

(5) a. *Jim obliquated the door of the closet.*

b. **The door of the closet obliquated Jim.*

Because *obliquate* is a made-up verb, anyone encountering these sentences will not have heard them or any sentences like them before. Thus, any constraints on the way the arguments of this verb are realized cannot be due to any syntactic or morphological priming. Instead, the pattern of acceptability in (5) must be ascribed to constraints imposed by the semantics of the verb.

What aspects of semantic structure are relevant for constraints on syntactic form? Pinker (1989) outlines two main hypotheses (see also Mohanan 1994): on the first hypothesis, which Pinker calls "Unrestricted Conceptual Representation," any kind of semantic or culturally salient distinction can be reflected in syntax and can constrain syntactic form and syntactic relations. The second hypothesis, the "Grammatically Relevant Subsystem" hypothesis, is more restricted: only certain semantic features, those represented at what is generally termed *argument structure*, are relevant for syntax. As Pinker notes, the second hypothesis is more satisfying. It allows not only for a more precise characterization of argument structure, but also for the prospect of an explanatory theory of the relation between argument structure and syntax, as well as a realistic theory of language learning. Work in LFG adheres to the "Grammatically Relevant Subsystem" paradigm, with different researchers adopting different views of argument structure and the subset of semantic information it contains.

In the following, we will explore some influential theories of argument structure and its representation in LFG, and we will examine some proposed constraints on the relation between argument structure and syntax. We will concentrate primarily on the theory of argument structure and its role in mapping between argument roles and grammatical functions.[1]

2. CONTENT AND REPRESENTATION OF ARGUMENT STRUCTURE

It is generally agreed that argument structure contains some amount of semantic information, but researchers do not agree on how much. Some researchers claim that there is very little semantic information in argument structure; other researchers hold that argument structure is semantically richer, individuating semantic roles like AGENT, THEME, or GOAL, and even drawing aspectual and other semantic distinctions.

The representation of argument structure has also evolved since the early days of LFG. Kaplan and Bresnan (1982) propose that the semantic form value of the PRED attribute encodes the relation between semantic roles and syntactic functions (see also Bresnan 1982a,c), as in the following semantic form for the verb *give*:

[1] In the following, we will not provide much discussion of the interaction of argument structure with other grammatical processes. See Manning (1996b,a) for a very interesting theory of the syntactic role of argument structure and its interactions with functional structure. Manning proposes that what he calls *construal* processes such as binding, determination of imperative addressee, and control in adverbial and complement clauses are sensitive to relations at argument structure rather than a syntactic representation like functional structure.

(6) Semantic form for *give* (Kaplan and Bresnan 1982):

$$\text{`give}\langle \begin{array}{ccccc} \text{SUBJ} & & \text{OBJ} & & \text{OBL}_{\text{GOAL}} \\ — & , & — & , & — \\ \text{AGENT} & & \text{THEME} & & \text{GOAL} \end{array} \rangle\text{'}$$

This expression indicates that *give* has three arguments that are associated with the roles of AGENT, THEME, and GOAL.[2] Such roles are often referred to as *thematic roles*. These three arguments are also associated with syntactic functions: the AGENT with SUBJ, the THEME with OBJ, and the GOALwith OBL_{GOAL}. As Kaplan and Bresnan (1982) note, the angled brackets are supposed to remind us of the parentheses commonly used in logical expressions, so that the semantic form is thought of as expressing a kind of logical formula encoding aspects of the meaning of the sentence as well as the relation between thematic roles and their syntactic functions.

More recent work in LFG that provides an explicit view of the content and representation of argument structure stems primarily from two rather different sources: Zaenen (1993), Alsina (1993, 1996), Ackerman (1992), Ackerman and Moore (2001), and others adopt and modify the Proto-Role argument classification proposals of Dowty (1991), while Butt (1996, 1998), Broadwell (1998), and others present analyses based on the Conceptual Semantics framework of Jackendoff (1983, 1990).

Alsina (1993, 1996) assumes that argument structure constitutes a refinement of the traditional notion of semantic form, appearing in the f-structure as the value of the PRED feature. In line with much research on argument structure and thematic roles, Alsina relies on an ordering of the arguments of a predicate according to a semantically motivated hierarchy of thematic roles, as described in Section 4.3 of this chapter: for example, arguments bearing the AGENT role are higher on the hierarchy than PATIENTS. Alsina claims, however, that the difference between thematic roles like AGENT and PATIENT is best represented semantically rather than at argument structure; in his theory, such distinctions do not play a direct role in determining the syntactic functions of arguments. Thus, Alsina claims that the argument structure of a predicate consists simply of the predicate name and a list of argument positions ordered according to the thematic hierarchy, with no association of arguments with particular thematic roles.

Besides the hierarchical ordering imposed on the arguments of a predicate, Alsina builds on work by Dowty (1991) in distinguishing between *proto-agent* and *proto-patient* properties of arguments. Unlike Dowty, however, Alsina assumes a set of criterial definitions for proto-agent and proto-patient arguments; for example, he proposes that a causer argument with volitional involvement is necessarily classified as a proto-agent, while an "incremental theme" argument

[2] Recall from Chapter 2, Section 3.6.1 that syntactic arguments that are not associated with semantic roles are represented outside the angled brackets.

is a proto-patient. Arguments that do not meet these criteria are not assigned proto-role status. Alsina (1996) provides the following argument structure representations for the verbs *come* and *give*:

(7) Argument structures for *come* and *give* (Alsina 1996):

$$\left[\text{PRED} \quad \text{`COME}\langle[\text{P-P}]\;[\;]\rangle\text{'}\right]$$

$$\left[\text{PRED} \quad \text{`GIVE}\langle[\text{P-A}]\;[(\text{P-P})]\;[\text{P-P}]\rangle\text{'}\right]$$

The verb *come* has two arguments: the individual who comes is the proto-patient, and the destination argument bears no proto-role status. The verb *give* has three arguments: a proto-agent or giver; a recipient argument that is optionally classified as a proto-patient, depending on whether or not it is causally affected; and a second proto-patient, the entity that is given. On Alsina's view of argument structure, no other semantic information appears in argument structure besides the proto-role status of each argument and the ordering of arguments according to the thematic hierarchy.

Zaenen (1993) also builds on Dowty's proposals to define argument structure in terms of semantically based properties of a predicate. Unlike Dowty, however, Zaenen does not assume that these properties relate to semantic entailments of particular uses of the predicate. Instead, Zaenen proposes that predicates have lexically specified, semantically definable characteristics that she terms *dimensions*; for instance, a predicate has a volitional dimension if it can be used with an adverb like *on purpose*. The existence of a volitional dimension in the argument structure of a verb does not entail that every use of the verb denotes a volitional act; rather, the verb denotes an act that *can* be volitional. Based on these semantic dimensions, Zaenen (1993) proposes a set of role-defining properties that the arguments of a verb can bear, and the assignment of a grammatical function to an argument is made on the basis of these properties. Ackerman (1990), Ackerman and Moore (1999), Joshi (1993), and Markantonatou (1995) also use Dowty's Proto-Role classifications in their treatments of argument structure.

In contrast with Proto-Role-theoretic accounts, Butt (1996) proposes a semantically richer theory of argument structure, based on the Conceptual Semantics approach of Jackendoff (1990). On her view, the argument structure for a verb like *give* is:

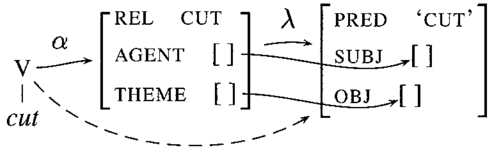
(8) Lexical Conceptual Structure for *give* (Butt 1996):

$$\left[\begin{array}{l} \text{give} \\ \quad \left[\begin{array}{l} \text{CS}([\alpha], \text{GO}_{\text{Poss}}([\;], \text{TO}[\beta])) \\ \text{AFF}^{+}([\;]^{\alpha},) \\ \text{ASP}(\_\_, \_\_, \_\_) \end{array}\right]_{\text{EVENT}} \end{array}\right]$$

This argument structure representation is tripartite. The first line represents the Thematic Tier, a broad representation of those aspects of the meaning of *give* that are relevant at argument structure: here, that the "cause" relation CS holds between an actor α and an event $\mathrm{GO}_{\mathrm{Poss}}$ in which some entity is possessed by a beneficiary β. The second line represents the Action Tier, which encodes information about motion or location and highlights and defines thematic relations like AGENT and PATIENT; here, the second line indicates that α is the first argument of the AFF^{+} predicate and is therefore the AGENT. As Butt points out, the Action Tier represents roughly the same subset of information that is taken to be relevant for the Proto-Role theory of Dowty (1991). An innovation in Butt's approach is the postulation of an Aspectual Tier, represented on the third line, which indicates whether the beginning, middle, and end of the event are lexically specified; in the case of the verb *give*, neither the inception, duration, nor end point of the event is intrinsically specified. As Butt shows, aspectual information is important in characterizing the mapping relation for complex predicates in Urdu.

Butt et al. (1997) assume a more traditional view of argument structure in which the arguments of a predicate bear particular thematic roles like AGENT, GOAL, and PATIENT. Unlike many other approaches, they provide an explicit characterization of the relation between argument structure and other grammatical structures, situated within the assumptions of the projection architecture (Chapter 7, Section 2):

(9) C-structure, f-structure, and argument structure for *cut* (Butt et al. 1997):



$$\phi = \alpha \circ \lambda$$

On this view, argument structure bears a direct relation to the c-structure, defined by the correspondence function α from c-structure nodes to argument structures. In turn, the f-structure is related to argument structure by the function λ, relating argument structures to f-structures. Thus, the familiar ϕ mapping between c-structure nodes and f-structures, described in Chapter 4, Section 1, is redefined as the composition $\alpha \circ \lambda$ of the α and λ functions. Based on these structures and the relations between them, Butt et al. propose a theory of argument-function mapping in which candidate mappings defined by the λ function are evaluated and the highest-ranking candidate mapping is selected.

Other theories of the nature and content of argument structure have also been proposed; Alsina (1996, Chapter 1) includes a useful overview and summary of some of this work.

3. GRAMMATICAL FUNCTION ALTERNATIONS

A primary focus of LFG theory since its inception has been the characterization of grammatical function alternations and the relation between different syntactic realizations of a predicate and its arguments. LFG adheres to the *Principle of Direct Syntactic Encoding* (Kaplan and Bresnan 1982), which states that syntactic operations may not alter the subcategorization requirements specified by a predicate:

(10) Principle of Direct Syntactic Encoding:

No rule of syntax may replace one function name by another.

Within the constraints imposed by the Principle of Direct Syntactic Encoding, there has been a steady evolution in the LFG view of grammatical function alternations. In early formulations, grammatical function alternations like the active/passive relation were characterized by *lexical rules* relating lexical entries corresponding to different diatheses of a verbal form. As Bresnan (1990) notes, however, the theory of lexical rules did not provide a completely general picture of the linking between grammatical functions and semantic roles or of grammatical function alternations and their interactions, and the need for a more general theory subsequently became apparent. In more recent work, grammatical function alternations have been the focus of *mapping* or *linking theory*, a theory of the mapping between thematic roles and grammatical functions. A number of different versions of mapping theory have been proposed. We will discuss some of the more influential views in the following sections; Section 8 of this chapter contains a brief overview of other proposals.

3.1. Lexical Rules

Kaplan and Bresnan (1982) represent the relation between grammatical functions and thematic roles as a part of the semantic form value of the PRED feature. Different assignments of grammatical functions to the same thematic roles (as in, for example, the active and passive versions of a verb) are treated by means of *lexical redundancy rules*, rules encoding a regular lexical relation between different forms. The architecture of a richly structured lexicon and an articulated theory of relations among lexical forms is a hallmark of LFG, and the relations between lexical forms and the structure of the lexicon continue to be a focus of theoretical work.

Unlike more recent work on lexical rules and lexical relations, early LFG treatments of lexical rules focused primarily on the analysis of grammatical function alternations such as the active/passive relation, and the rules that were formulated to treat these alternations had a distinctly transformational character. Bresnan

(1982c) proposes the following lexical rule for relating passive verbs to their active counterparts:

$$\text{(11)}\quad \text{`L}\langle\ \underset{\text{AGENT}}{\overset{\text{SUBJ}}{\text{—}}}\quad \underset{\text{THEME}}{\overset{\text{OBJ}}{\text{—}}}\ \rangle\text{'} \rightarrow \text{`L}\langle\ \underset{\text{AGENT}}{\overset{\text{OBL}_{\text{AGENT}}/\emptyset}{\text{—}}}\quad \underset{\text{THEME}}{\overset{\text{SUBJ}}{\text{—}}}\ \rangle\text{'}$$

or in abbreviated form:

(12) SUBJ→ OBL$_{\text{AGENT}}$
OBJ→ SUBJ

The fact that relation-changing lexical rules were formulated as operations on one lexical form to produce another is telling of the transformationally oriented way in which these rules tended to be viewed in the early days of LFG: though the rules were formulated to apply to structures pairing thematic roles with grammatical functions, thematic roles did not usually play much of a role in regulating or constraining the rules. Instead, research focused primarily on the morphological and syntactic characteristics of grammatical function alternations rather than on a theory of the alignment of thematic roles with grammatical functions.

3.2. A Theory of Argument-Function Mapping

That there are regularities in the mapping between argument structure roles and grammatical functions has been clear since the pioneering work of Gruber (1965) and Fillmore (1968); an early attempt to characterize these regularities was made by Ostler (1979), who proposed that the relation between grammatical function and thematic roles is given by a set of *linking rules*. Within LFG, Zaenen and Maling (1983) were among the first to propose a set of *Association Principles* relating grammatical functions to thematic roles. For example, they give the following Association Principles for Icelandic:

(13) Icelandic Association Principles (Zaenen and Maling 1983):

1. Agents are linked to SUBJ. (universal)
2. Casemarked Themes are assigned to the lowest available grammatical function. (specific to Icelandic)
3. If there is only one thematic role, it is assigned to SUBJ; if there are two, they are assigned to SUBJ and OBJ; if there are three, they are assigned to SUBJ, OBJ, and the secondary object function OBJ2. This principle applies after principle 2 and after the assignment of restricted GFs.

Zaenen et al. (1985) provide further discussion of association principles for Icelandic and also give a similar set of association principles for German.

From this beginning, a vibrant body of research has developed. We will only have room here to scratch the surface of the complex issues that are involved; see Section 8 of this chapter for a brief discussion of some views other than those presented in the following.

4. ARGUMENT CLASSIFICATION

In subsequent research on argument mapping, it was found that grammatical functions like SUBJ and OBJ can be grouped into natural classes, and thematic roles can be associated with these classes rather than specific roles. This allows for thematic roles to be only partially specified as associated with particular grammatical functions. Levin (1986) was the first to propose such restrictions on the mapping between thematic roles and grammatical functions as:

(14) THEME is unrestricted.

As discussed in Chapter 2, Section 1.4, the (semantically) unrestricted functions are SUBJ and OBJ; these are the functions that can be filled by an argument with any thematic role or by an expletive or semantically empty argument. Thus, on Levin's view, arguments can be linked to a class of grammatical functions such as the unrestricted functions, as in the case of her THEME rule.

Constraints such as these delimit a range of possible mappings between thematic roles and grammatical functions. Principles of wellformedness of the mapping relation further specify and constrain these associations, determining which particular grammatical function is associated with each role. One such principle is the *Subject Condition* (Chapter 2, Section 1.5), which holds in some (and perhaps all) languages; this principle requires each verbal predicate to have an argument associated with the SUBJ function. Another is the principle of Function-Argument Biuniqueness, originally proposed by Bresnan (1980) (see also Bresnan and Kanerva 1989). In the following definition, $g_1 \ldots g_n$ is a list of grammatical functions and $P(1 \ldots m)$ is a semantic form with a list of arguments $1 \ldots m$:

(15) Function-Argument Biuniqueness:

$G = g_1 \ldots g_n$ is a possible grammatical function assignment to $P(1 \ldots m)$ if and only if the mapping from $1 \ldots m$ to G defined by $i \mapsto g_i$ is injective (one-to-one and onto).

This principle rules out a situation where the same grammatical function is associated with more than one thematic role, or where the same role is associated with more than one grammatical function.[3]

Continuing the line of research forged by Zaenen and Maling (1983), Levin (1986), and others, Bresnan and Kanerva (1989) present a comprehensive theory of mapping from thematic roles to grammatical functions, usually called *lexical mapping theory*. A number of other approaches have been explored, some of which will be briefly described at the end of this chapter; in the following, we will use Bresnan and Kanerva's proposal to present the basic concepts of the theory.

4.1. Cross-Classification of Grammatical Features

Bresnan and Kanerva (1989) propose that the syntactic functions SUBJ, OBJ, OBJ$_\theta$, and OBL$_\theta$ are decomposable into the feature specifications ±R(estricted) and ±O(bjective) in the following way:

(16) −R: semantically unrestricted functions SUBJ and OBJ; arguments with any thematic role (or with no thematic role) can fill these functions (see Chapter 2, Section 1.4).

+R: semantically restricted functions OBJ$_\theta$ and OBL$_\theta$: only arguments bearing particular thematic roles can fill these functions. For example, only an argument with the role of AGENT can appear as an OBL$_{\text{AGENT}}$.

−O: nonobjective (non-object-like) functions SUBJ and OBL$_\theta$; these functions roughly correspond to external arguments of a predicate.

+O: objective functions OBJ and OBJ$_\theta$.

These features cross-classify the grammatical functions in the way represented in (17):

(17)

	−R	+R
−O	SUBJ	OBL$_\theta$
+O	OBJ	OBJ$_\theta$

4.2. Intrinsic Classifications

Bresnan and Kanerva (1989) assume that at argument structure, the arguments of a predicate are associated with thematic roles such as AGENT, PATIENT, and THEME. The relation between argument roles and grammatical functions is expressed by associating the features ±R and ±O to thematic roles, given a theory

[3] In recent work, the Function-Argument Biuniqueness principle has been challenged by Alsina (1995, 1996).

of intrinsic and default classification of grammatical functions depending on the particular roles involved.

Bresnan and Kanerva propose that arguments bearing the AGENT role are intrinsically classified as in (18):

(18) AGENT encoding:
$$\begin{array}{c} \text{AGENT} \\ | \\ [-\text{o}] \end{array}$$

This rule states that the AGENT argument must be a −o function: that is, the AGENT may not be associated with an objective or object-like function. Therefore the AGENT role must be associated either with the SUBJ or an OBL function

specifically the $\text{OBL}_{\text{AGENT}}$ function.

This classification of the AGENT argument as filling a nonobjective −o role is valid for many but not all languages. Kroeger (1993) argues that in Tagalog, the AGENT can be syntactically realized as a nonsubject term, or object. Manning (1996b) makes a similar argument for Inuit and other ergative languages, and

4.3. Default Classifications

In addition to the intrinsic classifications, Bresnan and Kanerva (1989) propose a set of default assignments depending on the relative ranking of thematic roles on the thematic hierarchy defined in (21):

(21) Thematic hierarchy (Bresnan and Kanerva 1989):

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

These default classifications apply in every instance except when a conflict with intrinsic specifications would result. The thematic role of a predicate that is highest on the thematic hierarchy, notationally represented as $\hat{\theta}$, is classified as unrestricted:

(22)
$$\begin{array}{c}\hat{\theta}\\ | \\ [-\mathrm{R}]\end{array}$$

Other roles are classified as restricted:

(23)
$$\begin{array}{c}\theta\\ | \\ [+\mathrm{R}]\end{array}$$

The intrinsic and default argument classifications are further constrained by well-formedness conditions on the relation between thematic roles and grammatical functions. The *Subject Condition* (Chapter 2, Section 1.5) requires each verbal lexical form to have a SUBJ. The requirement of Function-Argument Biuniqueness [see (15) above] governs the relation between thematic roles and grammatical functions, requiring each thematic role selected by a predicate to be realized by a unique grammatical function.

5. THE ACTIVE/PASSIVE ALTERNATION

As a first illustration of the theory, we will consider the active and passive versions of a verb like *select* in examples like (24):

(24) a. *David selected Chris.*

b. *Chris was selected.*

The intrinsic classifications associated with the AGENT and PATIENT arguments are:

(25)

	select	⟨	AGENT	PATIENT	⟩
			│	│	
intrinsic:			[−O]	[−R]	

As shown in (18), the AGENT is intrinsically associated with the feature −O, meaning that it must bear a nonobjective function, either the SUBJ or the OBL$_{AGENT}$ function. The PATIENT is −R, meaning that it must fill an unrestricted function, either the SUBJ or the OBJ.

The default argument classifications now apply, based on the position of the AGENT and PATIENT arguments on the thematic hierarchy:

(26)

	select	⟨	AGENT	PATIENT	⟩
			│	│	
intrinsic:			[−O]	[−R]	
defaults:			[−R]		

The AGENT argument ranks higher on the hierarchy than the PATIENT argument, and its intrinsic classification −O is compatible with the default feature assignment −R. Arguments lower on the thematic hierarchy are associated with the default feature +R unless this would lead to a contradiction; in this case, a conflict with the intrinsic feature −R would be produced, so the default +R classification does not apply to the PATIENT.

The combination of intrinsic and default classification rules gives the result that the AGENT role is associated with a function that is [−O] and [−R]: the function SUBJ. The PATIENT is associated with a [−R] function, either the SUBJ or the OBJ:

(27)

	select	⟨	AGENT	PATIENT	⟩
			│	│	
intrinsic:			[−O]	[−R]	
defaults:			[−R]		
			SUBJ	SUBJ/OBJ	

These assignments are constrained by wellformedness conditions on the argument mapping relations, abbreviated as "w.f." in (28). The Subject Condition is satisfied, since the AGENT argument is realized as the SUBJ. Function-Argument Biuniqueness precludes the presence of two different roles linked to the same grammatical function: therefore, it is not possible for both roles to be realized as SUBJ, and so we must choose the OBJ possibility for the PATIENT role:

(28)

	select	⟨	AGENT	PATIENT	⟩
			│	│	
intrinsic:			[−O]	[−R]	
defaults:			[−R]		
			SUBJ	SUBJ/OBJ	
w.f.			SUBJ	OBJ	

Thus, the theory correctly predicts that the AGENT argument of *select* is the SUBJ, and the PATIENT is the OBJ:

(29) *David selected Chris.*

$$\begin{bmatrix} \text{PRED} & \text{`SELECT}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`DAVID'}\end{bmatrix} \\ \text{OBJ} & \begin{bmatrix}\text{PRED} & \text{`CHRIS'}\end{bmatrix}\end{bmatrix}$$

We now examine the passive counterpart of this active sentence. Bresnan and Moshi (1990) propose that passivization serves to suppress the thematically highest argument, making it unavailable for linking.[4]

(30)
$$\begin{array}{lcccc} \text{select} & \langle & \text{AGENT} & \text{PATIENT} & \rangle \\ & & | & & \\ & & \emptyset & & \end{array}$$

Thus, only the PATIENT argument may be assigned argument classification features. Application of the intrinsic and default classifications gives the following result:

(31)
$$\begin{array}{llcccc} & \text{select} & \langle & \text{AGENT} & \text{PATIENT} & \rangle \\ & & & | & | & \\ \text{intrinsic:} & & & \emptyset & [-\text{R}] & \\ \text{defaults:} & & & & & \end{array}$$

The PATIENT is intrinsically associated with the feature −R, meaning that it must be realized as either a SUBJ or an OBJ. Although the AGENT argument is suppressed by the passive rule, preventing it from being linked to a syntactic function, it remains the highest thematic role of the predicate, $\hat{\theta}$. According to the default classification rule for arguments not bearing the highest thematic role, then, the PATIENT argument should be marked with +R. However, as in the active version, this would clash with its intrinsic specification −R, and therefore the default classification rule does not apply.

We assume that the Subject Condition holds in English, requiring every verbal lexical form to have a subject. This forces us to choose the SUBJ realization for the PATIENT, since otherwise the sentence would have no subject:

[4] On this view of passivization, the AGENT may be expressed only as a modifying adjunct, not as an oblique argument of the verb.

(32)

	select	⟨ AGENT	PATIENT ⟩
		|	|
intrinsic:		∅	[−R]
defaults:			
			SUBJ/OBJ
w.f.			SUBJ

This produces the correct result, that the PATIENT argument of passive *select* is the SUBJ:

(33) *Chris was selected.*

$$\begin{bmatrix} \text{PRED} & \text{`SELECT}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`CHRIS'} \end{bmatrix} \end{bmatrix}$$

6. LOCATIVE INVERSION

The locative inversion construction is analyzed in detail by Bresnan and Kanerva (1989) (see also Bresnan 1994) in an exploration of the effect of information at other levels of linguistic structure on mapping principles. The locative inversion construction is a particularly fruitful arena for such study, since locative inversion involves the interaction of a particular information-structural property, presentational focus, with mapping principles.

Bresnan and Kanerva (1989) examine argument structure alternations with verbs like Chicheŵa *-im-* 'stand'. Example (34) illustrates the uninverted construction, with a THEME subject and a LOCATIVE oblique argument:[5]

(34)

nkhandwe	*y-a-im-a*	*pa-m-chenga*
9.fox	9SUBJ-PERFECT-stand-INDICATIVE	16-3-sand

'The fox is standing on the sand.'

Example (35), with a preposed LOCATIVE and a focused THEME argument, illustrates the locative inversion construction:

(35)

pa-m-chenga	*p-a-im-a*	*nkhandwe*
16-3-sand	16SUBJ-PERFECT-stand-INDICATIVE	9.fox

'On the sand is standing the fox.'

In the locative inversion construction, as Bresnan and Kanerva (1989) demonstrate, the noun phrase *nkhandwe* 'fox' bears the OBJ relation, while the locative phrase *pa-m-chenga* 'on the sand' bears the SUBJ relation. Evidence for this comes

[5] Numbers in the glosses indicate the noun class of the arguments.

from patterns of verb agreement: in example (34), the verb agrees in noun class with the subject *nkhandwe* 'fox', which is class 9, whereas in example (35), the verb shows class 16 agreement with the locative subject *pa-m-chenga* 'on the sand'. Additionally, Bresnan and Kanerva describe syntactic restrictions on nonfinite verb phrase modifiers, which may not appear with an overt subject. In verb phrase modifiers involving locative inversion, it is the locative subject that must be nonovert:

(36)	*m-nkhalangó*	*[m-ó-khál-á*	*mi-ângo]*$_{VP}$
	18-9-forest	18SUBJ-INF-live-INDICATIVE	4-lion

'in the forest where there live lions'

Constructions involving subject raising in Chicheŵa also provide evidence that the locative phrase is the SUBJ and the THEME phrase is the OBJ in examples like (35).

Uninverted examples such as example (34) are analyzed as follows:

(37)	stand	⟨ THEME	LOCATIVE ⟩
		│	│
intrinsic:		[−R]	[−O]
defaults:			[+R]
		SUBJ/OBJ	OBL$_{LOC}$
w.f.		SUBJ	OBL$_{LOC}$

The THEME argument is intrinsically classified with the −R feature, and the locative argument is intrinsically −O. Default argument classification rules add the additional information that the locative is +R. Because the locative argument is −O and +R, it is realized as an oblique locative argument OBL$_{LOC}$. According to intrinsic and default classifications, the THEME may be either SUBJ or OBJ; since the Subject Condition holds in Chicheŵa, it must be realized as SUBJ. This gives the desired result: the THEME is SUBJ, and the locative argument is OBL$_{LOC}$.

Bresnan and Kanerva (1989) propose a special default linking rule for presentational focus constructions, including locative inversion constructions like (35). In presentational focus constructions, marked by the presence of a semantic feature [*f*], a special default rule ensures that a locative −R argument appears:

(38)	[*f*]	LOCATIVE
		│
		[−R]

As Bresnan and Kanerva note, the [*f*] feature localizing the focus information is associated with the THEME argument in Chicheŵa, but may be associated with other arguments (or even the verb) in other languages.

Given this special default rule, the presentational focus/locative inversion construction is analyzed in the following way:

(39)	stand ⟨	THEME	LOCATIVE ⟩
		|	|
intrinsic:		[−R]	[−O]
defaults:			[−R]
		SUBJ/OBJ	SUBJ
w.f.		OBJ	SUBJ

The locative argument is intrinsically classified as −O, as above; in the presentational focus context, it is also associated with the special default assignment −R, so that it must be the SUBJ. The THEME is intrinsically associated with the −R feature, so that it may be realized either as the SUBJ or as the OBJ. However, Function-Argument Biuniqueness forbids the presence of two arguments linked to the SUBJ function, so that the THEME argument must bear the OBJ function in (35), as desired.

7. COMPLEX PREDICATES

It is sometimes assumed that argument linking is a lexical process, one that applies to the thematic roles of a lexical predicate to determine the syntactic realizations of those roles. Butt (1996) and Alsina (1996) challenge this view by providing evidence that monoclausal syntactic structures can be associated with constructions consisting of more than one potentially nonadjacent word.

Butt (1996) discusses the "permissive construction" in Urdu, illustrated in (40):

(40) *Anjum-ne Saddaf-ko ciṭṭhi likhne dii*
Anjum-ERG Saddaf-DAT letter.NOM write.PARTICIPLE let
'Anjum let Saddaf write a letter.'

The permissive construction consists of two verbs, the main verb (here *likhne* 'write') and the light verb *dii* 'let'. Butt shows that at the level of c-structure , the permissive construction involves either a V′ constituent containing the main and light verbs, or a VP constituent containing the main verb and the OBJ argument:

Although the main verb/light verb combination need not form a c-structure constituent, and in fact need not appear adjacent to each other, Butt (1996) shows that the permissive construction is monoclausal at functional structure, involving a single *complex predicate* constructed from the two verbal forms. Evidence for this comes from verb agreement patterns in Urdu: the Urdu verb agrees with the nominative argument that is highest on a hierarchy of grammatical functions. If none of its arguments is nominative, the verb shows default (third person masculine singular) agreement. In example (40), the verb *dii* 'let' agrees with the nominative feminine singular noun *ciṭṭhi* 'letter':

(42) *Anjum-ne Saddaf-ko ciṭṭhi likhne dii*
Anjum-ERG Saddaf-DAT letter.FEM.SG.NOM write.PARTICIPLE let.FEM.SG
'Anjum let Saddaf write a letter.'

This shows that *ciṭṭhi* 'letter' is a syntactic argument of the main predicate, since agreement is only possible between a predicate and one of its syntactic arguments. Butt provides further evidence from constructions involving control and anaphora that points to the same conclusion: the verbs *likhne* 'write' and *dii* 'let' combine to form a single syntactic predicate at f-structure, taking a single array of syntactic arguments.

As Butt's work makes clear, complex predicate formation — and, therefore, argument linking — cannot be defined by reference to the argument roles of a single word or even a single phrase structure constituent; data discussed by Alsina (1996) point to a similar conclusion. This and other research has led to a new view of mapping theory, termed *functional mapping theory* or simply *mapping theory* by some researchers in acknowledgment of the fact that the theory cannot apply exclusively to individual words.

The data presented by Butt (1996) and Alsina (1996) have been taken as conclusive evidence that mapping theory may not apply in the lexicon. However, alternative views have also been suggested. Kaplan and Wedekind (1993) propose an analysis of complex predicates in Urdu that makes use of the *restriction operator* (Chapter 6, Section 3.4) to provide an alternative lexical entry for verbs, like *likhne* 'write' in the examples in (40–41), which combines with light verbs like *dii* 'let' to form a complex predicate. The restriction operator is used in a lexical rule that provides a new lexical entry for *likhne* 'write': the SUBJ of the original entry appears as a thematically restricted OBJ of the new entry. This new lexical item must be used with light verbs that introduce a new subject, as *dii* 'let' does.

Frank (1996) also addresses the issue of complex predicate formation, noting that the assumption that French and Italian complex predicates are formed in a similar manner does not allow for a revealing analysis of the differences between complex predicates in the two languages. She proposes a new sort of lexical rule that combines two verbs to produce a new lexical specification for complex predicates such as the Urdu examples described earlier. On this view, complex predicate formation is lexically specified, though involving possibly discontinuous expressions consisting of more than one word.

8. FURTHER READING AND RELATED ISSUES

This brief discussion cannot do justice to the wide range of literature on the theory of the mapping between argument structure and functional structure. For more information on the nature and motivation for argument structure, see Simpson (1991, Chapter 1), who presents an interesting discussion of thematic roles, syntactic functions, and the regularities that hold between syntax, semantics, and argument structure. Bresnan (2001b, Chapter 14) provides an overview discussion of mapping theory, discussing a range of alternative approaches and providing pointers to related literature. Falk (2001) also provides a useful overview, including a very interesting discussion of linking in ditransitive verbs.

More recent alternatives to the theory of argument mapping originally proposed by Bresnan and Kanerva (1989) have been very influential in subsequent work on mapping theory. Bresnan and Zaenen (1990) present an analysis of unaccusative verbs and resultatives that makes use of the standard ±R, ±O argument classifications, but eschews the use of default principles in argument mapping. On their view, arguments like AGENT and PATIENT are intrinsically associated with the ±R, ±O argument-classifying features, and further restrictions are imposed on the basis of how "patientlike" the argument role is. As above, these feature assignments are further constrained by wellformedness conditions, the Subject Condition and Function-Argument Biuniqueness. Butt (1998) builds on and refines the approach presented by Bresnan and Zaenen, introducing an additional semantic parameter of aspectual affectedness in analyzing complex predicates.

A number of other influential approaches to argument mapping differ significantly from the theories mentioned earlier. The following is a partial list of papers addressing other aspects of the theory or other formal approaches to the theory of argument mapping.

Zaenen (1993) presents a theory of argument mapping based on her theory of argument structure, discussed in Section 2 of this chapter, according to which arguments of a predicate are associated with certain semantically defined proto-agent and proto-patient properties. Zaenen proposes that the assignment of intrinsic feature classifications to an argument is based on whether the argument has a preponderance of proto-agent or proto-patient properties. Her analysis successfully accounts for the syntactic and semantic differences between two classes of unaccusative verbs in Dutch.

Alsina (1996) argues for a very different feature decomposition of grammatical functions, making use of the features ±SUBJ, indicating whether or not the argument is "subject-like," and ±OBL, indicating whether the argument is a direct (−OBL) or an oblique argument. Alsina also eliminates the Function-Argument Biuniqueness condition, arguing that its elimination allows the correct treatment of, among other constructions, reflexive clitics in some Romance languages.

8.1. Locative Inversion

Her and Huang (1998) present an account of locative inversion in English and Chinese that involves a morpholexical operation that assigns argument classification features associated with the locative inversion construction. Morimoto (1999) proposes an Optimality-theoretic account of argument mapping in locative inversion and related constructions.

8.2. Complex Predicates

One of the earliest treatments of complex predicates in an LFG setting was proposed by Ishikawa (1985), who discussed Japanese morphological causatives, passives, potentials, and desideratives, and proposed to treat them as involving a type of raising that can involve the OBJ as well as the SUBJ of the complement verb. Ackerman and Webelhuth (1996, 1998) present a theory of complex predicates in which a single lexical item can be expressed by more than one morphological word. Matsumoto (1996) also explores the notion of word at different levels of linguistic representation, providing an illustrative examination of complex predicates in Japanese; Matsumoto (1998) examines Japanese causatives and proposes a parameter of semantic variation in the typology of causatives.

Andrews and Manning (1999) present another approach to the analysis of complex predicates and serial verbs. Their approach is LFG-based but architecturally quite different, involving structures resembling those used in Head-Driven Phrase Structure Grammar much more closely. Serial verbs have also been studied by Bodomo (1996, 1997), who analyzes serial verbs in Dagaare based on more standard LFG assumptions.

8.3. COMP, XCOMP, and OBL

There has been comparatively little work on mapping of the grammatical functions COMP and XCOMP. Zaenen and Engdahl (1994) were among the first to propose a detailed theory of argument mapping to COMP and XCOMP, assuming that these arguments bear the thematic role PROPOSITION and are intrinsically associated with the −o feature.[6] Culy (1994) proposes linking rules for COMP in English and also explores the possibility that discourse functions like TOPIC and FOCUS can be subcategorized and can participate in linking theory. Butt (1996) and Alsina (1996) also discuss the application of linking theory to XCOMP, COMP, and raising verbs.

Other work concentrates on mapping theory for oblique arguments. Markantonatou and Sadler (1995) provide a mapping theory encompassing indirect arguments, and Her (1999) discusses the dative alternation in Chinese in the context of a new set of mapping proposals.

8.4. Linking in Nominal Predicates

Iida (1987) and Saiki (1987) discuss deverbal nominals in Japanese, the realization of grammatical functions in nominals, and the role of argument structure; Markantonatou (1992, 1995) presents an analysis of linking in deverbal nominals in Modern Greek. Laczkó (1995) presents a detailed analysis of the syntax of Hungarian noun phrases, including a theory of linking of nominal arguments.

8.5. Optimality-Theoretic Accounts

Most recently, work within LFG's mapping theory has been conducted within an Optimality-theoretic framework (Prince and Smolensky 1993), discussed in Chapter 15, Section 3. Lødrup (1999) provides an Optimality-theoretic analysis of the Norwegian presentational focus construction. Asudeh (2001a) proposes an Optimality-theoretic account of argument linking in Marathi, noting some consequences for Dowty's (1991) theory of proto-roles. Morimoto (1999) presents an Optimality-theoretic account of locative inversion and argument reversal, and this work is continued in Morimoto (2000).

[6] Zaenen and Engdahl (1994) also discuss argument mapping and nonthematic or semantically empty arguments, which they claim are present at argument structure with the intrinsic feature−R.

9

MEANING AND SEMANTIC COMPOSITION

We now embark on an exploration of the theory of the relation between syntax and meaning, examining how the meaning of an utterance is determined on the basis of its syntactic structure. Early work in LFG proposed that the semantic form value of the f-structure PRED represented certain aspects of the meaning of the f-structure. More recent work assumes the existence of a *semantic structure*, related to the f-structure by a correspondence function. In this chapter, we briefly review some previous LFG approaches to semantics and the syntax-semantics interface. We then present the *glue approach* to semantic composition, the approach we adopt in the remainder of the book. This approach gives a firm theoretical foundation for the discussions in the next five chapters.

1. SYNTAX AND SEMANTIC INTERPRETATION

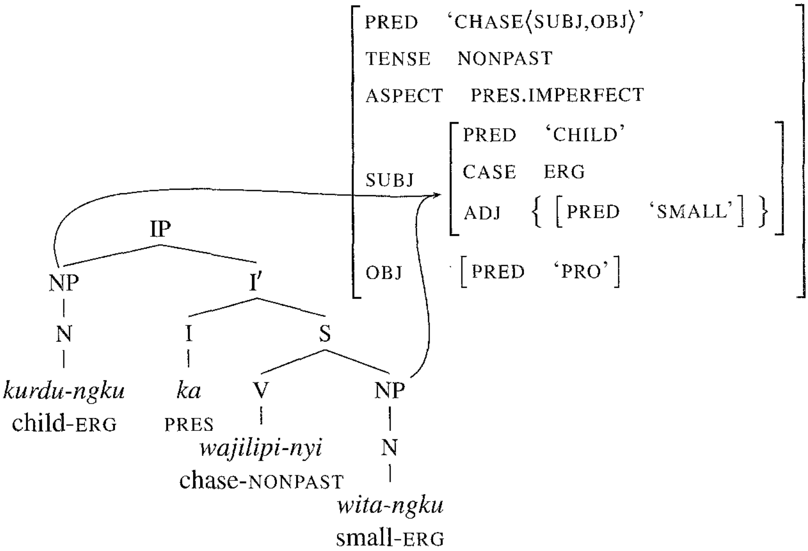
The central problem of semantic interpretation is plain: people have no trouble understanding the meanings of sentences in their language that they have never heard before. Thus, people must be able to determine the meaning of a novel sentence on the basis of the meanings of its component parts. The idea that the meanings of larger pieces are assembled from the meanings of the smaller pieces that make them up is known as the Principle of Compositionality, and is generally attributed to Gottlob Frege (though the accuracy of this attribution has been disputed; see, for example, Janssen 1997). An adequate treatment of linguistic meaning requires, then, a theory of the meanings of the most basic units of a sentence, together with a theory of how these meanings are put together.

A commonly accepted version of the Principle of Compositionality is the *rule-to-rule* hypothesis, which states that "a very close relation is supposed to exist between the rules of the syntax and the rules of the semantics" (Bach 1989). This means that each syntactic rule for combining syntactic units to form a larger syntactic unit corresponds to a semantic rule that tells how to put the meanings of those units together to form the meaning of the larger unit. The syntactic rules in question are often assumed to be phrase structure rules, so that instructions for combining meanings are paired with instructions for forming constituent structure phrases.

However, this version of the rule-to-rule hypothesis is actually just one way of enforcing an orderly theory of semantic composition, one in which the intuition that the meaning of a whole depends on the meanings of its parts is made explicit by defining the relevant parts as phrase structure constituents. In fact, research on the syntax-semantics interface and semantic composition in LFG has shown that we can remain faithful to the Principle of Compositionality without assuming that rules for putting meanings together must depend on phrasal primitives such as linear order and phrasal dominance.

Since the inception of semantic research in LFG, researchers have presented convincing arguments that semantic composition should proceed mainly by reference to functional structure rather than constituent structure organization. As argued by Fenstad et al. (1987, Chapter 2), the units that are primarily relevant for semantic composition are units at f-structure and not necessarily at c-structure. For example, as we have seen, a semantic unit may correspond to discontinuous portions of the c-structure tree. Example (23) of Chapter 4, repeated in example (1) (page 219), shows that the Warlpiri analog of the English phrase *small child* need not form a c-structure constituent; the noun *kurdu-ngku* 'child' appears at the beginning of the sentence, while its modifier *wita-ngku* 'small' appears at the end. However, rules for semantic composition in both English and Warlpiri treat the subject of the Warlpiri sentence *kurdu-ngku ka wajilipi-nyi wita-ngku* and the English sentence *The small child is chasing it* as an f-structure constituent and as a semantic unit; in fact, the rules for semantic composition in the two languages are remarkably similar, considering the great differences between the two languages at the constituent structure level. Guiding semantic composition by reference to f-structure and not c-structure relations brings out and clarifies crosslinguistic

(1) *kurdu-ngku ka wajilipi-nyi wita-ngku*
child-ERG PRES chase-NONPAST small-ERG
'The small child is chasing it.'



commonalities in principles of semantic composition, commonalities that would otherwise be obscured by properties of the more variant constituent structure.

Even given the centrality of functional structure in semantic composition, however, it must be kept in mind that semantic composition does not depend solely on functional structure. For example, as pointed out by Halvorsen (1983) and Fenstad et al. (1987), intonation has a strong effect in determining semantic interpretation; intonational information is represented at prosodic structure, a structure that is related to but separate from the c-structure. Information structure, described in Chapter 7, Section 3, also plays a central role in semantic interpretation. We will not examine constraints on meaning assembly defined at nonsyntactic levels of representation in the following, but it is worth keeping in mind that these other levels also constrain or contribute to semantic content.

2. SEMANTIC FORMS

As discussed in Chapter 2, Section 3.1, the value of the PRED feature in the f-structure is called a *semantic form*. This nomenclature reveals an early LFG view of meaning and its relation to f-structure: as discussed in Chapter 8, semantic forms were originally seen as the locus of semantic description. On the view presented by Kaplan and Bresnan (1982), semantic forms represent four types of information (see also Dalrymple et al. 1993):

(2) a. Specification of the semantic relation

b. Mapping of grammatical functions to semantic roles

c. Subcategorization information (the governed grammatical functions)

d. Instantiation to indicate distinctness (predicate uniqueness)

Chapter 8, Section 2 discussed this view of semantic forms, which assumes a semantic form like the one in (3) for the verb *give*:

(3) Semantic form for *give* (Kaplan and Bresnan 1982):

$$\text{`GIVE}\langle \begin{array}{c}\text{SUBJ}\\ \text{—}\\ \text{AGENT}\end{array} , \begin{array}{c}\text{OBJ}\\ \text{—}\\ \text{THEME}\end{array} , \begin{array}{c}\text{OBL}_{\text{GOAL}}\\ \text{—}\\ \text{GOAL}\end{array} \rangle\text{'}$$

This semantic form specifies that the predicate GIVE has three arguments with roles AGENT, THEME, and GOAL; that the AGENT is mapped to SUBJ, the THEME is mapped to OBJ, and the GOAL is mapped to OBL$_{\text{GOAL}}$; that the f-structure for a sentence with this verb must contain a SUBJ, an OBJ, and an OBL$_{\text{GOAL}}$ in order for the Completeness and Coherence conditions to be met; and that this use of the verb *give* is distinct from other uses of the same verb, since each use of a semantic form is uniquely indexed (Chapter 5, Section 2.2.1).

More elaborated theories of several of these aspects of semantic forms have emerged in the years since Kaplan and Bresnan's original work. Most obviously, the mapping of grammatical functions to semantic roles has been the focus of much theoretical attention and is discussed in detail in Chapter 8.

Further, the semantic form is no longer assumed to represent semantic relations in f-structure. Instead, the semantic contribution of a verb like *give* is reflected in the semantic structure and its relation to meanings, to be described in this chapter, as well as in argument structure (Chapter 8). This separation leads to a more modular theory, since on this view f-structure is a purely syntactic level of representation, not a mix of syntactic and semantic information. In addition, a more adequate view of meaning and its relation to syntax is thereby available: the original view of the semantic form was inadequate to represent anything but the most basic semantic relations. Semantic forms could not represent many aspects of interpretation, including scope of modifiers, quantification, and notions of coreference.

What, then, is the role of the semantic form value of the PRED feature in the current setting? First, the function of instantiation to indicate distinctness remains.

There are often cases in which a phrase makes a syntactically unique contribution, and the fact that semantic forms are instantiated uniquely for each instance of their use enforces this requirement.

Second, semantic forms represent the array of syntactic arguments that a predicate requires, making explicit the result of the application of mapping principles to the argument structure of a predicate. As discussed in Chapter 8, syntactic and semantic argument structure are not the same; verbs like intransitive *eat* and *rain* illustrate this point:

(4) a. *Chris ate.*

b. *It rained.*

Although *eat* denotes a two-place relation between an eater and an eaten thing, syntactically it has an intransitive use; conversely, *rain* does not take a semantic argument, but is syntactically monovalent. Semantic forms represent the syntactic grammatical functions required by a predicate, whether or not they make a semantic contribution.

3. SEMANTIC STRUCTURE AND MEANING COMPOSITION

Approaches to meaning and the syntax-semantics interface within the LFG framework share a striking degree of commonality: rules for semantic composition are formulated primarily by reference to syntactic predicate-argument structure, the syntactic organization of f-structure; and a theory of either implicit or explicit instructions for combining the meanings of the parts of a sentence into the meaning of the whole, what Fenstad et al. (1987) call a "logical syntax," is based on these f-structure relations.

In the first comprehensive treatment of semantics and its relation to syntax within LFG theory, Halvorsen (1983) proposes a semantic structure that is obtained by analysis of the f-structure, as described in Chapter 7, Section 4.2. The semantic structure that Halvorsen proposes consists of instructions on how to assemble meanings represented as formulas of the intensional logic of Montague (1974b); thus, the semantic structure represents an explicitly stated and clearly worked out theory of semantic composition, a set of instructions for meaning assembly.

Reyle (1988) provides a different view of semantic composition, one that is in some sense more closely tied to c-structure composition but that is interestingly different from the standard assumptions of the rule-to-rule hypothesis. On Reyle's approach, the basic meaning contributions of the daughters in a phrase structure rule are gathered up into a set of contributions associated with the mother node.

These contributions consist of expressions of intensional logic that are indexed by f-structure relations like SUBJ and OBJ. These contributions can combine in different orders, and these different orders can correspond to different meanings — for instance, to different scopes for quantifiers, similar in some respects to the treatment of quantifier scope ambiguity described in Dalrymple et al. (1997b) and Section 8 of this chapter: the order in which meanings are combined does not necessarily mirror the order of phrasal composition, and a freer order is allowed.

Wedekind and Kaplan (1993) and Kaplan and Wedekind (1993) present a theory of semantic interpretation that relies on the *restriction operator*, discussed in Chapter 6, Section 3.4. The restriction operator allows reference to the f-structure that results from removing an attribute and its value from another f-structure. Wedekind and Kaplan's analysis is primarily targeted at a treatment of the semantics of modification, which had proven problematic in various ways in previous approaches. An interesting and important aspect of Wedekind and Kaplan's proposal is that it incorporates a form of *resource accounting*: the semantic argument of a modifier is defined in terms of the meaning that results from removing the modifier from the structure, and the final meaning is obtained by applying the meaning of the modifier to this argument. This means that each modifier is required to make exactly one contribution to the final meaning. In the following, we will see why this property is a particularly desirable one.

These approaches illustrate three important and desirable properties of a theory of semantic composition. First, the theory should incorporate a systematic and explicit theory of how meanings combine, grounded in a thorough understanding of the space of theoretical possibilities, structures, and results. Second, it should not impose an explicit order of composition that is tied to constituent structure organization. Third, it should treat meanings as resources that are accounted for in the course of semantic composition. Section 5 of this chapter introduces an approach to semantic composition and the syntax-semantics interface, the *glue approach*, that meets these conditions.

Before introducing the theory, however, we must decide on a method for representing the meaning of an utterance and its parts; in the next section, we turn to the issue of meaning representation.

4. EXPRESSING MEANINGS

In formulating a theory of the relation between syntax and meaning, one of our first decisions is how to represent the meanings of words and phrases. In this book, we will concentrate primarily on issues related to semantic composition and the syntax-semantics interface. Many details of semantic interpretation do not interact significantly with principles of meaning assembly and semantic composition;

thus, our overall goal will be to use the simplest possible meaning representations that are adequate to represent the semantic distinctions we are interested in.

We will generally use standard predicate logic as a way of expressing meanings.[1] This formal system has several advantages: it is a simple and uncluttered representation, and it is widely known and generally familiar. Further, meanings represented as terms of predicate logic can often be readily translated into the representations used in other semantic theories, so that the use of predicate logic is not unduly limiting or confining. In fact, our predicate logic representations might profitably be viewed as abbreviations for the full semantic representations proposed in other semantic theories. Formally, the only requirement we impose on our system of meaning representation is that it must permit function abstraction and application, with a well-defined notion of variable binding, and predicate logic meets this desideratum.

It is of course possible to work within a different, more expressive theory of meaning representation, such as intensional logic (Montague 1974b), Discourse Representation Theory (Kamp 1981; Kamp and Reyle 1993), or Situation Semantics (Barwise and Perry 1983). Importantly, these semantic theories are fully compatible with the 'glue' approach to semantic composition that we present here. Dalrymple et al. (1999b) provide a short illustrative example of the use of Discourse Representation Structures in a glue setting to represent meanings. Other glue-based approaches to the syntax-semantics interface using intensional logic or Discourse Representation Theory are described in Section 4.2 of this chapter. Since the semantic issues treated in those works are not the main focus of our discussion in this book, we will be content with a simpler system.

The following few pages contain a brief introduction to some basic concepts of predicate logic. Gamut (1991a,b) and Partee et al. (1993, Chapter 7) give a much more complete explication of the concepts introduced below, as well as a full exposition of their formal underpinnings.

4.1. Predicate Logic

The expression in (5) represents the meaning of the proper noun *David*:

(5) *David*

David is a constant representing the individual David. Representing the meaning of a proper noun as an individual constant is a convenient simplification; to do complete justice to the meaning of proper names, a fairly complex theory of

[1] In our discussion of anaphoric binding in Chapter 11 and of noun phrase coordination in Chapter 13, some extensions to predicate logic will be necessary. In particular, to treat anaphoric binding we require a representation of individuals relevant in the current context and a notion of dynamic variable binding; to treat noun phrase coordination we need a basic theory of plurals and group formation. Elsewhere, predicate logic adequately represents the semantic distinctions we need to draw.

individual reference would be required. We stress that such a theory is fully compatible with the glue theory of semantics and meaning assembly that we present and that the constant *David* can be thought of as an abbreviated representation of the fully fleshed-out semantic contribution of the proper name *David.*

We use the expression in (6) to represent the meaning of the sentence *David yawned*:

(6) *yawn*(*David*)

Formally, the expression in (6) indicates that the one-place function *yawn* is applied to *David* — or, to say the same thing in a different way, the predicate *yawn* holds of *David*. This expression means that David yawned, but does not represent many details of the meaning of the sentence, including its tense. Again, when these details are not immediately relevant to our discussion, we will usually omit them.

4.1.1. Lambda Expressions

The expression *yawn* represents a function that takes an argument like *David*. For greater flexibility, we would like to have a general method for constructing functions from other expressions; this is made possible by the use of the *lambda operator* λ, which allows the construction of new functions by abstracting over variables in logical expressions:[2]

(7) Lambda abstraction:

$\lambda X.P$ represents a function from entities represented by X to entities represented by P.

Usually, the expression P contains at least one occurrence of the variable X, and we say that these occurrences are *bound* by the λ lambda operator.

To avoid situations where the same variable name has accidentally been chosen for two different variables, we might sometimes need to rename the variables that are bound by a lambda operator. The expressions in (8a) are equivalent, and so are the ones in (8b):

(8) a. $\lambda X.\mathit{person}(X) \equiv \lambda Y.\mathit{person}(Y)$

b. $\lambda X.\mathit{admire}(X,X) \equiv \lambda Y.\mathit{admire}(Y,Y)$

Besides the equivalences that come from variable renaming, there are many other equivalent ways of writing a function. We will generally try to represent a function in the clearest way possible, which will usually be the simplest and shortest way. For example, in the case of a one-place function like *person*, the shortest

[2] A more complete discussion of the lambda operator and the lambda calculus can be found in Gamut (1991b, Chapter 4) and Partee et al. (1993, Chapter 13).

and simplest way is just to write the name of the function *person*. Alternatively, we can apply the function *person* to an argument X that is bound by the λ lambda operator, and we have constructed a one-place function $\lambda X.\mathit{person}(X)$ that is the same as the function *person*. The following two expressions are equivalent:

(9) $\lambda X.\mathit{person}(X) \equiv \mathit{person}$

At times it will be clearer to write a function in this way; for example, writing the function as $\lambda X.\mathit{person}(X)$ shows explicitly that it is a function that takes one argument.

Another way of thinking of a function like $\lambda X.\mathit{person}(X)$ is that it picks out the set of individuals that are people — that is, the set of individuals X for whom the expression $\mathit{person}(X)$ is true. The function $\lambda X.\mathit{person}(X)$ is called the *characteristic function* of the set of people. We will sometimes refer to sets and their characteristic functions in our discussions of meaning.

4.1.2. Function Application

As in (6), we can apply a function to its argument:

(10) Function application:

$[\lambda X.P](a)$

The function $\lambda X.P$ is applied to the argument a.

Square brackets around the function expression have been added to make the groupings in this expression explicit. This expression is equivalent to the expression that results from replacing all occurrences of X in P with a. For example, the expression $[\lambda X.\mathit{yawn}(X)](\mathit{David})$ is equivalent to the expression $\mathit{yawn}(\mathit{David})$, which is the expression that results from replacing all occurrences of X in $\mathit{yawn}(X)$ with *David*:

(11) $[\lambda X.\mathit{yawn}(X)](\mathit{David}) \equiv \mathit{yawn}(\mathit{David})$

There is usually at least one occurrence of X in P. If there is more than one occurrence of X, as in example (8b), each occurrence is replaced by the argument of the function.

4.1.3. Types

We assume that the expressions we are working with are *typed*. As shown earlier, we propose the individual constant meaning *David* for the proper name *David*; this meaning has type e (for entity), the type of individuals:

(12) $\mathit{David} : e$

The expression in (12) indicates that the constant *David* is of type e. We assume that there are only two basic types: e is associated with individual-denoting expressions and t (for truth value) is associated with proposition-denoting expressions, which have a truth value (i.e., which are either true or false). The expression $\mathit{yawn}(\mathit{David})$ is of type t:

(13) $\mathit{yawn}(\mathit{David}) : t$

Types of other expressions are built up from these basic types. For example, the type of a one-place relation like *yawn* is:

(14) $\lambda X.\mathit{yawn}(X) : \langle e \rightarrow t \rangle$

The function $\lambda X.\mathit{yawn}(X)$ is of type $\langle e \rightarrow t \rangle$, a function from expressions of type e (represented by X) to expressions of type t (represented by $\mathit{yawn}(X)$). This function is true when applied to any individual that yawned and false otherwise.

The type of a two-place relation like *selected* is:

(15) $\lambda X.\lambda Y.\mathit{select}(X, Y) : \langle e \rightarrow \langle e \rightarrow t \rangle \rangle$

This is a function from expressions of type e (represented by X) to functions from expressions of type e (represented by Y) to expressions of type t (represented by $\mathit{select}(X, Y)$).

The types we have examined so far are:

(16)

expression	type
David	e
$\mathit{yawn}(\mathit{David})$	t
$\lambda X.\mathit{yawn}(X)$	$\langle e \rightarrow t \rangle$
$\lambda X.\lambda Y.\mathit{select}(Y, X)$	$\langle e \rightarrow \langle e \rightarrow t \rangle \rangle$

As we will see, the type of an argument can be important in constraining possibilities for meaning assembly.

4.1.4. QUANTIFICATION

Since the work of Montague (1974b) and Barwise and Cooper (1981), there has been a great deal of interest in the properties of quantifiers like *every* and *most*. Here we present a brief discussion of quantification; a more complete discussion can be found in Gamut (1991b, Chapter 7), Partee et al. (1993, Chapter 14), and Keenan and Westerståhl (1997). In Section 8 of this chapter, we will discuss how quantifiers are treated in the glue approach adopted in this work.

The noun phrase *everyone* is a quantifier. A sentence like *Everyone yawned* has a meaning that can be represented in the following way:[3]

(17) *Everyone yawned.*

$\mathit{every}(X, \mathit{person}(X), \mathit{yawn}(X))$

The quantifier *every* represents a relation between an individual (here X) and two propositions involving that individual, the proposition $\mathit{person}(X)$ and the proposition $\mathit{yawn}(X)$. The first proposition corresponds to what is often called the *restriction* of the quantifier *every*, and the second proposition corresponds to the *scope*. The type of a quantifier like *every* is:

(18) *every*: $\langle\langle e \multimap \langle t, t\rangle\rangle \multimap t\rangle$

This type associates an individual e with a pair of propositions $\langle t, t\rangle$ that involve that individual. Different quantifiers place different requirements on this relation. For example, for $\mathit{every}(X, \mathit{person}(X), \mathit{yawn}(X))$ to be true, any individual X that is a person — for whom $\mathit{person}(X)$ is true — must also yawn, satisfying $\mathit{yawn}(X)$. In other words, every individual that is a person must also be an individual that yawns.

(19) *Most people yawned.*

$\mathit{most}(X, \mathit{person}(X), \mathit{yawn}(X))$

The quantifier *most* requires that more than half of the individuals X satisfying the proposition $\mathit{person}(X)$ must also satisfy the proposition $\mathit{yawn}(X)$.

(20) *No person yawned.*

$\mathit{no}(X, \mathit{person}(X), \mathit{yawn}(X))$

The quantifier *no* requires that any individual X who satisfies the proposition $\mathit{person}(X)$ must *not* satisfy the proposition $\mathit{yawn}(X)$ — that is, there should be no individuals that are people that also yawn.

The restriction of a quantifier — its first propositional argument — is syntactically fixed, given by the meaning of the quantified common noun (*person* or *people* in the examples above) and any modifiers it might have. In contrast, the scope of a quantifier — its second propositional argument — is chosen more freely. As we will discuss in Chapter 12, Section 2.1, example (21) is syntactically unambiguous, with only one c-structure tree and one f-structure. It is semantically ambiguous, however, since the scope of the quantifier can vary:

[3] In our analysis of quantification we use*pair quantifiers*, expressions like the one in (17), instead of standard generalized quantifiers (*every(person,yawn)*). There is a one-to-one correspondence between the two types of quantifiers, as shown by Dalrymple et al. (1991).

(21) *Someone seemed to yawn.*

Reading 1: $\mathit{seem}(\mathit{a}(X, \mathit{person}(X), \mathit{yawn}(X)))$
Reading 2: $\mathit{a}(X, \mathit{person}(X), \mathit{seem}(\mathit{yawn}(X)))$

According to Reading 1, the proposition $\mathit{a}(X, \mathit{person}(X), \mathit{yawn}(X))$ seems to hold; that is, it seems to be the case that someone yawned, although in fact no one may actually have yawned. In contrast, Reading 2 claims that there is some individual X that satisfies the proposition $\mathit{person}(X)$ and that also seemed to yawn, satisfying the proposition $\mathit{seem}(\mathit{yawn}(X))$. Such examples show that it is not adequate to rely on the f-structure as a representation of the meaning of a sentence; the single f-structure for example (21) corresponds to more than one meaning. Our theory of semantics and the syntax-semantics interface allows us to deduce exactly these two meanings for example (21), given its unambiguous syntactic structure.

This concludes our brief introduction to predicate logic. We have seen that predicate logic provides a basic yet sufficiently expressive way of representing linguistic meaning. This is an advantage from our perspective, since much of our discussion will focus on issues in meaning assembly, and our claims about the meanings of particular constructions will be fairly general.

4.2. Other Semantic Theories

In much LFG work on meaning and semantic composition, specific assumptions about the nature and representation of linguistic meaning and its relation to syntactic structure have been explored, and a close analysis of the semantic contributions of particular phrases or constructions has been the main focus of concern. Work on integrating an LFG view of semantic composition with other semantic theories is important and valuable, since this work not only allows for a fuller exploration of the relation of syntactic structure to meaning, but also makes important contributions to semantic theory and meaning representation.

Since the work of Montague (1970), it has been common to use *intensional logic* to express linguistic meaning. Halvorsen (1983) proposed a theory of the association between f-structures and meanings, outlined briefly in Chapter 7, Section 4.2, which allowed the construction of formulas of intensional logic to represent the meanings of utterances based on their f-structures. Meanings have also been represented as formulas of intensional logic in an LFG setting by Wedekind and Kaplan (1993) and by Dalrymple et al. (1997b).

In other work, the semantic theory of Situation Semantics (Barwise and Perry 1983) is assumed. Fenstad et al. (1987) propose that functional descriptions in rules and lexical entries describe not only the f-structure for an utterance but also a *Situation Schema*, which represents information provided by linguistic form that is relevant for semantic interpretation. Situation Semantics adheres to the *Relational Theory of Meaning*, whereby the meaning of an utterance is a relation between the situation in which an utterance is made — the *utterance situation* — and the situation described by the utterance, the *described situation*. Accordingly, the situation schemata proposed by Fenstad et al. (1987) represent a potentially underdetermined description of the relation between an utterance situation and a described situation. Fenstad et al. provide an extensive treatment of constraints on situation schemata as well as an algorithm for their interpretation. Gawron and Peters (1990) also propose a Situation-Theoretic view of anaphora, quantification, and their interactions from an LFG perspective, and their work includes an appendix containing an LFG grammar for the fragment of English that they treat.

Perhaps the most widely adopted theory of semantics among LFG researchers is Discourse Representation Theory (Kamp 1981; Kamp and Reyle 1993). Discourse Representation Theory assumes that each sentence in a discourse contributes to the construction of a *Discourse Representation Structure* representing the discourse referents that are introduced as well as the conditions they must meet. Frey and Reyle (1983) advanced one of the first proposals for constructing Discourse Representation Structures for utterances based on their f-structures, and this work was continued by Wada and Asher (1986) and Asher and Wada (1988) in their proposals for LFG-based DRS construction. Muskens (1995) also proposes an analysis involving Underspecified Discourse Representation Structures (Reyle 1993) with syntactic assumptions that are very close to LFG.

Within the glue approach to semantic composition that we are about to explore, there is no obstacle to representing linguistic meanings according to these semantic theories. Dalrymple et al. (1997b) discuss quantification and intensionality in the glue approach, using intensional logic to represent meanings. As mentioned earlier, Dalrymple et al. (1999b) briefly discuss the construction of Discourse Representation Structures in a glue setting, where meanings are given as expressions of Lambda DRT (Bos et al. 1994). Van Genabith and Crouch (1999a) provide a detailed and very interesting discussion of different methods for incorporating dynamic and underspecified meaning representations, similar to the structures of Underspecified Discourse Representation Theory, within the glue approach (see also van Genabith and Crouch 1999b).

5. MEANING ASSEMBLY AND LOGICAL 'GLUE'

This section introduces the glue theory of semantic composition and presents some basic examples of meaning assembly in the glue setting. We propose a logically based theory of semantic composition: instructions for combining meanings are stated as premises in a logical deduction. The deduction of the meaning of an utterance proceeds by combining these premises as the logic requires, which means that meaning composition need not proceed according to the rules of phrasal composition. And the logic used to state constraints on meaning combination is a resource logic, *linear logic*, which treats meaning contributions as resources that are accounted for in the meaning deduction. Thus, the theory conforms to the desiderata introduced at the end of Section 3 of this chapter. The theory is often referred to as the *glue approach* because of the role of linear logic in stating how the meanings of the parts of an utterance can be "glued together" to form the meaning of the whole utterance.

5.1. Meaning Specifications and the Projection Architecture

The lexical entry for a proper name like *David* contains at least the syntactic information shown in (22):

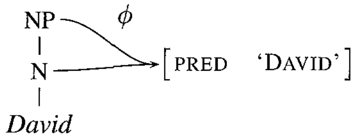
(22) *David* N ($\uparrow$ PRED) = 'DAVID'

We also adopt the following simplified phrase structure rule for NP:

(23) NP $\longrightarrow$ $\underset{\uparrow=\downarrow}{\text{N}}$

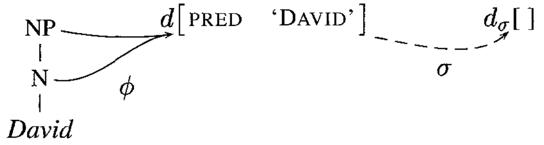
As discussed in Chapter 5, this lexical entry and phrase structure rule give rise to the syntactic structures in (24):

(24)



We now augment our theory with a *semantic structure* and its associated meaning. As described in Chapter 7, a linguistic structure like the semantic structure is related to other linguistic structures by means of a *correspondence function*. Here, the function σ relates f-structures to semantic structures, and we say that the semantic structure is a *projection* of the functional structure. In (25), d_σ is the semantic structure that is related to the f-structure labeled d by the correspondence function σ, represented as a dotted line:

(25)



As noted in Chapter 7, Section 2, there are two common and equivalent notations for the correspondence function:

(26) $d_\sigma \equiv \sigma(d)$

In the following, we will use the subscript notation that is used most commonly in recent LFG literature, rather than the parenthesis notation: that is, we will write d_σ rather than $\sigma(d)$ for the semantic structure corresponding to d via the correspondence function σ. Nothing of substance depends on this notational choice; using the parenthesis notation would be equally correct.

We propose the augmented lexical entry in (27) for the proper name *David*. This lexical entry differs from the one in (22) in that the expression $\mathit{David} : \uparrow_\sigma$ has been added. No additions or changes to the phrase structure rule in (23) are necessary:

(27) *David* N ($\uparrow$ PRED) = ‘DAVID’
$\mathit{David} : \uparrow_\sigma$

The expression $\mathit{David} : \uparrow_\sigma$ is called a *meaning constructor*, since it is an expression that tells us how to construct meanings.[4] In this simple case, there is no real meaning construction involved, since the meaning *David* is complete on its own. Other cases are more complex, as we will soon see.

Meaning constructors are pairs, with the left-hand side (the *meaning side*) representing a meaning and the right-hand side (the *glue side*) representing a logical formula over semantic structures corresponding to that meaning. The expression $\mathit{David} : \uparrow_\sigma$ says that *David* is the meaning associated with $\uparrow_\sigma$, the semantic projection of the f-structure $\uparrow$. In (25), the f-structure metavariable $\uparrow$ is instantiated to the f-structure labeled d, and so the meaning constructor pairs the meaning *David* with the semantic structure d_σ.

As discussed in Section 4.1.3 of this chapter, meaning expressions are typed; the constant *David* is of type e. We assume that the basic types e and t are associated with semantic structures, since the type of an expression is important in determining how it can combine with other expressions. Types are written on the semantic structure as subscripts enclosed in angled brackets:

(28) $\mathit{David} : d_{\sigma\langle e\rangle}$

[4] The meaning constructors we assume in this work are cast in the so-called “Curry-Howard” or “new glue” format, conforming to the proposals made by Dalrymple et al. (1999a). This format departs from much earlier work in the glue framework, including most of the papers collected in Dalrymple (1999), in which the meaning constructor for*David* would have been written as $\uparrow_\sigma \leadsto \mathit{David}$ (read as ‘$\uparrow_\sigma$ means *David*’). The two formats have different expressive power, and in fact the “new glue” format adopted here is the more constrained of the two; see Dalrymple et al. (1999a) for more discussion of the two formats and the formal differences between them.

When the type of a semantic structure is clear from the context, we will often omit it to reduce notational clutter.

For brevity, we can use the label **[David]** to refer to this meaning constructor. In (29), the label **[David]** refers to the typed meaning constructor $\mathit{David} : d_{\sigma\langle e \rangle}$, in which d_σ is a semantic structure of type e and *David* is an individual constant representing the individual named David :

(29) **[David]** $\mathit{David} : d_{\sigma\langle e \rangle}$

Using names or labels for meaning constructors proves to be useful for presenting deductions in a more compact form.

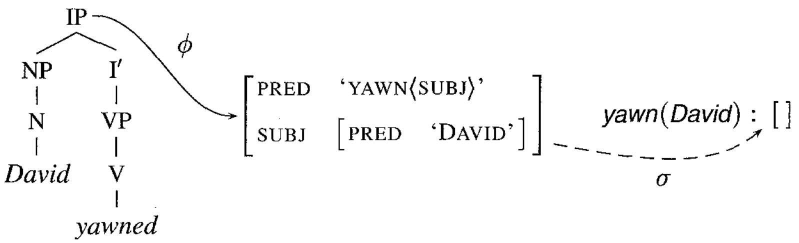
5.2. Assembling Meanings

Some words, like verbs, must combine with other meanings to produce a complete meaning. For example, an intransitive verb combines with its subject to produce a meaning for the sentence. This means that we must provide instructions for combining the meaning of a verb with its arguments to form the meaning of the sentence as a whole. We provide these instructions in logical terms, the "glue language" of linear logic.

5.2.1. Example One: Intransitive Verbs

The syntactic structures for the sentence *David yawned*, together with the semantic result we desire, are displayed in (30):

(30) *David yawned.*



The semantic structure for the sentence is related to its f-structure by the correspondence function σ, represented as a dotted line. We are not concerned with the internal structure of the semantic structure here, and so we have represented the semantic structure with no internal attributes or values, as the structure [ ]. Below, we will see cases in which the semantic structure has attributes whose values play a crucial role in meaning deduction.

Let us see how the meaning *yawn*(*David*) for the sentence *David yawned* is obtained. We propose the following simplified lexical entry for the verb *yawned*:

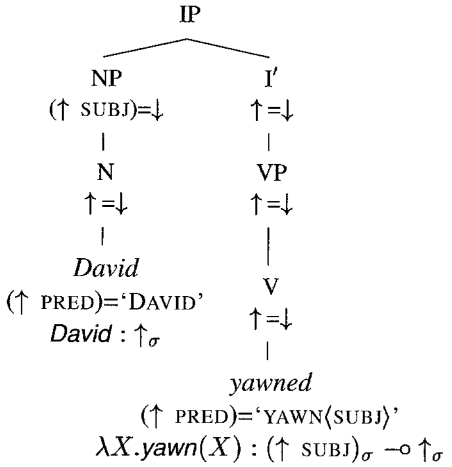
(31) *yawned* V $(\uparrow \text{PRED}) = \text{`YAWN}\langle\text{SUBJ}\rangle\text{'}$
$\lambda X.\mathit{yawn}(X) : (\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma$

The meaning constructor for *yawned* pairs the meaning for *yawned*, the one-place predicate $\lambda X.\mathit{yawn}(X)$, with the linear logic formula $(\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma$. This formula contains a new expression: the connective $\multimap$ is the *linear implication* symbol of linear logic, which we will discuss in more detail in Section 7 of this chapter. For the moment, we can think of the symbol as expressing a meaning like *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.

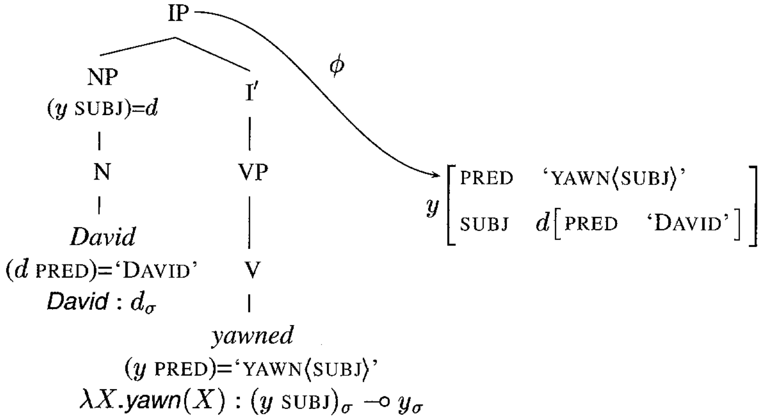
Additionally, the linear implication operator $\multimap$ carries with it a requirement for *consumption* and *production* of semantic resources: the formula $(\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma$ indicates that if a semantic resource $(\uparrow \text{SUBJ})_\sigma$ is found, it is *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. Thus, in the linear logic formulas that comprise the glue (right-hand) sides of meaning constructors, semantic structures are treated as resources that are contributed by the words and structures of the sentence.

In example (32) (page 234), we display the syntactic structures described by the lexical entries in (27) and (31), together with the meaning constructors contributed by the words *David* and *yawned*. We assume the standard phrase structure rules for English as outlined in Chapter 5, Section 4.1. In (33), we instantiate the metavariables represented by $\uparrow$ and $\downarrow$ in this tree, using the label y for the f-structure of the entire sentence and *d* for the SUBJ f-structure. Only the instantiated c-structure annotations that are important for our current discussion are displayed in (33).

(32) *David yawned.*



(33) *David yawned.*



The f-structure for *yawn* in (33) is labeled y, and the f-structure d for *David* is y's SUBJ. Since $(y \text{ SUBJ}) = d$, we can replace the expression $(y \text{ SUBJ})_\sigma$ by d_σ in the meaning constructors in (33), yielding the instantiated meaning constructors for *David* (labeled **[David]**) and *yawned* (labeled **[yawn]**) in (34):

(34) Meaning constructors for *David yawned*:

[David]	$\mathit{David} : d_\sigma$
[yawn]	$\lambda X.\mathit{yawn}(X) : d_\sigma \multimap y_\sigma$

The meaning (left-hand) sides of the meaning constructors in (34) are familiar from our discussion of predicate logic formulas in Section 4.1 of this chapter. The meaning side of the meaning constructor labeled **[David]** is the proper noun meaning *David*, and the meaning side of the meaning constructor labeled **[yawn]** is the meaning of the intransitive verb *yawned*, the one-place predicate $\lambda X.\mathit{yawn}(X)$.

The glue (right-hand) sides of these meaning constructors indicate how these meanings are associated with the different parts of this sentence. The constant *David* is associated with the semantic structure d_σ. The glue side of the meaning constructor labeled **[yawn]** is more complex: as explained earlier, the connective $\multimap$ is the linear implication symbol of linear logic, which we can think of as expressing a meaning like *if* d_σ, *then* y_σ. In other words, the glue side of the meaning constructor labeled **[yawn]** in (34) states that if we can find a resource associated with the semantic structure d_σ, then we can produce a resource associated with the semantic structure y_σ.

We must also provide rules for how the glue side of each of the meaning constructors in (34) relates to the meaning side in a meaning deduction. For simple, nonimplicational meaning constructors like **[David]** in (34), the meaning on the left-hand side is the meaning of the semantic structure on the right-hand side. For implicational meaning constructors like **[yawn]**, which contain the linear implication operator $\multimap$, performing a deductive step on the glue side corresponds to applying a function to its argument on the meaning side:[5]

(35)
$$\frac{\begin{array}{r}X : f_\sigma\\ P : f_\sigma \multimap g_\sigma\end{array}}{P(X) : g_\sigma}$$

Each side of an implicational meaning constructor $P : f_\sigma \multimap g_\sigma$ requires a contribution: the glue side requires as its argument a semantic structure f_σ, and the meaning side requires an argument for the predicate P. When an appropriate resource such as $X : f_\sigma$ is available to provide the appropriate contributions on both the meaning and the glue sides, the result is a complete semantic resource on the glue side and its corresponding meaning on the meaning side. In the case at hand, the pairing of the linear logic formula $d_\sigma \multimap y_\sigma$ with the meaning term $\lambda X.\mathit{yawn}(X)$ means that we apply the function $\lambda X.\mathit{yawn}(X)$ to the meaning *David* associated with d_σ, obtaining the meaning constructor $\mathit{yawn}(\mathit{David}) : y_\sigma$ for the sentence.

Besides this rule for function application, we also require a rule of abstraction that allows us to create functions. The rule in (36) allows us to temporarily posit an additional premise in the deduction, a semantic resource f_σ associated with

[5] This is the standard correspondence as defined by the*Curry-Howard Isomorphism* relating propositions like $d_\sigma \multimap y_\sigma$ to terms like $\lambda X.\mathit{yawn}(X)$; see Crouch and van Genabith (2000) for more discussion.

the meaning X. A semantic resource hypothesized in this way is notationally distinguished from other premises in that it is enclosed in square brackets: $[f_\sigma]$. If we can successfully perform a deduction (represented by elliptical dots $\vdots$) from this and other meaning constructor premises, producing a semantic resource g_σ with meaning P as in (36), we *discharge* the assumption $X : [f_\sigma]$, and we are left with the meaning constructor $\lambda X.P(X) : f_\sigma \multimap g_\sigma$.

(36)
$$\frac{\begin{array}{c} X : [f_\sigma] \\ \vdots \\ P(X) : g_\sigma \end{array}}{\lambda X.P(X) : f_\sigma \multimap g_\sigma}$$

Intuitively, we have shown that if we are given a resource f_σ, we can then obtain g_σ, exactly the import of the linear logic expression $f_\sigma \multimap g_\sigma$. On the meaning side, we have shown that by providing X, we can produce the meaning $P(X)$ — in other words, that we have proven the existence of a function $\lambda X.P(X)$. We will not use this abstraction rule in the immediately following examples, but it will be helpful in future discussion, especially in our discussion of raising verbs in Chapter 12 and of noun phrase coordination in Chapter 13. The appendix (page 433) contains the full set of rules of deduction for our fragment of linear logic.

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

(37) $\mathit{David} : d_\sigma$

The meaning *David* is associated with the SUBJ semantic structure d_σ.

$\lambda X.\mathit{yawn}(X) : d_\sigma \multimap y_\sigma$

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 $\lambda X.\mathit{yawn}(X)$ to the meaning associated with d_σ.

$\mathit{yawn}(\mathit{David}) : y_\sigma$

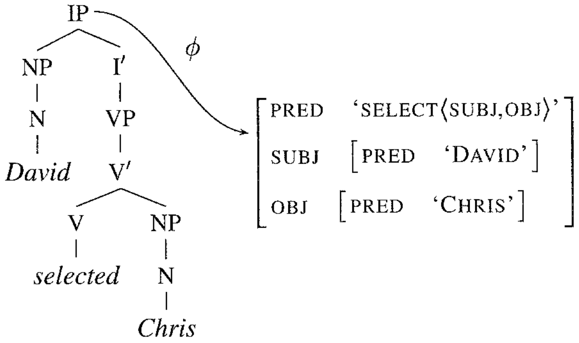
We have produced a semantic structure for the full sentence y_σ, associated with the meaning *yawn*(*David*).

By using the function application rule and the meaning constructors for *David* and *yawned*, we have deduced the meaning *yawn*(*David*) for the sentence *David yawned*, as desired.

5.2.2. Example Two: Transitive Verbs

Our next example of meaning deduction involves a transitive verb; the example differs from the one just presented only in that the verb takes two arguments instead of one. The c-structure and f-structure for the sentence *David selected Chris* are displayed in (38):

(38) *David selected Chris.*



The lexical entry for the transitive verb *selected* is shown in (39):

(39) *selected* V $(\uparrow \text{PRED}) = \text{`SELECT}\langle\text{SUBJ,OBJ}\rangle\text{'}$
$\lambda X.\lambda Y.\mathit{select}(X,Y) : (\uparrow \text{SUBJ})_\sigma \multimap [(\uparrow \text{OBJ})_\sigma \multimap \uparrow_\sigma]$

In the meaning constructor for the transitive verb *selected*, two arguments are required: a resource for the SUBJ, $(\uparrow \text{SUBJ})_\sigma$, and a resource for the OBJ, $(\uparrow \text{OBJ})_\sigma$. The square brackets in this expression are just added to make the groupings in the expression clear: *selected* requires a meaning for its SUBJ, then a meaning for its OBJ, to form a meaning for the sentence.[6] In other words, this formula can be paraphrased as: "If we find a resource for the subject and a resource for the object, we can produce a resource for the entire sentence." The meaning side represents a

[6] The glue side of the meaning constructor in (39) requires the verb to combine with its arguments in a particular order — the SUBJ first, then the OBJ — since this order must respect the order of combination of meanings specified in the lambda expression on the meaning side. The meaning constructor shown in (a) is exactly equivalent to the one in (39) except that the order of argument combination on both the meaning and glue sides is reversed, so that the verb combines with itsOBJ first and then its SUBJ:

(a) $\lambda Y.\lambda X.\mathit{select}(X,Y) : (\uparrow \text{OBJ})_\sigma \multimap [(\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma]$

In formal terms, the glue side of this meaning constructor is*logically equivalent* to the glue side of the meaning constructor in (39). In principle, we can choose any order of combination of premises, with no theoretical significance attached to the choice we make. For simplicity, in our discussion here and in the following chapters we will usually choose one particular order in which to combine premises.

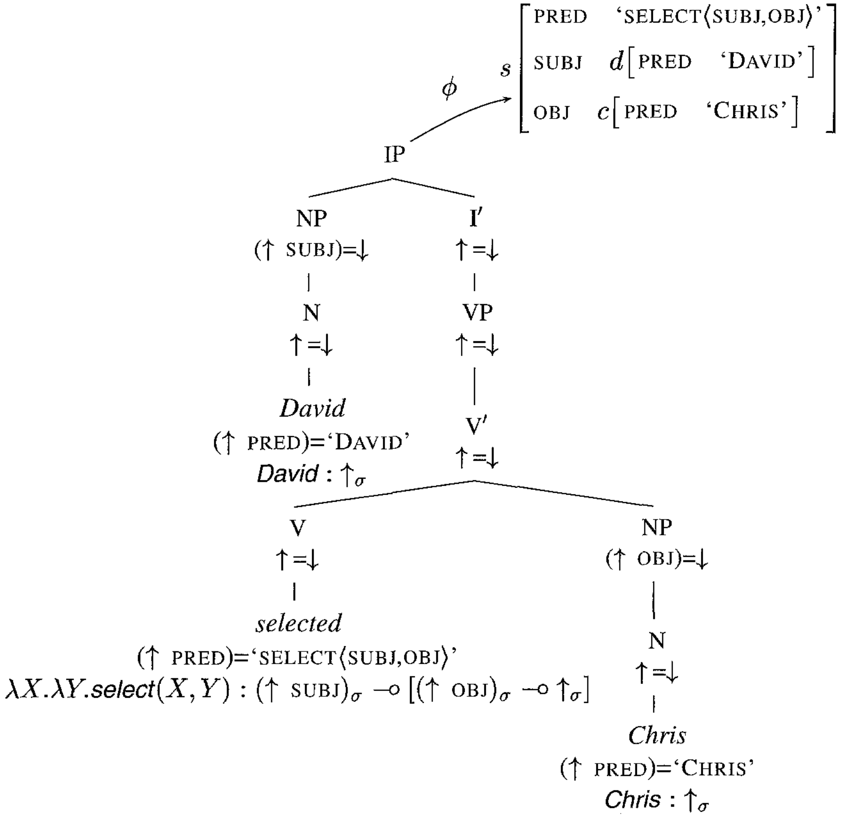
function that requires two arguments and is applied to those arguments to produce a meaning for the sentence.

The lexical entry for *Chris* is analogous to the one for *David*, providing a semantic structure as a resource associated with the meaning *Chris*:

(40) *Chris* N $(\uparrow \text{ PRED}) = \text{`CHRIS'}$
$\qquad\qquad\qquad\textit{Chris} : \uparrow_{\sigma}$

With these lexical entries for the words in the sentence, we have the following structures:

(41) *David selected Chris.*



Instantiating the ↑ metavariables in the meaning constructors for *David*, *Chris*, and *select*, we have the following meaning constructors:

(42) Meaning constructor premises for *David selected Chris*:

[David] $\mathit{David} : d_\sigma$

[Chris] $\mathit{Chris} : c_\sigma$

[select] $\lambda X.\lambda Y.\mathit{select}(X,Y) : d_\sigma \multimap [c_\sigma \multimap s_\sigma]$

From these premises, we can make the following logical deduction:

(43) $\mathit{David} : d_\sigma$ The subject semantic structure d_σ is associated with the meaning *David.*

$\mathit{Chris} : c_\sigma$ The object semantic structure c_σ is associated with the meaning *Chris.*

$\lambda X.\lambda Y.\mathit{select}(X,Y) : d_\sigma \multimap [c_\sigma \multimap s_\sigma]$

On the glue side, if semantic resources for the subject d_σ and the object c_σ are found, a resource for the sentence can be produced. On the meaning side, the two-place predicate *select* is applied to the subject meaning X and then the object meaning Y to produce the meaning $\mathit{select}(X,Y)$ for the sentence.

$\mathit{select}(\mathit{David}, \mathit{Chris}) : s_\sigma$ We have produced a semantic structure s_σ for the full sentence, associated with the meaning *select(David,Chris)*.

As desired, we have concluded that the meaning for the sentence *David selected Chris* is *select*(*David*, *Chris*).

In Section 5.1 of this chapter, we noted that meaning constructors can be assigned labels. We will sometimes take advantage of this possibility to present an abbreviated representation of a derivation from a set of premises. For example, we can abbreviate the derivation outlined in (43) in the following way:

(44) **[David]** $\mathit{David} : d_\sigma$

[Chris] $\mathit{Chris} : c_\sigma$

[select] $\lambda X.\lambda Y.\mathit{select}(X,Y) : d_\sigma \multimap [c_\sigma \multimap s_\sigma]$

[David], **[Chris]**, **[select]** $\vdash \mathit{select}(\mathit{David}, \mathit{Chris}) : s_\sigma$

The final line in (44) represents the derivation of the meaning *select*(*David*, *Chris*) for the semantic structure s_σ from the premises labeled **[David]**, **[Chris]**, and **[select]**. It contains a new expression $\vdash$, sometimes called the *turnstile*, which indicates that the conclusion on the right is derivable from the premises on the left. Thus, the final line in (44) means that the conclusion $\mathit{select}(\mathit{David}, \mathit{Chris}) : s_\sigma$ is derivable from the premises labeled **[David]**, **[Chris]**, and **[select]**.

In sum, we have used linear logic as a *glue language* to provide instructions on how to glue together or assemble meanings, based on the relations between the syntactic structures they correspond to. The use of this logical language lets us express constraints on meaning combinations in a formally coherent and flexible way, taking advantage of the syntactic relations imposed by the f-structure.

6. CONSTRUCTIONAL MEANING

In the examples just presented, meaning terms are associated with words and not phrase structure rules. In a language like English, annotations on phrase structure rules serve mainly to determine the functional syntactic role of a constituent. For the most part, phrase structure rules play only this syntactic organizing function and do not contribute meaning on their own. This is true for many other languages as well.

However, this generalization is not exceptionless. There are cases in which meaning is associated with a phrasal construction as a whole, where the semantic properties of the construction go beyond the semantic properties of the words it contains. A particularly clear example of meaning associated with phrasal configuration is provided by relative clauses with no relative pronoun, such as:

(45) *the man I met*

In this example, the phrase *I met* is a relative clause modifier of *man*. This information is not lexically associated with either the word *I* or the word *met*. Instead, the interpretation of *I met* as a relative clause is due to the phrasal configuration in which it appears. In Chapter 14, we will propose an analysis of the semantics of relative clauses, and we will see that the phrase structure rule associated with relative clause formation in English can in fact make a contribution to meaning.

The view that meanings can be attached either to lexical items or to c-structure configurations accords with the views of Kaplan and Bresnan (1982, fn. 2), but not with some other proposals. In particular, Halvorsen (1983) proposes that semantic content is introduced only in the lexicon, not by phrase structure rules (see also Bresnan 1982a). In a very interesting discussion of verbless sentences, including the topic-comment construction in Vietnamese and nominal sentences with no copula in Maori, Rosén (1996) shows that attempts to restrict semantic content to appearing only in the lexicon are inadvisable. Phrase structure configurations can be associated with meaning constructors, and these constructors can make an essential contribution to meaning deduction.

7. THE 'GLUE' LANGUAGE: LINEAR LOGIC

We use expressions of *linear logic* (Girard 1987) to give instructions on how to assemble meanings. Here, we informally describe only the properties of the small fragment of linear logic (the *multiplicative fragment*) that we will use.[7]

Intuitively, linear logic is different from classical logic in that premises in a linear logic deduction are treated as resources that must be kept track of, while this is not true in classical logic. Premises in a deduction in classical logic are statements about what is or is not true. In contrast, premises in a linear logic deduction are commodities, occurrences of resources that can be introduced or consumed.

To illustrate this difference, let us assume that we can deduce the statement *You will get wet* from the premises *If it is raining outside, you will get wet* and *It is raining outside* in classical logic:

(46) Classical logic:
If it is raining outside, you will get wet.
It is raining outside.
You will get wet.

In classical logic, if a conclusion can be deduced from a set of premises, the same conclusion can still be deduced if additional premises are added:

(47) Classical logic:
If it is raining outside, you will get wet.
It often rains in March.
It was raining yesterday.
It is raining outside.
You will get wet.

In contrast, linear logic does not allow the same conclusion to be deduced when additional premises are introduced. Instead, propositions in linear logic can be thought of as resources, and an economic metaphor is sometimes used.

For instance, we can use the symbol $\$1$ for the proposition that you have a dollar, $\$1 \multimap \mathit{apple}$ for the linear logic proposition that if you have $\$1$, you can get an apple, and apple for the proposition that you have an apple. The following is valid in linear logic:

(48) $[\$1 \multimap \mathit{apple}],\ \$1 \vdash \mathit{apple}$

This can be read as:

[7] In this section, we describe only the properties of the linear implication operator $\multimap$. In Chapter 11 we introduce the *multiplicative conjunction* operator $\otimes$, and in Chapter 13 we introduce the *of course* operator !. Proof rules for our fragment of linear logic are given in the appendix (page 433).

(49) *If you have $1, you can get an apple.*
You have $1.
$\overline{}$
You can get an apple.

Just as in the real world, it is not possible to get two apples with $1, or to still have $1 as well as the apple:

(50) INCORRECT (obtaining two apples with $1):
$[\$1 \multimap \mathit{apple}], \$1 \vdash \mathit{apple},\ \mathit{apple}$

INCORRECT (obtaining an apple while keeping $1):
$\$1, [\$1 \multimap \mathit{apple}] \vdash \$1, \mathit{apple}$

More schematically, inferences in linear logic work in the following way:

(51) INCORRECT: $A \vdash A, A$
We cannot deduce A, A from A.
A resource cannot be duplicated.

INCORRECT: $A, B \vdash A$
We cannot deduce A from A, B.
A resource cannot be discarded.

INCORRECT: $A, [A \multimap B] \vdash A, B$
The resource A is consumed by $A \multimap B$ to conclude B.
A resource is consumed by an implication.

INCORRECT: $A, [A \multimap B] \vdash A \multimap B, B$
Both A and $A \multimap B$ are consumed in concluding B.
A linear implication is also a resource and is consumed in the deduction.

CORRECT: $A, [A \multimap B] \vdash B$

This resource-sensitivity of linear logic allows us to model the meaning contributions of words as *semantic resources* that must be accounted for. The meaning of a sentence is deduced from the meanings of its component parts; it would be incorrect to deduce the same meaning for the sentence if words or phrases are added or subtracted. Each word or phrase makes a unique contribution that must be reflected in the final meaning of the sentence, and meanings cannot be arbitrarily duplicated, added, or discarded.

7.1. Semantic Completeness and Coherence

Formally, we say that a meaning derivation for an utterance is *semantically complete* if a meaning derivation from the premises contributed by the meaning-bearing items in the sentence produces a meaning for the semantic structure for the utterance that does not contain any unsaturated expressions (that is, in which all of the meaning contribution requirements are satisfied). If no such meaning can be produced, some required material is missing and the utterance is *semantically incomplete*.

We say that a meaning derivation for an utterance is *semantically coherent* if the meaning derivation produces a meaning for the utterance with no additional unused premises remaining. If extra resources besides the semantic resource for the utterance remain, the utterance is *semantically incoherent*.

Semantic completeness and coherence are related in a clear way to the syntactic Completeness and Coherence conditions on f-structures discussed in Chapter 2, Section 3.6. This is as expected, since most syntactic arguments also make a semantic contribution and thus must be accounted for in a meaning derivation; indeed, our logically defined semantic completeness and coherence conditions subsume syntactic Completeness and Coherence in all cases except for pleonastic or semantically empty arguments, which make no semantic contribution and are not accounted for in a semantic derivation. The following sentence is syntactically and semantically incomplete:

(52) **Yawned.*

The sentence is syntactically incomplete because the verb *yawned* requires a SUBJ, and no subject is present; the sentence is semantically incomplete because the meaning constructor for *yawned* requires a semantic resource corresponding to its subject, but none can be found. Example (53) is both syntactically and semantically incoherent:

(53) **David yawned Chris.*

This example is syntactically incoherent due to the presence of an OBJ argument, which *yawned* does not require. It is semantically incoherent because the meaning constructor for *yawned* requires only a SUBJ resource, and in a meaning deduction for these premises the semantic resource for *Chris* remains unused.

Semantic and syntactic completeness differ for arguments that make no semantic contribution:

(54) **Rained.*

The verb *rained* requires a SUBJ, but there is no SUBJ in example (54); therefore, the sentence is syntactically incomplete. The semantic completeness condition is not violated, however, because the SUBJ of *rained* is not required to make a semantic contribution.

Another difference between syntactic and semantic coherence involves modifying adjuncts: a semantic deduction in which the meaning contribution of a modifier is not incorporated is semantically incoherent, since all meanings must be taken into account. That is, the semantic coherence condition prevents us from assigning an unmodified meaning to a sentence with a modifier:

(55) *David ran quickly.*

cannot mean: *run*(*David*)

The modifier *quickly* is not constrained by the syntactic Completeness and Coherence conditions, which apply only to governable grammatical functions. Semantically, however, its meaning contribution must be taken into account, and the deduction is semantically incoherent if the modifier meaning does not appear.

7.2. Glue Deductions and Meaning

Glue semantic deductions have an interesting property: as shown by Dalrymple et al. (1999a), whether or not a glue deduction is possible depends only on the linear logic glue formulas on the right-hand side of the meaning constructor, never on the meanings involved in the deduction. This means that we can think of the meaning deduction process purely in terms of the linear logic deduction over semantic structures; on the basis of the resulting deduction, we can determine the meaning of the resulting constituents by function abstraction and application.

For example, we can present deductions in an abbreviated form like (56), which is the same as the deduction in (37) of this chapter except that meaning terms have been omitted. On the basis of this deduction, we can determine the meaning corresponding to the semantic structure y_σ by function application, following the function application rule presented in (35) of this chapter.

(56) d_σ The SUBJ semantic structure d_σ is present.

$d_\sigma \multimap y_\sigma$ If we find a resource for the SUBJ semantic structure d_σ, we can produce a resource for the semantic structure for the full sentence y_σ.

---

y_σ We have produced a semantic structure for the full sentence y_σ.

In fact, as discussed by Dalrymple et al. (1999a), this aspect of glue semantic deductions is strongly similar to Categorial Grammar (Oehrle et al. 1988; Moortgat 1988, 1996; Morrill 1994; Steedman 1996). Linguistic analysis in Categorial Grammar is a deductive process, in which the syntactic structure and the meaning of a sentence are obtained by a logical deduction from premises contributed by its words. The Lambek calculus (Lambek 1958), the logical system commonly used in syntactic analysis in categorial frameworks, is actually a fragment of *noncommutative multiplicative linear logic* and so is very close to the linear logic glue language.

Probably the most important difference between the categorial approach and the glue approach is in the syntactic primitives that are relevant for semantic composition. In categorial grammar, a predicate combines with its arguments on the basis of relations defined on the surface string, like to-the-left-of and to-the-right-of; in the glue approach, in contrast, semantic deductions are guided by f-structural relations like SUBJ, COMP, and ADJ. This frees the glue approach from concerns with crosslinguistically variable constituent structure relations and allows semantic composition to proceed according to the more abstract syntactic organization of f-structure.

8. QUANTIFICATION

Here we will briefly outline our theory of quantification and the treatment of generalized quantifiers, since an explicit theory of the syntax and semantics of noun phrases will be important in subsequent discussion, particularly in our discussion of adjectival modification and relative clauses. For a full explication of the theory of quantification presented in this section, see Dalrymple et al. (1997b).

8.1. Quantifier Scope

As discussed in Section 4.1.4 of this chapter, the meaning of a sentence like *Everyone yawned* is:

(57) *Everyone yawned.*

$\mathit{every}(X, \mathit{person}(X), \mathit{yawn}(X))$

Here, *every* relates an arbitrary individual represented by X to two propositions about that individual, $\mathit{person}(X)$ and $\mathit{yawn}(X)$. We propose the lexical entry in (58) for the quantifier *everyone*:

(58) *everyone* N (↑ PRED) = 'EVERYONE'

$\lambda S.\mathit{every}(X, \mathit{person}(X), S(X)) : \forall H.[\uparrow_\sigma \multimap H] \multimap H$

This entry has a number of new features, which we will explain in the following sections.

8.1.1. Quantifier Scope and Meaning Assembly

The glue side of the meaning constructor in the second line of the lexical entry in (58) has several new aspects, different from the meaning constructors for proper names and verbs that we have examined thus far:

(59) $\forall H.[\uparrow_\sigma \multimap H] \multimap H$

First, a universal quantifier $\forall$ binds the variable H, which ranges over semantic structures that correspond to possible scopes of the quantifier. The universal quantifier $\forall$ means something close to the English word *all* or *every*, and it binds the variable that follows it; see Partee et al. (1993, Chapter 7) for a full explanation. In (59), the expression $[\uparrow_\sigma \multimap H] \multimap H$ is asserted to be true for any H: if we find a resource for any H that satisfies the implication $\uparrow_\sigma \multimap H$, we can obtain the resource H.

The second new aspect of the meaning constructor in (58) is that it contains an *embedded implication*: the implication $\uparrow_\sigma \multimap H$ appears on the left side of the main linear implication operator. We can think of the expression $\uparrow_\sigma \multimap H$ as the *argument* required by the meaning constructor for *everyone*. As we have seen, the arguments required by a meaning constructor appear on the left side of the main implication operator. An intransitive verb like *yawned* requires as its argument the meaning of its subject, $(\uparrow \textsc{subj})_\sigma$:

(60) $\lambda Y.\mathit{yawn}(Y) : (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma$

In contrast, the quantifier *every* takes a more complex argument, an implicational meaning constructor $\uparrow_\sigma \multimap H$, in the lexical entry in (58). That is, *every* requires as its argument a meaning constructor that consumes a resource for $\uparrow_\sigma$ to produce some semantic structure H. An intransitive verb with the quantifier *everyone* as its subject would provide such a meaning, since it consumes a meaning for $\uparrow_\sigma$, the semantic structure for *everyone*, to produce another semantic resource which we can call H. Any other meaning constructor that consumes a meaning for $\uparrow_\sigma$ to produce another semantic structure H will also fill the bill.

As Saraswat (1999) notes, another way to think of the embedded implication in (58) is that the quantifier must perform a test on its environment to determine whether some implicational resource can be found which matches the required resource $\uparrow_\sigma \multimap H$. To perform this test, the quantifier proposes the resource $\uparrow_\sigma$, just as the abstraction rule given in (36) of this chapter allows a hypothetical resource to be proposed in order to create a function. If a resource H can then be obtained for some semantic structure H, the requirements of the quantifier are satisfied, and the conclusion H is valid.

8.1.2. Quantifier Scope Meaning

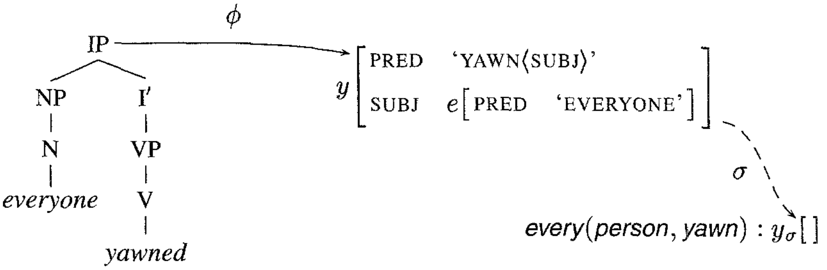
The meaning (left-hand) side of the lexical entry for *everyone* in (58) is:

(61) $\lambda S.\mathit{every}(X, \mathit{person}(X), S(X))$

In this expression, the expression $S(X)$ represents possible meanings for the scope of the quantifier.

To take a concrete example, we begin with the c-structure, f-structure, and meaning constructors for the sentence *Everyone yawned*, displayed in (62):

(62) *Everyone yawned.*



$$\begin{array}{lrl} \textbf{[everyone]} & \lambda S.\mathit{every}(X, \mathit{person}(X), S(X)) & : \forall H.[e_\sigma \multimap H] \multimap H \\ \textbf{[yawn]} & \lambda Y.\mathit{yawn}(Y) & : e_\sigma \multimap y_\sigma \end{array}$$

The right-hand side of the meaning constructor labeled **[everyone]** requires as its argument a meaning constructor of the form in (63):

(63) $e_\sigma \multimap H$

The glue side of the meaning constructor labeled **[yawn]** is of just this form, and the derivation is successful if the variable H for the scope semantic structure is instantiated to y_σ. Following the discussion in Section 7.2 of this chapter, we perform the glue deduction shown in example (66) (page 248), displaying only the glue sides of the meaning constructors. To determine the meaning that results from combining the meaning constructors labeled **[everyone]** and **[yawn]** according to the glue deduction in (66), we follow the function application rule presented in (35) of this chapter, applying the meaning of the quantifier $\lambda S.\mathit{every}(X, \mathit{person}(X), S(X))$ to its argument $\lambda Y.\mathit{yawn}(Y)$. The resulting meaning expression is:

(64) $\mathit{every}(X, \mathit{person}(X), [\lambda Y.\mathit{yawn}(Y)](X))$

or, equivalently:

(65) $\mathit{every}(X, \mathit{person}(X), \mathit{yawn}(X))$

(66) $\forall H.[e_\sigma \multimap H] \multimap H$

If we are given a resource $e_\sigma \multimap H$ for some semantic structure H, we can produce a resource for H.

$e_\sigma \multimap y_\sigma$

If we are given a resource e_σ corresponding to the SUBJ, we can produce a resource y_σ for the entire sentence.

---

y_σ

We have produced a resource y_σ for the full sentence.

In sum, assuming the meaning constructors shown in (62) for *everyone* and *yawned*, we can perform the following full glue deduction:

(67) $\lambda S.\mathit{every}(X, \mathit{person}(X), S(X)) : \forall H.[e_\sigma \multimap H] \multimap H$

On the glue side, if we are given a resource $e_\sigma \multimap H$ for some semantic structure H, we can produce a resource for H. On the meaning side, we apply the predicate $\lambda S.\mathit{every}(X, \mathit{person}(X), S(X))$ to the meaning corresponding to the resource $e_\sigma \multimap H$.

$\lambda Y.\mathit{yawn}(Y) : e_\sigma \multimap y_\sigma$

If we are given a resource e_σ corresponding to the SUBJ, we can produce a resource y_σ for the entire sentence. The meaning corresponding to this expression is $\lambda Y.\mathit{yawn}(Y)$.

$\mathit{every}(X, \mathit{person}(X), \mathit{yawn}(X)) : y_\sigma$

We have produced a resource y_σ for the full sentence, corresponding to the meaning $\mathit{every}(X, \mathit{person}(X), \mathit{yawn}(X))$, by assuming that H is the semantic structure y_σ.

We conclude that the sentence has the meaning $\mathit{every}(X, \mathit{person}(X), \mathit{yawn}(X))$, as desired.

8.1.3. Determination of Scope Semantic Structure

Example (67) shows that the variable H in the semantic constructor for the quantifier *everyone* can be instantiated to the semantic structure y_σ. In Section 4.1.4 of this chapter, we saw that the scope of a quantifier is not syntactically fixed: sentences with quantifiers may exhibit quantifier scope ambiguity. What are the possible semantic structures that can be chosen as the scope of a quantifier?

First, we note that the semantic structure that is chosen as the scope of a quantifier need not correspond to any f-structure constituent. For example, it has long been noted that the restriction of a quantifier can serve as the scope of another quantifier (Dalrymple et al. 1997b):

(68) *Every relative of a student attended.*

One reading of this sentence is:

(69) $\textit{every}(X, \textit{a}(Y, \textit{student}(Y), \textit{relative-of}(X, Y)), \textit{attend}(X))$

An abbreviated f-structure for this sentence is:

(70) *Every relative of a student attended.*

$$\left[\begin{array}{ll}
\text{PRED} & \text{`ATTEND}\langle\text{SUBJ}\rangle\text{'} \\
\text{SUBJ} & \left[\begin{array}{ll}
\text{SPEC} & \left[\text{PRED} \quad \text{`EVERY'}\right] \\
\text{PRED} & \text{`RELATIVE}\langle\text{OBL}_{\text{OF}}\rangle\text{'} \\
\text{OBL}_{\text{OF}} & \left[\begin{array}{ll}
\text{SPEC} & \left[\text{PRED} \quad \text{`A'}\right] \\
\text{PRED} & \text{`STUDENT'}
\end{array}\right]
\end{array}\right]
\end{array}\right]$$

Treating the determiner *a* as a quantifier, we see that its scope is $\textit{relative-of}(X, Y)$, the proposition that Y is a relative of X. This meaning corresponds roughly to the subphrase *relative of*, but does not correspond to an f-structure constituent. Instead, the more fine-grained semantic structure is the appropriate level to define quantifier scoping possibilities; this will become clear in our discussion of the meanings of determiners and common noun phrases in Section 8.2 of this chapter.

Second, we require the scope of the quantifier to contain the variable bound by the quantifier. That is, the scope of the quantifier must be a function of the argument position in which the quantifier appears. As noted by Dalrymple et al. (1997b), this follows without stipulation from our logical system: the embedded implication that the quantifier requires to determine its scope meaning must consume the meaning of the quantified noun phrase to produce the scope meaning.

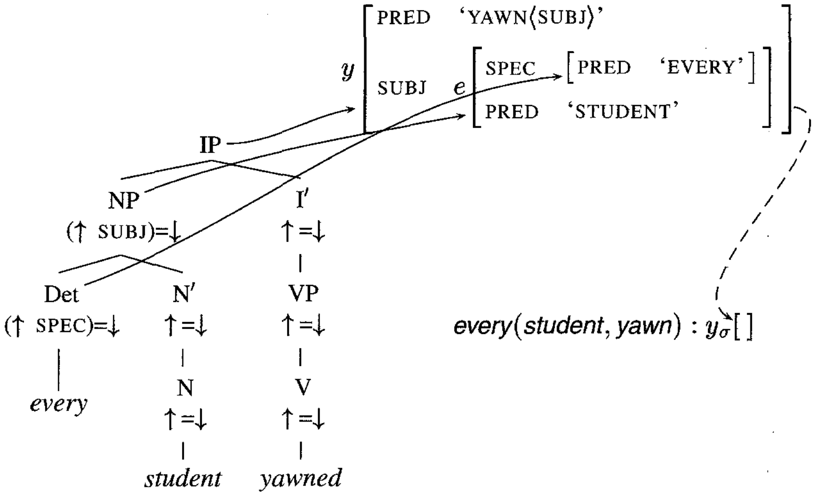
A number of other constraints on quantifier scoping have been proposed: quantifiers may be required to find their scope inside some syntactically definable domain, or to scope either inside or outside another quantifier. Since our focus here is not on a complete theory of quantification, we will not discuss constraints like these or show how they can be incorporated into the framework we propose. For detailed discussion of quantifier scoping constraints and a proposal for how they should be imposed in a glue setting, see Crouch and van Genabith (1999).

8.2. Determiners and Nouns

We now turn to an example involving a determiner and noun, *Every student yawned*. This example illustrates how the meanings of the determiner *every* and the common noun *student* are combined. As we will see, a deduction from the meaning constructors for *every* and *student* produces a meaning similar to the one proposed in (58) of this chapter for *everyone*, which can play a similar role in meaning assembly.

The c-structure, f-structure, and semantic representation for the sentence *Every student yawned* are displayed in (71):

(71) *Every student yawned.*



We propose the lexical entry in (72) for the determiner *every*:

(72) *every* Det (↑ PRED) = 'EVERY'

$$\lambda R.\lambda S.\mathit{every}(X, R(X), S(X)) :$$
$$[((\textsc{spec}\uparrow)_\sigma\ \textsc{var}) \multimap ((\textsc{spec}\uparrow)_\sigma\ \textsc{restr})] \multimap$$
$$[\forall H.[(\textsc{spec}\uparrow)_\sigma \multimap H] \multimap H]$$

The meaning constructor for *every* uses *inside-out functional uncertainty* (Chapter 6, Section 1.2) to refer to the f-structure for the noun phrase that contains it. The expression (SPEC ↑) in this entry refers to the f-structure in which *every* appears as the SPEC value, which is the f-structure labeled *e* in (71).

The lexical entry for the common noun *student* is given in (73):

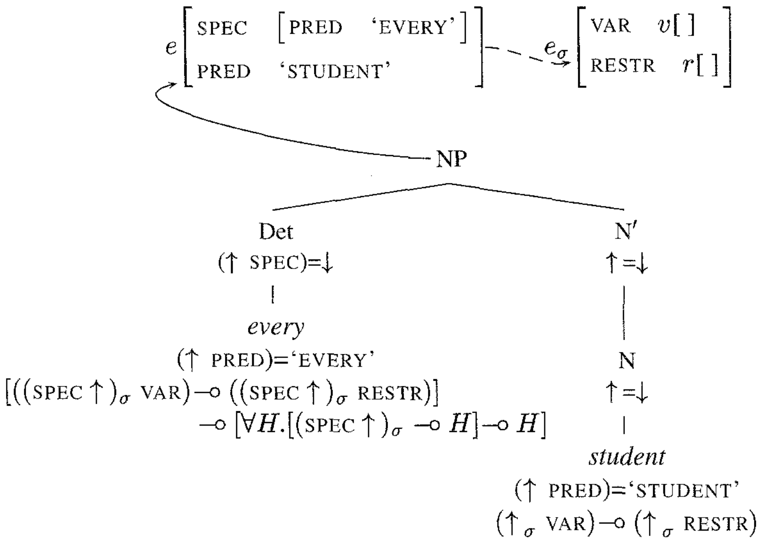
(73) *student* N (↑ PRED) = 'STUDENT'

$$\lambda X.\mathit{student}(X) : (\uparrow_{\sigma}\ \text{VAR}) \multimap (\uparrow_{\sigma}\ \text{RESTR})$$

The lexical entries in (72) and (73) indicate that the semantic structure e_σ corresponding to the SUBJ f-structure is complex and has internal structure; it contains two attributes, VAR and RESTR, with semantic structures as their values. The attribute VAR represents a variable of type e, and the attribute RESTR represents a restriction on that variable of type t: in this case, that the variable must range over individuals that are students.

These lexical entries, together with the standard English phrase structure rules, give rise to the structures shown in (74); to save space, only the glue sides of the meaning constructors for *every* and *student* are displayed, and the meaning sides are omitted:

(74) *every student*


SPEC [PRED 'EVERY']
e
PRED 'STUDENT'
e_σ
VAR v[ ]
RESTR r[ ]
NP
Det
(↑ SPEC)=↓
every
(↑ PRED)='EVERY'
$[((\text{SPEC}\uparrow)_\sigma\ \text{VAR}) \multimap ((\text{SPEC}\uparrow)_\sigma\ \text{RESTR})]$
$\multimap [\forall H.[(\text{SPEC}\uparrow)_\sigma \multimap H] \multimap H]$
N′
↑=↓
N
↑=↓
student
(↑ PRED)='STUDENT'
$(\uparrow_\sigma\ \text{VAR}) \multimap (\uparrow_\sigma\ \text{RESTR})$

Instantiating the ↑ and ↓ variables and using the labels v and r for the semantic structures $(e_\sigma\ \text{VAR})$ and $(e_\sigma\ \text{RESTR})$, we have the meaning constructors in (75), labeled **[every]** and **[student]**:

(75) Meaning constructor premises for *every student*:

[every] $\lambda R.\lambda S.\mathit{every}(X, R(X), S(X)) : [v \multimap r] \multimap [\forall H.[e_\sigma \multimap H] \multimap H]$

[student] $\lambda X.\mathit{student}(X) : v \multimap r$

The meaning constructor for **[every]** requires two arguments: just as a transitive verb needs two semantic contributions, one from its subject and one from its object, a quantifier like *every* needs a semantic contribution from its restriction (the meaning of the common noun and any modifiers it might have) and its scope.

The first requirement is for a meaning for the restriction of the quantifier:

(76) $v \multimap r$

This requirement exactly matches the contribution of the common noun *student*, and the meaning of *student* becomes the restriction of the quantifier *every*.

The second requirement for the quantifier *every* is a meaning for its scope:

(77) $e_\sigma \multimap H$

As described in Section 8.1 of this chapter for the quantifier *everyone*, the quantifier requires a contribution of the form $e_\sigma \multimap H$, whose meaning corresponds to the scope meaning S of *every*.

We can now deduce the meaning constructor for *every student* from the meaning constructors for *every* and for *student*:

(78) Combining the meanings of *every* and *student*:

$\lambda R.\lambda S.\mathit{every}(X, R(X), S(X)) : [v \multimap r] \multimap [\forall H.[e_\sigma \multimap H] \multimap H]$

The meaning constructor for *every* requires a resource $v \multimap r$ corresponding to its restriction meaning R, and a resource $e_\sigma \multimap H$ corresponding to its scope meaning S, to produce a resource H for its scope semantic structure.

$\lambda X.\mathit{student}(X) : v \multimap r$

The meaning constructor for *student* provides an implicational resource $v \multimap r$ corresponding to the meaning $\lambda X.\mathit{student}(X)$.

$\lambda S.\mathit{every}(X, \mathit{student}(X), S(X)) : \forall H.[e_\sigma \multimap H] \multimap H$

Therefore, by combining the meanings of *every* and *student*, we get a result that is like the meaning constructor for *everyone*, except that the restriction of the quantifier *every* is specified to involve students.

The resulting meaning constructor for *every student* is, as desired, of the same rough shape as the meaning for *everyone*, since in terms of meaning construction, they behave alike; only the meanings associated with the semantic structures differ.

Completing the deduction, we have the meaning *every*(*student*, *yawn*) for this sentence, which is the desired result:

(79) *Every student yawned.*

$$y\left[\begin{array}{ll}\textsc{pred} & \text{`}\textsc{yawn}\langle\textsc{subj}\rangle\text{'}\\[1ex] \textsc{subj} & e\left[\begin{array}{ll}\textsc{spec} & \left[\begin{array}{ll}\textsc{pred} & \text{`}\textsc{every}\text{'}\end{array}\right]\\ \textsc{pred} & \text{`}\textsc{student}\text{'}\end{array}\right]\end{array}\right]$$

$$\begin{array}{ll}\textbf{[every]} & \lambda R.\lambda S.\mathit{every}(X, R(X), S(X)) \ : \\ & \qquad [(e_\sigma\ \textsc{var})\multimap(e_\sigma\ \textsc{restr})]\multimap[\forall H.[e_\sigma \multimap H]\multimap H]\\[1ex] \textbf{[student]} & \lambda X.\mathit{student}(X) \ : \ (e_\sigma\ \textsc{var})\multimap(e_\sigma\ \textsc{restr})\\[1ex] \textbf{[yawn]} & \lambda X.\mathit{yawn}(X) \ : \ e_\sigma\multimap y_\sigma\end{array}$$

$$\textbf{[every]}, \textbf{[student]}, \textbf{[yawn]} \vdash \mathit{every}(X, \mathit{student}(X), \mathit{yawn}(X)) : y_\sigma$$

With this basis in the theory of meaning assembly, we are now ready to begin an exploration of the syntax and semantics of a variety of linguistic constructions. In the next five chapters, we will discuss the syntax and semantics of modification (Chapter 10); syntactic constraints on the anaphor-antecedent relation and the semantics of binding (Chapter 11); the syntax and semantics of functional and anaphoric control in constructions with raising and equi verbs (Chapter 12); the syntax of constituent and nonconstituent coordination, resource sharing at the syntax-semantics interface, and the syntax and semantics of noun phrase coordination (Chapter 13); and the syntax of long-distance dependencies and the semantics of relative clauses and wh-questions (Chapter 14).

9. FURTHER READING AND RELATED ISSUES

This chapter has been devoted to an exploration of linguistic meaning and the syntax-semantics interface. The intention has been to give the reader the linguistic intuitions behind the analyses, and we have not emphasized the formal and mathematical properties of the glue language, linear logic. The presentation of analyses in subsequent chapters is also aimed primarily at an intuitive understanding of how meaning deductions work. It is important to keep in mind, however, that despite the informal nature of the presentation here and in the following chapters, our theory of meaning composition is grounded in a mathematically precise, rigorously defined logic. We will not give a more technically oriented introduction or overview discussion of linear logic in this volume, since such material is readily available from other sources. Dalrymple et al. (1999b) give a more detailed introduction to linear logic in the current setting (see also Dalrymple et al. 1995e). For the logically inclined, the appendix presents the proof rules for our fragment of linear logic. Linear logic originated in the work of Girard (1987); a very accessible general overview is given by Scedrov (1993), and Crouch and van Genabith (2000) provide an in-depth treatment with a linguistic orientation.

We will also omit discussion of proof methods or algorithms for deduction in linear logic; again, this material is widely available for consultation by those interested in formal and computational aspects of glue theory. Girard (1987) introduced the notion of *proof nets* for proofs in the fragment of linear logic we use; for a lucid description of the use of proof nets for deduction in the glue approach, see Fry (1999a). Efficient proof techniques for glue semantic deductions are also explored by Lamping and Gupta (1998).

Besides the work mentioned in this chapter, there are a number of papers on linguistic issues relating to glue theory. The papers in Dalrymple (1999) provide an overview of the theory as well as discussions of formal aspects of the theory and particular linguistic phenomena. Included are treatments of quantifier scoping constraints (Crouch and van Genabith 1999), intensionality and quantifier scope (Dalrymple et al. 1997b), negative polarity (Fry 1999a), and dynamic and underspecified semantics (van Genabith and Crouch 1999a). Additional work within the glue framework includes work on ellipsis (Crouch 1999), translation within the semantic framework of Underspecified Discourse Representation Theory (Crouch et al. 2001), event semantics (Fry 1999b), and the German split NP construction (Kuhn 2001b).

10

MODIFICATION

This chapter explores issues in the syntax and semantics of modification. Since there is in principle no limit to the number of modifiers that a phrase can have, we represent modifiers at functional structure as members of a set of modifying adjuncts ADJ (Chapter 2, Section 3.4). Functional annotations on c-structure rules ensure that each modifier appears as a member of the adjunct set associated with the phrase it modifies.

In the following, we will concentrate in particular on adjectival modification, since the syntax and semantics of adjectives is fairly complex and illustrates many of the issues of interest to us. Section 1 of this chapter provides an overview of the syntax of adjectival modification, Section 2 discusses three semantic classes of adjectives and how their meanings are represented, and Section 3 discusses adjectival modification at the syntax-semantics interface within the glue approach.

Defining the semantic contribution of a modifier brings up a set of tricky problems, as first noticed by Kasper (1995). In Section 4, we will address these issues and show that they have a straightforward solution within our framework.

The chapter concludes with a brief examination of the syntax and semantics of adverbial modification: Section 5 discusses the syntax and semantics of manner adverbials like *skillfully* as well as sentential adverbs like *necessarily*.

1. SYNTAX OF ADJECTIVAL MODIFICATION

1.1. Modification at Functional Structure

As discussed in Chapter 2, Section 1.2, modifiers are different from arguments in that they can be multiply specificational:

(1) a. *The girl handed the baby a toy on Tuesday in the morning.*

b. **David saw Tony Mr. Gilroy my next-door neighbor.*

At f-structure, each modifier is a member of the set of modifiers of a phrase. In example (2), the adjectival modifier *Swedish* is treated as a member of the modifying adjunct set ADJ of modifiers of the noun *man*:

(2) *Swedish man*

$$\left[\begin{array}{ll}\textsc{pred} & \text{‘}\textsc{man}\text{’}\\ \textsc{adj} & \left\{\left[\begin{array}{ll}\textsc{pred} & \text{‘}\textsc{Swedish}\text{’}\end{array}\right]\right\}\end{array}\right]$$

In phrases with more than one modifier, the f-structure for each modifier appears as a member of the ADJ set:

(3) *tall Swedish man*

$$\left[\begin{array}{ll}\textsc{pred} & \text{‘}\textsc{man}\text{’}\\ \textsc{adj} & \left\{\begin{array}{l}\left[\begin{array}{ll}\textsc{pred} & \text{‘}\textsc{tall}\text{’}\end{array}\right]\\ \\ \left[\begin{array}{ll}\textsc{pred} & \text{‘}\textsc{Swedish}\text{’}\end{array}\right]\end{array}\right\}\end{array}\right]$$

The lexical entries for *tall*, *Swedish*, and *man* contain at least the following syntactic information:

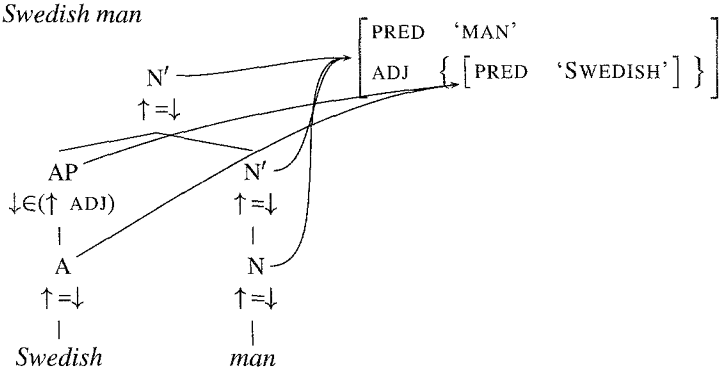
(4)

tall	A	$(\uparrow \textsc{pred}) = \text{‘}\textsc{tall}\text{’}$
Swedish	A	$(\uparrow \textsc{pred}) = \text{‘}\textsc{Swedish}\text{’}$
man	N	$(\uparrow \textsc{pred}) = \text{‘}\textsc{man}\text{’}$

1.2. Constituent Structure Constraints

At constituent structure, modifiers are often adjoined to the phrases they modify (Chapter 3, Section 4.2). The c-structure and f-structure for the English noun phrase *Swedish man* is shown in (5), with the modifier *Swedish* adjoined at the N′ level:

(5) *Swedish man*

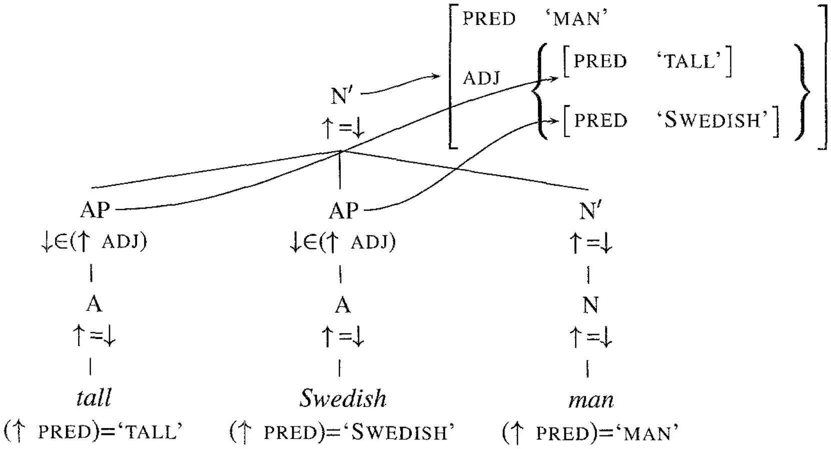


We propose the following adjunction rule for adjective phrase modifiers in English:

(6) $$N' \longrightarrow \underset{\downarrow\in(\uparrow\ \text{ADJ})}{\text{AP}^*} \quad \left(\underset{\uparrow=\downarrow}{N'}\right)$$

This rule, which supplements the rule in which N′ dominates only its head N and any arguments of N, allows for any number of adjectives to be adjoined at the N′ level. At f-structure, each modifying adjective is a member of the modifying adjunct set ADJ. In example (7), two adjectives have been adjoined:

(7) *tall Swedish man*



2. SEMANTIC CLASSES OF ADJECTIVES

Influential work on the semantics of adjectives was done by Montague (1974a) and Kamp (1975), who focused primarily on the three types of adjectives to be examined in this section. Their basic view of the semantics of adjectival modification has been widely adopted.

As discussed in Chapter 9, Section 8.2, the meaning of the proper noun *man* is:

(8) *man*

$\lambda X.\mathit{man}(X)$

This meaning is of type $\langle e \rightarrow t \rangle$. It picks out the set of men — that is, the set of entities X for whom the proposition $\mathit{man}(X)$ is true. When a meaning like the one in (8) is modified, the result is a meaning which is of the same type but which reflects a modified meaning rather than the original unmodified meaning. In the following, we describe how noun meanings are modified in different ways by different semantic classes of adjectives.

The meaning of *Swedish man* can be represented as in (9), in which the conjunction operator $\wedge$ conjoins the two expressions $\mathit{Swedish}(X)$ and $\mathit{man}(X)$:

(9) *Swedish man*

$\lambda X.\mathit{Swedish}(X) \wedge \mathit{man}(X)$

The type of this meaning is $\langle e \rightarrow t \rangle$, just like the unmodified meaning *man*; the difference in meaning is that this expression picks out the set of individuals X that satisfy both the predicate $\mathit{Swedish}(X)$ and the predicate $\mathit{man}(X)$ — the individuals that are both Swedish and men. Adjectives like *Swedish* are called *intersective*, since the individuals that are Swedish men are those that are in the intersection of the set of individuals that are Swedish with the set of individuals that are men.

Adjectives like *big* or *tall* are called *gradable adjectives*. As noted by Montague (1974a), Kamp (1975), Siegel (1976), Kennedy (1997), and many others, gradable adjectives like *big* or *tall* must be interpreted relative to some relevant standard. For example, some individual mouse might count as a *big mouse*, even though the same mouse is probably not a big animal or even a big rodent. Similarly, a second-grade boy can be correctly characterized as a *tall second-grader* even if he is not tall compared to an adult.

We propose the following simplified meaning for *big mouse* (see Kennedy 1997 for a full discussion of the semantics of gradability and comparison):

(10) *big mouse*

$\lambda X.\mathit{big}(X, \mathcal{P}) \wedge \mathit{mouse}(X)$

The argument $\mathcal{P}$ of big represents the property that determines the relevant standard of measurement; as Kennedy (1997) shows, the standard according to which gradable adjectives are interpreted is determined by some contextually salient property of the individual. If the contextually salient property $\mathcal{P}$ of the individual is that it is a mouse, modification by the adjective *big* requires the individual to exceed some standard of size that is determined by reference to mousehood. In other words, if something is big relative to the property of being a mouse, we need to know the range of sizes that are appropriate for mice, and we need to know that this individual is bigger than a standard-size mouse.

In a neutral context, the contextually relevant property is often the property denoted by the modified noun; for example, the contextually salient property $\mathcal{P}$ in an example like *big mouse* is generally resolved to the property of being a mouse. However, as pointed out by McConnell-Ginet (1979) and Pollard and Sag (1994), in certain contexts other interpretations are also possible. Pollard and Sag provide the following example:

(11) *The Linguistics Department has an important volleyball game coming up against the Philosophy Department. I see the Phils have recruited Julius to play with them, which means we are in real trouble unless we can find a good linguist to add to our team in time for the game.*

Here the property $\mathcal{P}$ relevant to the interpretation of the adjective *good* is being a volleyball player, since in this example *good linguist* means, more or less, *linguist that is good at playing volleyball*. Examples such as these show that the property $\mathcal{P}$ need not correspond to the property denoted by the modified noun, but is determined contextually.

Of course, modified phrases can undergo further modification. The meaning of the doubly modified phrase *tall Swedish man* is:

(12) *tall Swedish man*

$$\lambda X.\mathit{tall}(X,\mathcal{P}) \wedge \mathit{Swedish}(X) \wedge \mathit{man}(X)$$

Even in a neutral context, the contextually relevant property $\mathcal{P}$ involved in the interpretation of the adjective *tall* can be resolved in several ways. It can refer to someone who is Swedish, a man, and tall for a man, in which case the contextually relevant property $\mathcal{P}$ is the property of being a man. It can also refer to someone who is Swedish, a man, and tall for a Swedish man, in which case the contextually relevant property $\mathcal{P}$ is the property of being a Swedish man.

Another class of modifying adjectives, studied by Kamp (1975) and in more detail by Siegel (1976), is the class of *intensional adjectives* such as *imaginary*, *former*, *fake*, and *alleged*. These adjectives are different from those discussed in the previous section in an important way: a *Swedish man* is a man, and a *big mouse* is a mouse, but a *fake gun* is not a gun; instead, it may actually be a toy or a piece of soap. Thus, the meaning of a phrase with an intensional adjective depends on the meaning of the unmodified phrase, but the resulting property may hold of an individual even if the unmodified meaning does not.

Like other adjectives, an intensional adjective operates on the description it modifies and produces a new description of the same type:

(13) *former senator*

$\lambda X.\textit{former}(\textit{senator}, X)$

A former senator is one who at some previous time was a senator, but who is no longer a senator; the meaning of *senator* is important in understanding the meaning of *former senator*, but the individuals represented by X in the meaning given in (13) for *former senator* are not required to be senators. Thus, *former* in (13) denotes a relation between the property of being a senator and some individual who formerly had that property. Similarly, a fake gun is an entity that is not a gun, but which has some properties in common (for example, appearance) with entities that are actually guns; again, although a fake gun is not a gun, the meaning of *gun* is important in determining the meaning of *fake gun*:

(14) *fake gun*

$\lambda X.\textit{fake}(\textit{gun}, X)$

Importantly, the resulting meaning still has type $\langle e \rightarrow t\rangle$; intensional adjectives, like intersective adjectives and gradable adjectives, turn an unmodified $\langle e \rightarrow t\rangle$ meaning into a modified $\langle e \rightarrow t\rangle$ meaning. This characteristic is shared by all modifiers and will be important in our analysis of modification and meaning composition.

3. MODIFIERS AND SEMANTIC COMPOSITION

As Montague (1974a) and Kamp (1975) point out, adjectival modifiers are functions that take a property of type $\langle e \rightarrow t\rangle$ (such as the property of being a man) and produce a new property (such as the property of being a Swedish man). This intuition is reflected in the glue semantic premises contributed by modifiers.

3.1. Adjectival Modification

As shown in Chapter 9, Section 8.2, a common noun like *man* is associated with the syntactic and semantic structures and meaning constructor given in (15), where the semantic structures v and r are the values of the attributes VAR and RESTR in the semantic structure f_σ:

(15) *man*

$$f\begin{bmatrix}\textsc{pred} & \text{`}\textsc{man}\text{'}\end{bmatrix} \dashrightarrow f_\sigma\begin{bmatrix}\textsc{var} & v[\;]\\ \textsc{restr} & r[\;]\end{bmatrix}$$

$$\lambda X.man(X) : v \multimap r$$

A modified noun like *Swedish man* is associated with a meaning constructor whose right-hand side is exactly the same as the meaning constructor for *man*, but whose left-hand side is associated with a modified meaning rather than an unmodified one:

(16) *Swedish man*

$$f\begin{bmatrix}\textsc{pred} & \text{`}\textsc{man}\text{'}\\ \textsc{adj} & \left\{\begin{bmatrix}\textsc{pred} & \text{`}\textsc{swedish}\text{'}\end{bmatrix}\right\}\end{bmatrix} \dashrightarrow f_\sigma\begin{bmatrix}\textsc{var} & v[\;]\\ \textsc{restr} & r[\;]\end{bmatrix}$$

$$\lambda X.Swedish(X) \wedge man(X) : v \multimap r$$

In this section, we show how a meaning constructor like the one in (16) is derived from the meaning constructors for *Swedish* and *man*.

The lexical entries for *Swedish* and *man*, augmented with meaning constructors, are given in (17):

(17) *man* $(\uparrow\ \textsc{pred}) = \text{`}\textsc{man}\text{'}$
$\lambda X.man(X) : (\uparrow_\sigma\ \textsc{var}) \multimap (\uparrow_\sigma\ \textsc{restr})$

Swedish $(\uparrow\ \textsc{pred}) = \text{`}\textsc{swedish}\text{'}$
$\lambda P.\lambda X.Swedish(X) \wedge P(X)$:
$[((\textsc{adj}\ \in\ \uparrow)_\sigma\ \textsc{var}) \multimap ((\textsc{adj}\ \in\ \uparrow)_\sigma\ \textsc{restr})] \multimap$
$[((\textsc{adj}\ \in\ \uparrow)_\sigma\ \textsc{var}) \multimap ((\textsc{adj}\ \in\ \uparrow)_\sigma\ \textsc{restr})]$

The meaning constructor for *man* is familiar from our discussion of common nouns in Chapter 9, Section 8.2. The meaning constructor for *Swedish* uses *inside-out functional uncertainty* (Chapter 6, Section 1.2) to refer to the semantic structure of the phrase it modifies. The expression $(\textsc{adj}\ \in\ \uparrow)$ refers to the f-structure in which $\uparrow$ appears as a member of the modifier set,[1] the expression $(\textsc{adj}\ \in\ \uparrow)_\sigma$ refers to the semantic structure corresponding to that f-structure, and the expression $((\textsc{adj}\ \in\ \uparrow)_\sigma\ \textsc{var})$ refers to the value of the attribute VAR in that semantic structure, labeled v in (16) above. Similarly, the expression $((\textsc{adj}\ \in\ \uparrow)_\sigma\ \textsc{restr})$ refers to the value of the RESTR attribute, labeled r.

Instantiating the meaning constructors in (17) according to the labels on the structures displayed in (16), we have the following instantiated meaning constructors for *Swedish* and *man*:

[1] The use of the set membership symbol $\in$ as an attribute is discussed in Chapter 6, Section 2.1.

(18) Meaning constructor premises for *Swedish man*:

[man] $\lambda X.\mathit{man}(X) \; : \; v \multimap r$

[Swedish] $\lambda P.\lambda X.\mathit{Swedish}(X) \wedge P(X) \; : \; [v \multimap r] \multimap [v \multimap r]$

The right-hand side of the meaning constructor for *Swedish* illustrates the characteristic glue contribution of a modifier: it requires a resource of the form $v \multimap r$ as its argument and produces a resource of exactly the same form. The general form for modifiers is given in (19), where $\mathcal{M}$ is the meaning of the modifier and S is the glue contribution of the phrase it modifies:

(19) $\mathcal{M} : S \multimap S$

Modifiers consume a meaning resource S and produce an identical new meaning resource S for the phrases they modify.

Given the premises **[Swedish]** and **[man]**, we can perform a deduction that produces the meaning constructor for *Swedish man* given in (16).

(20) $\lambda X.\mathit{man}(X) : v \multimap r$

The meaning $\lambda X.\mathit{man}(X)$ is associated with the implicational contribution $v \multimap r$.

$\lambda P.\mathit{Swedish}(X) \wedge P(X) : [v \multimap r] \multimap [v \multimap r]$

On the glue side, the meaning constructor consumes the noun contribution $v \multimap r$ and produces a new modified meaning which is also associated with $v \multimap r$. On the meaning side, we apply the function $\lambda P.\mathit{Swedish}(X) \wedge P(X)$ to the unmodified meaning contributed by *man*, $\lambda X.\mathit{man}(X)$.

$\lambda X.\mathit{Swedish}(X) \wedge \mathit{man}(X) : v \multimap r$

We have produced a modified meaning $\lambda X.\mathit{Swedish}(X) \wedge \mathit{man}(X)$ associated with the implicational contribution $v \multimap r$.

We can also represent this deduction in abbreviated form, as shown in Chapter 9, using the labels in (18):

(21) **[Swedish]**, **[man]** $\vdash \lambda X.\mathit{Swedish}(X) \wedge \mathit{man}(X) : v \multimap r$

3.2. Gradable Adjectives

Gradable adjectives like *big* differ from intersective adjectives like *Swedish* in introducing a contextually salient property $\mathcal{P}$ in their interpretation:

(22) *big mouse*

$$f\left[\begin{array}{ll}\text{PRED} & \text{`MOUSE'}\\ \text{ADJ} & \left\{\left[\text{PRED}\quad \text{`BIG'}\right]\right\}\end{array}\right] \overset{f_\sigma}{\dashrightarrow} \left[\begin{array}{ll}\text{VAR} & v[\,]\\ \text{RESTR} & r[\,]\end{array}\right]$$

$$\lambda X.\mathit{big}(X,\mathcal{P}) \wedge \mathit{mouse}(X) : v \multimap r$$

The meaning contribution of *big mouse* given in (22) refers to a mouse that exceeds the size of individuals that are described by the contextually determined property $\mathcal{P}$. Since the property $\mathcal{P}$ is determined by contextual factors, not syntactically, we will not specify a means for determining $\mathcal{P}$ but instead will leave it uninstantiated.

Although the meaning contribution of a gradable adjective like *big* is not the same as that of an intersective adjective like *Swedish*, the right-hand sides of the two meaning constructors are the same, since the two kinds of adjective play a similar role in meaning assembly. The lexical entry for *big* is given in (23):

(23) *big* $(\uparrow \text{PRED}) = \text{`BIG'}$
$\lambda R.\lambda X.\mathit{big}(X,\mathcal{P}) \wedge R(X)$:
$[((\text{ADJ} \in \uparrow)_\sigma\ \text{VAR}) \multimap ((\text{ADJ} \in \uparrow)_\sigma\ \text{RESTR})] \multimap$
$[((\text{ADJ} \in \uparrow)_\sigma\ \text{VAR}) \multimap ((\text{ADJ} \in \uparrow)_\sigma\ \text{RESTR})]$

Like the entry for *Swedish* given in (17) earlier, this entry uses inside-out functional uncertainty to refer to the f-structure of the phrase it modifies. The lexical entry for *mouse* is exactly analogous to the one for *man* and will not be displayed.

Instantiating the lexical entries for *big* and *mouse* according to the labels in (22), we have the instantiated meaning constructors in (24):

(24) Meaning constructor premises for *big mouse*:

[mouse] $\lambda X.\mathit{mouse}(X) : v \multimap r$

[big] $\lambda R.\lambda X.\mathit{big}(X,\mathcal{P}) \wedge R(X) : [v \multimap r] \multimap [v \multimap r]$

The meaning constructor for *big* requires a meaning resource of the form $v \multimap r$; *mouse* provides such a resource. The resulting meaning is obtained by applying the expression $\lambda R.\lambda X.\mathit{big}(X,\mathcal{P}) \wedge R(X)$ to its argument $\lambda X.\mathit{mouse}(X)$. The result is as desired — from the meaning constructors labeled **[big]** and **[mouse]** in (24), we derive the meaning constructor for *big mouse*:

(25) **[big]**, **[mouse]** $\vdash \lambda X.\mathit{big}(X,\mathcal{P}) \wedge \mathit{mouse}(X) : v \multimap r$

3.3. Intensional Adjective Modification

The syntactic and semantic structures and meaning constructor for the phrase *former senator* are as shown in (26):

(26) *former senator*

$$f\begin{bmatrix}\text{PRED} & \text{`SENATOR'}\\ \text{ADJ} & \left\{\begin{bmatrix}\text{PRED} & \text{`FORMER'}\end{bmatrix}\right\}\end{bmatrix} \dashrightarrow f_\sigma\begin{bmatrix}\text{VAR} & v[\,]\\ \text{RESTR} & r[\,]\end{bmatrix}$$

$\lambda X.\mathit{former}(\mathit{senator}, X) : v \multimap r$

The lexical entry of *former* is given in (27):

(27) *former* $(\uparrow \text{PRED}) = \text{`FORMER'}$

$$\begin{array}{l}\lambda P.\lambda X.\mathit{former}(P, X):\\ \quad [((\text{ADJ} \in \uparrow)_\sigma\ \text{VAR}) \multimap ((\text{ADJ} \in \uparrow)_\sigma\ \text{RESTR})] \multimap\\ \qquad [((\text{ADJ} \in \uparrow)_\sigma\ \text{VAR}) \multimap ((\text{ADJ} \in \uparrow)_\sigma\ \text{RESTR})]\end{array}$$

As shown earlier, the meaning contribution of an intensional adjective like *former* is different from *Swedish* and *big*. Nevertheless, it contributes a meaning resource of the same form: it consumes a resource corresponding to the phrase it modifies and produces a new resource of the same form. The instantiated meaning constructors for *former* and *senator* are given in (28):

(28) Meaning constructor premises for *former senator*:

[senator]	$\lambda X.\mathit{senator}(X)$	:	$v \multimap r$
[former]	$\lambda P.\lambda X.\mathit{former}(P, X)$	:	$[v \multimap r] \multimap [v \multimap r]$

As desired, these meaning constructors combine to produce the meaning constructor for *former senator* given in (26):

(29) **[former], [senator]** $\vdash \lambda X.\mathit{former}(\mathit{senator}, X) : v \multimap r$

Although each type of modifier makes a different kind of contribution to meaning, their roles in meaning assembly are similar; this is reflected in the meaning resources on the right-hand sides of the meaning constructors for the modifying adjectives we have examined.

4. RECURSIVE MODIFICATION

In the foregoing, we have assumed that the function of a modifier is to specify the result that is obtained when it combines with the phrase it modifies — in other words, that the meaning of an adjective is defined in terms of its effect on the element that it modifies. This common assumption is challenged in an important paper by Kasper (1995), who discusses evidence from *recursive modification*, cases in which a modifier is itself modified. In this section, we review Kasper's observations and show how they are accounted for in the glue approach.

Consider a modifier like *Swedish*, which we have assumed to have a meaning constructor like the one shown in (30):

(30) *Swedish*

$$f\begin{bmatrix}\text{ADJ} & \left\{\begin{bmatrix}\text{PRED} & \text{`SWEDISH'}\end{bmatrix}\right\}\end{bmatrix} \dashrightarrow f_{\sigma}\begin{bmatrix}\text{VAR} & v[\;] \\ \text{RESTR} & r[\;]\end{bmatrix}$$

$$\lambda P.\lambda X.\mathit{Swedish}(X) \wedge P(X) : [v \multimap r] \multimap [v \multimap r]$$

The meaning constructor for *Swedish* given in (30) provides information about how to determine the meaning of the phrase it modifies. It does not provide a representation for the meaning of *Swedish* independent of its modifying effect; instead, it represents only the conjunctive meaning that results from combining *Swedish* with the phrase it modifies.

Kasper (1995) shows that this view is inadequate by considering examples like (31):

(31) *apparently Swedish*

$$f\begin{bmatrix}\text{ADJ} & \left\{ g\begin{bmatrix}\text{PRED} & \text{`SWEDISH'} \\ \text{ADJ} & \left\{ h\begin{bmatrix}\text{PRED} & \text{`APPARENTLY'}\end{bmatrix} \right\}\end{bmatrix} \right\}\end{bmatrix} \dashrightarrow f_{\sigma}\begin{bmatrix}\text{VAR} & v[\;] \\ \text{RESTR} & r[\;]\end{bmatrix}$$

$$\lambda P.\lambda X.\mathit{apparently}(\mathit{Swedish}(X)) \wedge P(X) : v \multimap r$$

In this example, the modifier *Swedish* is itself modified by the adverb *apparently*. The effect of modification by *apparently* is to modify the proposition that X is Swedish, $\mathit{Swedish}(X)$, to produce a new proposition $\mathit{apparently}(\mathit{Swedish}(X))$. However, the proposition $\mathit{Swedish}(X)$ is not by itself associated with the meaning of the adjective *Swedish*, and in fact there is no obvious way to disentangle the meaning $\mathit{Swedish}(X)$ from the rest of the meaning contribution for *Swedish* in (30).

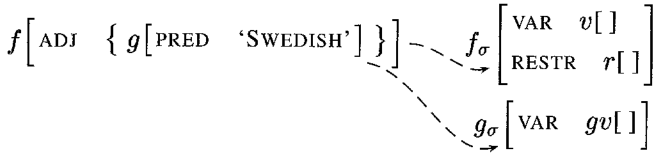
For a meaning like $\mathit{Swedish}(X)$ to be available, we require an independent, modifiable characterization of the intrinsic meaning of *Swedish*, together with a theory of how this meaning combines with the meaning of the modified noun. Kasper (1995) provides an analysis of examples like (31) within the framework of Head-Driven Phrase Structure Grammar. Though it is stated in different formal terms, our analysis has a clear basis in Kasper's intuitions.

4.1. Meaning Constructors for Modifiers

To provide a full account of adjectival modification, we assume that the semantic structures of adjectives are internally structured, containing the attribute VAR. In (32), the f-structure f corresponds to a semantic structure f_{σ} with the attributes

VAR and RESTR; as shown earlier, the values of these attributes are labeled v and r. The f-structure g of the adjective *Swedish* also has an attribute VAR, whose value we have labeled gv:

(32) *Swedish*



The intrinsic meaning of the adjective *Swedish* is of type $\langle e \rightarrow t\rangle$. Since we assume that the basic types e and t are associated with semantic structures, we assign the type e to gv and the type t to g_σ.

We now refine our assumptions about the meaning contributions of modifiers: we propose that adjectives make two separate meaning contributions. The first meaning constructor for the adjective *Swedish* in the lexical entry in (33) contributes the intrinsic meaning of the modifier, while the second meaning constructor provides instructions for combining the first meaning constructor with the noun it modifies:

(33) Lexical entry for *Swedish* (final)

Swedish $(\uparrow \text{PRED}) = \text{`SWEDISH'}$

$\lambda X.\mathit{Swedish}(X) : [(\uparrow_\sigma \text{VAR}) \multimap \uparrow_\sigma]$

$\lambda Q.\lambda P.\lambda X.Q(X) \wedge P(X) :$
$[(\uparrow_\sigma \text{VAR}) \multimap \uparrow_\sigma] \multimap$
$[[((\text{ADJ} \in \uparrow)_\sigma \text{VAR}) \multimap ((\text{ADJ} \in \uparrow)_\sigma \text{RESTR})] \multimap$
$[((\text{ADJ} \in \uparrow)_\sigma \text{VAR}) \multimap ((\text{ADJ} \in \uparrow)_\sigma \text{RESTR})]]$

Instantiating these two meaning constructors according to the labels given in (32) makes them much easier to read; we have labeled the first meaning constructor in the lexical entry in (33) **[Swedish1]** and the second **[Swedish2]**:

(34) Meaning constructor premises for *Swedish*:

[Swedish1] $\lambda X.\mathit{Swedish}(X) : [gv \multimap g_\sigma]$

[Swedish2] $\lambda Q.\lambda P.\lambda X.Q(X) \wedge P(X) : [gv \multimap g_\sigma] \multimap [[v \multimap r] \multimap [v \multimap r]]$

Importantly, we can deduce the meaning constructor for *Swedish* given in (18) from the two meaning constructors in (34). The meaning constructor **[Swedish1]** provides the semantic resource $gv \multimap g_\sigma$ that is required by **[Swedish2]**, and the resulting meaning is obtained by function application: the meaning contribution of **[Swedish1]**, $\lambda Q.\lambda P.\lambda X.Q(X) \wedge P(X)$, is applied to the meaning contribution of **[Swedish2]**, $\lambda X.\mathit{Swedish}(X)$.

(35) **[Swedish1], [Swedish2]** $\vdash$ **[Swedish]**

Therefore, the two meaning constructors **[Swedish1]** and **[Swedish2]** can play exactly the same role in meaning assembly as the simple meaning constructor **[Swedish]** discussed in Section 3.1 of this chapter. In particular, from the premises **[Swedish1]**, **[Swedish2]**, and **[man]**, we correctly derive the meaning constructor for *Swedish man* given in (20) of this chapter:

(36) **[Swedish1], [Swedish2], [man]** $\vdash \lambda X.\mathit{Swedish}(X) \wedge \mathit{man}(X) : v \multimap r$

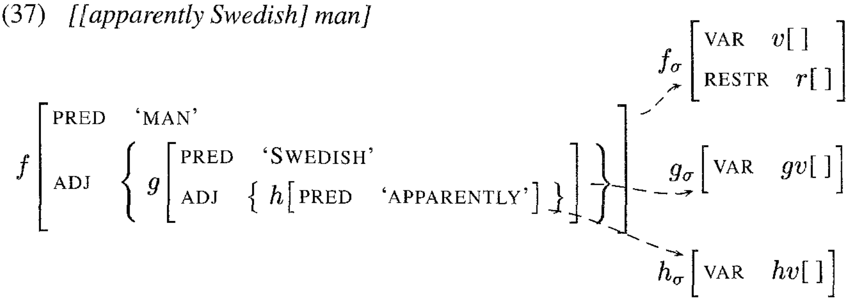
We treat other adjectival modifiers similarly: each adjective makes a twofold semantic contribution from which the simpler meaning constructors presented earlier can be deduced.

More generally, the example just presented illustrates that the simple and intuitive assumptions we make about meanings and how they combine often turn out to be largely correct, but in need of refinement to account for more complicated examples. In logical terms, the intuitively motivated meaning constructors often correspond to conclusions resulting from a deduction from a more refined set of basic meaning constructor premises. It is often easier to work with the simpler and more intuitive constructors; this is legitimate and theoretically sound as long as they follow as a logical consequence from the more basic premises.

4.2. Modification of Modifiers

We now demonstrate the derivation of the meaning for *apparently Swedish man*, an example in which the modifier *Swedish* is itself modified. As above, we introduce a VAR attribute with value hv in the semantic structure h_σ corresponding to *apparently*:

(37) *[[apparently Swedish] man]*



$\lambda X.\mathit{apparently}(\mathit{Swedish}(X)) \wedge \mathit{man}(X) : v \multimap r$

The lexical entry for *apparently* is given in (38):

(38) *apparently* $(\uparrow \text{PRED}) = \text{‘APPARENTLY’}$

$$\lambda P.\mathit{apparently}(P) : [(\uparrow_\sigma\ \text{VAR}) \multimap \uparrow_\sigma]$$

$$\begin{array}{l}\lambda Q.\lambda R.\lambda X.Q(R(X)) :\\ \quad [(\uparrow_\sigma\ \text{VAR}) \multimap \uparrow_\sigma] \multimap\\ \qquad [[((\text{ADJ} \in \uparrow)_\sigma\ \text{VAR}) \multimap (\text{ADJ} \in \uparrow)_\sigma] \multimap\\ \qquad\quad [((\text{ADJ} \in \uparrow)_\sigma\ \text{VAR}) \multimap (\text{ADJ} \in \uparrow)_\sigma]]\end{array}$$

Again, readability increases when the entries are instantiated according to the labels in (37):

(39) Meaning constructor premises for *apparently*:

[apparently1] $\lambda P.\mathit{apparently}(P) : [hv \multimap h_\sigma]$

[apparently2] $\lambda Q.\lambda R.\lambda X.Q(R(X)) : [hv \multimap h_\sigma] \multimap [[gv \multimap g_\sigma] \multimap [gv \multimap g_\sigma]]$

As with the lexical entry for *Swedish* given in (33) of this chapter, the meaning contribution of *apparently* is twofold. The first meaning constructor in the lexical entry in (38), labeled **[apparently1]** in (39), specifies the intrinsic meaning of *apparently*, and the second meaning constructor **[apparently2]** indicates how this intrinsic meaning combines with the meaning of the phrase it modifies.

The two meaning constructors **[apparently1]** and **[apparently2]** combine to produce the meaning constructor labeled **[apparently]** in (40):

(40) **[apparently]** $\lambda R.\lambda X.\mathit{apparently}(R(X)) : [gv \multimap g_\sigma] \multimap [gv \multimap g_\sigma]$

This meaning constructor consumes a meaning resource of the form $gv \multimap g_\sigma$, producing a new meaning resource of the same form but corresponding to a modified meaning.

We can now combine the meaning constructor **[apparently]** with the meaning constructor **[Swedish1]** in (34) to yield the meaning constructor in (41), labeled **[apparently-Swedish]**:

(41) **[apparently-Swedish]** $\lambda X.\mathit{apparently}(\mathit{Swedish}(X)) : [gv \multimap g_\sigma]$

Notably, the right-hand side of this meaning constructor is the same as the right-hand side of the unmodified meaning constructor **[Swedish1]** and plays the same role in meaning composition:

(42) **[Swedish1]** $\lambda X.\mathit{Swedish}(X) : [gv \multimap g_\sigma]$

Next, we combine the meaning constructors **[apparently-Swedish]**, **[Swedish2]**, and **[man]** to produce the meaning constructor given in (37) above for *apparently Swedish man*, the correct result:

(43) **[apparently-Swedish], [Swedish2], [man]** $\vdash$

$$\lambda X.\mathit{apparently}(\mathit{Swedish}(X)) \wedge \mathit{man}(X) : v \multimap r$$

Thus, our refined theory of the semantic contribution of modifiers enables the clean and intuitive treatment of modification presented in Section 3 of this chapter. However, it also allows an analysis of recursive modification, which, as Kasper (1995) shows, has proven problematic in many other approaches.

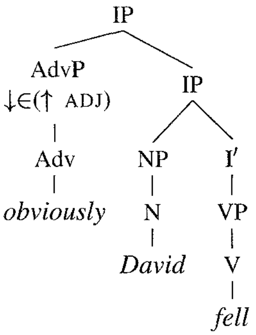
5. ADVERBIAL MODIFICATION

We now turn to an examination of the syntax and semantics of adverbial modification. The treatment provided here is brief, and we concentrate primarily on aspects of meaning composition; Butt et al. (1999, Chapter 7) provide more discussion of the syntax of adverbial modifiers from an LFG perspective.

5.1. Adverbs at C-Structure and F-Structure

In English, adverbs such as *obviously* and *skillfully* are adjoined to the phrases they modify. Like other modifiers, their f-structures appear as members of the set of ADJ modifiers. In (44), the sentential adverb *obviously* is adjoined to IP:

(44) *Obviously David fell.*

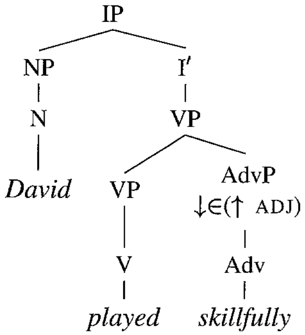


$$\left[\begin{array}{ll} \text{PRED} & \text{`FALL}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & \left[\text{PRED} \quad \text{`DAVID'}\right] \\ \text{ADJ} & \left\{ \left[\text{PRED} \quad \text{`OBVIOUSLY'}\right] \right\} \end{array}\right]$$

A manner adverb like *skillfully* can be adjoined to VP, as in example (46) (page 270). Evidence that the adverb *skillfully* is adjoined to VP in this example comes from the VP preposing construction, discussed in Chapter 3, Section 5, where a VP appears in fronted position. If the VP includes an adverb, it is also preposed, showing that the adverb forms a constituent with the VP.

(45) *David wants to play skillfully, and [play skillfully] he will.*

(46) *David played skillfully.*



$$\left[\begin{array}{ll} \text{PRED} & \text{‘PLAY}\langle\text{SUBJ}\rangle\text{’} \\ \text{SUBJ} & \left[\text{PRED} \quad \text{‘DAVID’}\right] \\ \text{ADJ} & \left\{ \left[\text{PRED} \quad \text{‘SKILLFULLY’}\right] \right\} \end{array}\right]$$

5.2. Adverbs and Semantic Composition

5.2.1. Adverb Meaning

The semantic contribution of adverbs has long been a focus of generative linguistic research. Heny (1973) gives a cogent overview of the state of research on adverb meaning in the early 1970s, when much research on adverb meaning was done; though it was conducted on the basis of very different syntactic assumptions, this work nevertheless forms the foundation upon which much current work on the semantics of adverbs is based. We will examine two semantically different kinds of adverbs, illustrated in the previous section by the sentential adverb *obviously* and the manner adverb *skillfully*.

Within the LFG semantic tradition, Halvorsen (1983) discusses sentential adverbs like *obviously* and *necessarily* and proposes to treat them in the standard way, as proposition modifiers. The meaning of the sentence *Obviously David fell* is given in (47):

(47) *Obviously David fell.*

obviously(fall(David))

The predicate *obviously* takes as its argument the proposition *David fell*, and the meaning represented in (47) is roughly paraphrasable as *It is obvious that David fell.*

Heny (1973), writing at the time at which Nixon was the president of the United States, considers the following pair of sentences:

(48) a. *The U.S. president is necessarily a citizen of the United States.*

b. *Nixon is necessarily a citizen of the United States.*

As Heny notes, sentence (48a) is true under the rules of the constitution of the United States, while sentence (48b) is not necessarily true. In other words, though it turns out to be true that Nixon is a citizen of the United States, this is not necessarily the case, since Nixon could have decided to become a citizen of another country. On the other hand, it is necessarily the case that the U.S. president must be a U.S. citizen under the laws of the United States. The sentences in (49), containing the sentential adverb *obviously*, differ from one another in a similar way:

(49) a. *Obviously the running back fell.*

b. *Obviously David fell.*

Even in a situation where the running back fell and David is the running back, it may not be obvious that David fell, since the identity of the running back may not be clear. Adverbs like *obviously* and *necessarily* are *opaque* in their subject position, since different ways of referring to the same individual can affect the truth or falsity of the sentence (Quine 1953).

This aspect of the meaning of sentential adverbs is different from manner adverbs. Intuitively, a manner adverb like *skillfully* modifies the action that is performed, producing a new action that is performed skillfully:

(50) *David played skillfully.*

$\mathit{skillfully}(\mathit{David}, \lambda X.\mathit{play}(X))$

In (50), *skillfully* is a two-place predicate: its arguments are the person that performed the action (here, David) and the action that is performed skillfully (here, playing). In general, a manner adverb like *skillfully* takes two arguments, one corresponding to the subject of the sentence and the other roughly corresponding to the verb phrase — the action that is performed. For this reason, such adverbs are sometimes called *VP* or *verb phrase adverbs*. As we will see, however, meaning combination with adverbs like *skillfully* depends on f-structural relations like SUBJ, not c-structure constituency relations.

Unlike the situation with sentential adverbs, the following two sentences are both true if David is the running back and he played skillfully. Manner adverbs like *skillfully* are not opaque in their subject position, so that if David is the running back, the sentences in (51) are true in the same circumstances:

(51) a. *David played skillfully.*

b. *The running back played skillfully.*

5.2.2. Adverbs and Meaning Assembly

We assume the syntactic and semantic structures and meaning constructor in (52) for the sentence *Obviously David fell*. The f-structure for *obviously* is labeled

h, and its semantic structure h_σ contains the attribute VAR whose value we have labeled hv:

(52) *Obviously David fell.*

$$f\left[\begin{array}{ll}\text{PRED} & \text{`FALL}\langle\text{SUBJ}\rangle\text{'}\\ \text{SUBJ} & g\left[\text{PRED}\ \ \text{`DAVID'}\right]\\ \text{ADJ} & \left\{\, h\left[\text{PRED}\ \ \text{`OBVIOUSLY'}\right] \right\}\end{array}\right] \dashrightarrow h_\sigma\left[\text{VAR}\ \ hv[\ ]\right]$$

$$\mathit{obviously}(\mathit{fall}(\mathit{David})) : f_\sigma$$

From now on, we will simplify our representations by displaying only semantic structures whose internal structure is of interest in the constructions we are considering. Therefore, we do not display the semantic structures f_σ or g_σ corresponding to the sentence f-structure f and the subject f-structure g.

We propose the lexical entry in (53) for the sentential adverb *obviously*:

(53) *obviously* $(\uparrow \text{PRED}) = \text{`OBVIOUSLY'}$

$$\lambda P.\mathit{obviously}(P) : (\uparrow_\sigma \text{VAR}) \multimap \uparrow_\sigma$$

$$\lambda P.\lambda Q.P(Q) :$$
$$[(\uparrow_\sigma \text{VAR}) \multimap \uparrow_\sigma] \multimap [(\text{ADJ} \in \uparrow)_\sigma \multimap (\text{ADJ} \in \uparrow)_\sigma]$$

As in the previous sections, the lexical entry in (53) uses inside-out functional uncertainty to refer to the f-structure of the phrase it modifies. The expression $(\text{ADJ} \in \uparrow)$ refers to the f-structure modified by *obviously*.

The instantiated meaning constructors for the sentence *Obviously David fell* are given in (54): the meaning constructors contributed by *obviously* are labeled **[obviously1]** and **[obviously2]**, and the meaning constructors **[David]** and **[fall]** follow the proposals for proper names and intransitive verbs given in Chapter 9.

(54) Meaning constructor premises for *Obviously David fell*:

$$\begin{array}{lrcl}\textbf{[David]} & \mathit{David} & : & g_\sigma\\ \textbf{[fall]} & \lambda X.\mathit{fall}(X) & : & g_\sigma \multimap f_\sigma\\ \textbf{[obviously1]} & \lambda P.\mathit{obviously}(P) & : & hv \multimap h_\sigma\\ \textbf{[obviously2]} & \lambda P.\lambda Q.P(Q) & : & [hv \multimap h_\sigma] \multimap [f_\sigma \multimap f_\sigma]\end{array}$$

Since the modifying adverb *obviously* is not itself modified, we first combine the two meaning constructor premises **[obviously1]** and **[obviously2]** to obtain the meaning constructor **[obviously]** given in (55):

(55) **[obviously]** $\lambda Q.\mathit{obviously}(Q) : [f_\sigma \multimap f_\sigma]$

As described in Chapter 9, Section 5.2.1, we can combine the premises labeled **[David]** and **[fall]** to obtain the meaning constructor labeled **[David-fall]** in (56):

(56) **[David-fall]** $\textit{fall}(\textit{David}) : f_\sigma$

Finally, we combine the meaning constructors **[David-fall]** and **[obviously]** to obtain the desired result, that the meaning of the sentence is $\textit{obviously}(\textit{fall}(\textit{David}))$:

(57) **[David-fall], [obviously]** $\vdash$ $\textit{obviously}(\textit{fall}(\textit{David})) : f_\sigma$

The derivation is semantically complete and coherent: we have obtained a well-formed, nonimplicational meaning constructor for the sentence, with no premises left unused.

The meaning deduction of a sentence with the manner adverb *skillfully* proceeds somewhat differently. The syntactic and semantic structures and meaning constructor for the sentence *David played skillfully* are given in (58), where the semantic structure h_σ corresponding to the adverb f-structure has the attribute VAR with value v and PROP with value p:

(58) *David played skillfully.*

$$f\begin{bmatrix} \text{PRED} & \text{'PLAY}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & g\begin{bmatrix}\text{PRED} & \text{'DAVID'}\end{bmatrix} \\ \text{ADJ} & \left\{ h\begin{bmatrix}\text{PRED} & \text{'SKILLFULLY'}\end{bmatrix} \right\} \end{bmatrix} \dashrightarrow h_\sigma\begin{bmatrix} \text{VAR} & v[\,] \\ \text{PROP} & p[\,] \end{bmatrix}$$

$\textit{skillfully}(\textit{David}, \lambda X.\textit{play}(X)) : f_\sigma$

Again, we assume a bipartite semantic contribution for the adverb *skillfully*. The lexical entry for *skillfully* is given in (59), and the instantiated meaning constructor premises for this sentence are given in (60).

(59) *skillfully* $(\uparrow \text{PRED}) = \text{'SKILLFULLY'}$

$$\begin{aligned} &\lambda P.\lambda X.\textit{skillfully}(X, P) \;: \\ &\quad [(\uparrow_\sigma \text{VAR}) \multimap (\uparrow_\sigma \text{PROP})] \multimap [(\uparrow_\sigma \text{VAR}) \multimap \uparrow_\sigma] \\ &\lambda P.\lambda Q.P(Q) \;: \\ &\quad [[(\uparrow_\sigma \text{VAR}) \multimap (\uparrow_\sigma \text{PROP})] \multimap [(\uparrow_\sigma \text{VAR}) \multimap \uparrow_\sigma]] \multimap \\ &\qquad\quad [((\text{ADJ} \in \uparrow)\ \text{SUBJ})_\sigma \multimap (\text{ADJ} \in \uparrow)_\sigma] \multimap \\ &\qquad\qquad [((\text{ADJ} \in \uparrow)\ \text{SUBJ})_\sigma \multimap (\text{ADJ} \in \uparrow)_\sigma] \end{aligned}$$

(60) Meaning constructor premises for *David played skillfully*:

[David]	$\textit{David}$	$: g_\sigma$
[play]	$\lambda X.\textit{play}(X)$	$: g_\sigma \multimap f_\sigma$
[skillfully1]	$\lambda P.\lambda Y.\textit{skillfully}(Y, P)$	$: [v \multimap p] \multimap [v \multimap h_\sigma]$
[skillfully2]	$\lambda P.\lambda Q.\lambda Y.P(Y, Q)$	$: [[v \multimap p] \multimap [v \multimap h_\sigma]] \multimap [[g_\sigma \multimap f_\sigma] \multimap [g_\sigma \multimap f_\sigma]]$

We begin the derivation by combining the premises **[skillfully1]** and **[skillfully2]** to obtain the meaning constructor labeled **[skillfully]** in (61):

(61) **[skillfully]** $\lambda Q.\lambda Y.\mathit{skillfully}(Y, Q) : [[g_\sigma \multimap f_\sigma] \multimap [g_\sigma \multimap f_\sigma]]$

The right-hand side of the meaning contribution of the intransitive verb *play*, $g_\sigma \multimap f_\sigma$ exactly matches the requirements of **[skillfully]**. We combine **[skillfully]** and **[play]**, obtaining the meaning constructor labeled **[skillfully-play]** in (62):

(62) **[skillfully-play]** $\lambda Y.\mathit{skillfully}(Y, \lambda X.\mathit{play}(X)) : g_\sigma \multimap f_\sigma$

Finally, we combine **[skillfully-play]** and **[David]** to obtain a wellformed, semantically complete and coherent meaning constructor for the sentence:

(63) **[skillfully-play], [David]** $\vdash$ $\mathit{skillfully}(\mathit{David}, \lambda X.\mathit{play}(X)) : f_\sigma$

6. FURTHER READING AND RELATED ISSUES

There has been much work on modification within LFG, particularly on the syntax of modifiers and adjunction, that has not been discussed in this chapter. In particular, Butt et al. (1999) discuss the syntax of adjectives and adverbs in English, French, and German, and Colban (1987) provides a syntactic and semantic analysis of prepositional phrases as verbal arguments and modifiers.

We have also omitted definition and discussion of scoping relations between modifiers. As noted by Andrews (1983b), Pollard and Sag (1994), and many others, the contribution of modifiers to the meaning of an utterance can depend on the order in which they appear:

(64) a. *Kim jogged for twenty minutes twice a day.*

b. *Kim jogged twice a day for twenty years.*

Syntactically, modifier scope is defined in terms of *f-precedence* (Chapter 6, Section 4.4), and semantic scope relations are in turn constrained by the syntactic scope relations defined by f-precedence. Crouch and van Genabith (1999) provide a theory of scoping relations and how they can be imposed within the glue approach.

11

ANAPHORA

The formal theory of LFG predicts a variety of types of anaphora, and research has shown that the full variety predicted by the theory is attested: Section 1 of this chapter shows that incorporated pronominal elements behave differently from elements that alternate with agreement markers, and these differ from morphologically independent pronouns in interesting ways. Anaphoric relations and binding patterns have also been fairly well studied; Section 2 discusses constraints on anaphoric binding stated in terms of f-structure relations as well as properties of other linguistic levels. Our glue-theoretic treatment of the semantics of anaphoric binding is presented in Section 3. This semantic treatment will be useful in subsequent chapters, particularly in our discussion of *anaphoric control* in Chapter 12.

1. INCORPORATED PRONOUNS

As discussed in Chapter 5, Section 4.3, a predicate may specify information about how its arguments are interpreted when no overt argument phrases are present. In Chicheŵa, for example, a verb like *zi-ná-wá-lum-a* 'bite' provides information about its subject and its object. The OBJ affix *-wá-* is unambiguously an incorporated OBJ pronoun, so that a better gloss for this form might be 'bite them'. This incorporated pronominal OBJ may be anaphorically linked to a TOPIC phrase, as in the English example *Those students, the bees bit them.*

In contrast, the SUBJ marker *zi-* behaves either as an agreement marker or as an incorporated pronoun. In the presence of an overt SUBJ phrase, *zi-* simply marks agreement with the subject. Alternately, when no overt SUBJ phrase appears, *zi-* is an incorporated pronoun like the OBJ marker. Since the subject marker may behave as an incorporated pronoun, it can be anaphorically linked to a TOPIC phrase, like the incorporated OBJ pronoun *-wá-*; the important difference between the two markers is that the SUBJ marker has an alternate use as an agreement marker in addition to its use as an incorporated pronominal.

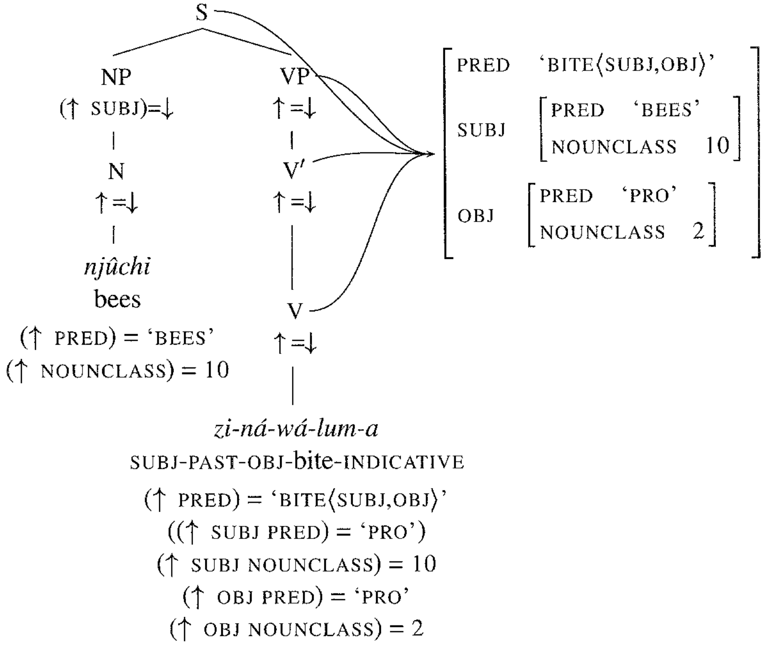
As Bresnan and Mchombo (1987) show, there is a great deal of evidence for the different status of the SUBJ and OBJ markers in Chicheŵa. For example, the subject marker can appear as an agreement marker with an idiomatic subject, but the object marker cannot appear with an idiomatic object, since an idiomatic object cannot be interpreted as a TOPIC and cannot bear an anaphoric relation to a incorporated pronominal object. Further, the subject can be questioned if a subject marker is present, but the object cannot be questioned if the object marker is present: since the question word bears the FOCUS function, it is compatible with the subject agreement marker, but not with the pronominal OBJ, which must bear an anaphoric relation to a TOPIC and not a FOCUS. Bresnan and Mchombo (1987) enumerate additional ways in which the SUBJ and OBJ markers behave differently, showing that all of these differences can be explained on the basis of the different status of the two markers.

The difference between the SUBJ and OBJ affixes is formally reflected in the following lexical entry:

(1) *zi-ná-wá-lum-a* (↑ PRED) = 'BITE⟨SUBJ,OBJ⟩'
((↑ SUBJ PRED) = 'PRO')
(↑ SUBJ NOUNCLASS) = 10
(↑ OBJ PRED) = 'PRO'
(↑ OBJ NOUNCLASS) = 2

In this lexical entry, the PRED value of the OBJ of *zi-ná-wá-lum-a* 'bite' is unambiguously specified: the object of this verb is pronominal. In contrast, the PRED value of the SUBJ is optionally specified, as denoted by the parentheses. In the presence of an overt SUBJ phrase, the subject marker specifies only agreement information, and the PRED value of the SUBJ is provided by the overt subject phrase, as shown in example (2). In contrast, example (3) (page 278) shows that when there is no subject phrase, the specifications associated with the verb provide the PRED value for the SUBJ.

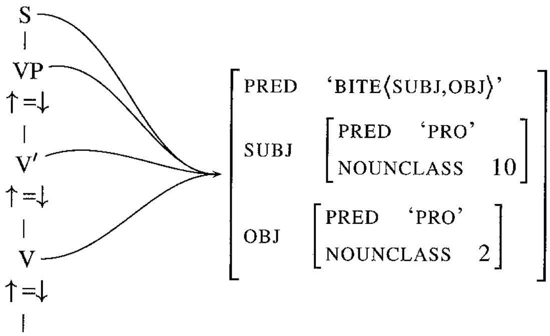
(2) *njûchi zi-ná-wá-lum-a*
bees SUBJ-PAST-OBJ-bite-INDICATIVE
'The bees bit them.'



As discussed in Chapter 5, Section 4, the typology of agreement and pronominal incorporation that is reflected in these different specifications is richer than is assumed in some other theories. In her analysis of nonconfigurationality, Jelinek (1984) proposes that all nonconfigurational languages should be analyzed as pronominal-incorporating, as we have analyzed the Chicheŵa incorporated object pronoun. Bresnan and Mchombo (1987) conclusively demonstrate that this inflexible approach is incorrect and that a wider range of distinctions is necessary. The object marking on the Chicheŵa verb must be analyzed as an incorporated pronoun with an obligatory pronominal PRED value supplied by the verb, while the subject marking represents an optional pronominal PRED feature, behaving as an agreement marker in the presence of an overt subject phrase. Further evidence for the necessity of the richer pronominal typology assumed in LFG is provided by Austin and Bresnan (1996) in their analysis of Warlpiri, a language that is nonconfigurational by Jelinek's criteria.

Bresnan (2001b) provides much more discussion of the typology of pronominal elements in LFG, including a detailed discussion of the differences between overt pronouns and "null" pronominals like the Chicheŵa incorporated OBJ pro-

(3) *zi-ná-wá-lum-a*
SUBJ-PAST-OBJ-bite-INDICATIVE
'They bit them.'



zi-ná-wá-lum-a
SUBJ-PAST-OBJ-bite-INDICATIVE
(↑ PRED) = 'BITE⟨SUBJ,OBJ⟩'
((↑ SUBJ PRED) = 'PRO')
(↑ SUBJ NOUNCLASS) = 10
(↑ OBJ PRED) = 'PRO'
(↑ OBJ NOUNCLASS) = 2

noun. In particular, Bresnan (2001b, Chapter 8) explores the role of null pronominals in introducing and referring to sentential and discourse topics and in focus constructions. These constraints are defined at the level of *information structure* (Chapter 7, Section 3). In the following, we will not provide a formal analysis of information-structural constraints and how they help to determine the space of possible referents for a pronoun: for discussion of these issues and an overview of theoretical approaches, see Kehler (2000).

2. BINDING RELATIONS

Anaphoric binding relations are semantic in nature, having to do with coreference between a pronoun and its antecedent. Nonsemantic levels of linguistic structure also play a role in anaphoric binding, however, since they are often important in constraining possible binding relations.

We first discuss constraints on the binding relation which are defined at f-structure. In this discussion, we will assume the phrase structure rules and representations presented earlier, and we will display only the f-structures for the examples under discussion. In Section 2.2, we briefly discuss constraints on anaphor-antecedent relations stated in terms of linear order and functional precedence, and in Section 2.3, we discuss binding conditions related to argument structure and the thematic hierarchy.

2.1. F-Structural Constraints

Within LFG, Bresnan et al. (1985a) were the first to explore binding constraints defined in terms of f-structural relations, proposing that a theory of syntactic constraints on anaphoric coreference relations can be stated in terms of f-structural properties, such as coargumenthood or the presence of a SUBJ function. Continuing this work, Dalrymple (1993) proposed a universally available and lexically specified inventory of binding constraints, and also provided a formal specification of these constraints. This work has been continued and extended by Strand (1992) in work on Norwegian anaphora, Sung (1996) in work on Korean, Lapata (1998) in work on Greek, and Henadeerage (1998) in work on Sinhala. Bresnan (2001b, Chapter 10) provides a detailed discussion of anaphoric binding constraints, extending the theory to cover coreference relations between nonpronominal elements as well. In the following, we give a brief overview of the theory.

2.1.1. Positive Binding Constraints

Some anaphoric elements, such as the English reflexive pronoun *himself*, must appear in a particular syntactic relation to their antecedent. We say that elements like *himself* obey *positive constraints* — that is, constraints that state the syntactic relation that an anaphor must bear to its antecedent.

In example (4), the antecedent of *himself* is the SUBJ, *David*, as indicated by the subscript i annotation coindexing *David* and *himself*:

(4) *David*$_i$ *compared Chris to himself*$_i$.

The f-structure for example (4) is:

(5)
$$f\left[\begin{array}{ll} \mathrm{PRED} & \text{`COMPARE}\langle\mathrm{SUBJ,OBJ,OBL_{GOAL}}\rangle\text{'} \\ \mathrm{SUBJ} & \left[\mathrm{PRED}\quad \text{`DAVID'}\right] \\ \mathrm{OBJ} & \left[\mathrm{PRED}\quad \text{`CHRIS'}\right] \\ \mathrm{OBL_{GOAL}} & \left[\begin{array}{ll} \mathrm{PRED} & \text{`PRO'} \\ \mathrm{PRONTYPE} & \mathrm{REFL} \end{array}\right] \end{array}\right]$$

In this f-structure, the antecedent *David* of the reflexive pronoun *himself* is the SUBJ of the f-structure labeled f, and the reflexive pronoun is the OBL_{GOAL} of the same f-structure. The semantic antecedency relation, which establishes coreference between the anaphor and its antecedent, is defined and discussed in Section 3.4.1 of this chapter; here we concentrate on syntactic factors that constrain the anaphor-antecedent relation.

The antecedent of the pronominal *himself* may also appear in an f-structure that does not contain the pronoun:

(6) *David$_i$ wrapped the blanket around himself$_i$.*

$$\begin{bmatrix} \text{PRED} & \text{`WRAP}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`DAVID'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{`BLANKET'} \end{bmatrix} \\ \text{ADJ} & \left\{ \begin{bmatrix} \text{PRED} & \text{`AROUND}\langle\text{OBJ}\rangle\text{'} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \\ \text{PRONTYPE} & \text{REFL} \end{bmatrix} \end{bmatrix} \right\} \end{bmatrix}$$

However, it is not possible for *himself* to appear in a sentence with no antecedent, or with an antecedent in a syntactically unacceptable relation to it. For example, *himself* may not be separated from its antecedent by a finite clause boundary:

(7) **David$_i$ thought that Chris had seen himself$_i$.*

$$\begin{bmatrix} \text{PRED} & \text{`THINK}\langle\text{SUBJ,COMP}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`DAVID'} \end{bmatrix} \\ \text{COMP} & \begin{bmatrix} \text{PRED} & \text{`SEE}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`CHRIS'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \\ \text{PRONTYPE} & \text{REFL} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

As demonstrated by Bresnan et al. (1985a), the English reflexive pronoun *himself* obeys the following positive constraint, constraining the syntactic relation between *himself* and its antecedent:

(8) The antecedent of the English reflexive pronoun *himself* must appear in the *Minimal Complete Nucleus* containing the pronoun.

The Minimal Complete Nucleus is defined by reference to the presence of a SUBJ function (Bresnan et al. 1985a; Dalrymple 1993; Bresnan 2001b):

(9) Minimal Complete Nucleus containing an f-structure f:

The smallest f-structure that contains f and a SUBJ function.

According to this definition, the antecedent of the anaphor *himself* must appear in the smallest f-structure that contains both the anaphor and a SUBJ. We call the domain in which the antecedent of the anaphor must appear the *binding domain* of the anaphor, so that we will say that the binding domain of *himself* is the Minimal Complete Nucleus.

Languages with multiple anaphors provide evidence for expanding the range of constraints that anaphors can obey and also demonstrate that constraints on anaphoric binding must be specified lexically, not universally or on a per-language basis: different anaphoric elements in the same language may obey different anaphoric binding constraints. The pronominal system of Norwegian is particularly rich.

Hellan (1988) shows that although the Norwegian reflexive anaphor *seg selv* and the reciprocal *hverandre* must be locally bound, the binding domain for the reciprocal *hverandre* is larger than the domain for *seg selv*. The reflexive *seg selv* must be bound to a *coargument*, an argument governed by the same PRED as the reflexive. In contrast, the reciprocal *hverandre*, like the English reflexive pronoun *himself*, must be bound in the Minimal Complete Nucleus. In example (10), the antecedent of the OBL$_{\text{GOAL}}$ *seg selv* is the SUBJ, *Jon* (Hellan 1988, p. 67):

(10) *Jon fortalte meg om seg selv.*
Jon told me about self
'Jon$_i$ told me about self$_i$.'

The f-structure for this example is:

(11)
$$\left[\begin{array}{ll} \text{PRED} & \text{`TELL}\langle\text{SUBJ,OBJ,OBL}_{\text{ABOUT}}\rangle\text{'} \\ \text{SUBJ} & \left[\text{PRED}\quad \text{`JON'}\right] \\ \text{OBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`PRO'}\\ \text{PERS} & 1 \\ \text{NUM} & \text{SG}\end{array}\right] \\ \text{OBL}_{\text{ABOUT}} & \left[\begin{array}{ll} \text{PRED} & \text{`PRO'} \\ \text{PRONTYPE} & \text{REFL}\end{array}\right]\end{array}\right]$$

The antecedent of *seg selv* is a coargument, as required: both *seg selv* and *Jon* are arguments of the verb *fortalte* 'tell'. Example (12) involves the reciprocal pronoun *hverandre*, and the antecedent is the coargument SUBJ *de* 'they' (Hellan 1988, p. 67). The f-structure for (12) is roughly the same as the one displayed in (11):

(12) *De fortalte meg om hverandre.*
they told me about each other
'They$_i$ told me about each other$_i$.'

However, the antecedent of the reciprocal *hverandre* is not required to be a coargument. The following examples show that a noncoargument antecedent is acceptable for the reciprocal *hverandre*, but not for the reflexive *seg selv* (Hellan 1988, p. 69):

(13) **Hun kastet meg fra seg selv.*
she threw me from herself
'She$_i$ threw me away from herself$_i$.'

In example (13), the reflexive pronoun *seg selv* is the OBJ of the preposition *fra* 'from', and the intended antecedent is the SUBJ of the verb *kastet* 'throw':

(14) **Hun kastet meg fra seg selv.*
she threw me from herself
'She$_i$ threw me away from herself$_i$.'

$$\begin{bmatrix} \text{PRED} & \text{`THROW}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`PRO'}\\ \text{PERS} & 3 \\ \text{NUM} & \text{SG}\end{bmatrix} \\ \text{OBJ} & \begin{bmatrix}\text{PRED} & \text{`PRO'}\\ \text{PERS} & 1\\ \text{NUM} & \text{SG}\end{bmatrix} \\ \text{ADJ} & \left\{\begin{bmatrix}\text{PRED} & \text{`FROM}\langle\text{OBJ}\rangle\text{'}\\ \text{OBJ} & \begin{bmatrix}\text{PRED} & \text{`PRO'}\\ \text{PRONTYPE} & \text{REFL}\end{bmatrix}\end{bmatrix}\right\} \end{bmatrix}$$

The reflexive and its intended antecedent are not coarguments of the same PRED, and the sentence is unacceptable.

In contrast, a similar example with the reciprocal *hverandre* is completely acceptable, since the reciprocal appears in the Minimal Complete Nucleus with its antecedent. The f-structure for (15) is roughly the same as the one in (14):

(15) *De kastet meg til og fra hverandre.*
they threw me to and from each other
'They$_i$ threw me to and from each other$_i$.'

These examples show that different anaphors in the same language may be required to obey different binding constraints. Thus, these constraints must be lexically specified for each anaphoric element. See Hellan (1988), Strand (1992), and Dalrymple (1993) for more discussion of binding constraints on Norwegian anaphors.

Crosslinguistic examination of anaphoric binding patterns reveals four domains relevant for anaphoric binding, defined by the f-structure properties SUBJ, TENSE, and PRED and by the entire utterance:

(16) Coargument Domain: minimal domain defined by a PRED and the grammatical functions it governs

Minimal Complete Nucleus: minimal domain with a SUBJ function

Minimal Finite Domain: minimal domain with a TENSE feature

Root Domain: f-structure of the entire utterance

Interestingly, all of these domains denote some syntactically or semantically complete entity: the Coargument Domain corresponds to a syntactically saturated argument structure; the Minimal Complete Nucleus corresponds to a predication involving some property and the subject; the Minimal Finite Domain represents an event that has been spatiotemporally anchored; and the Root Domain represents a complete utterance. The binding conditions defined in (16) are illustrated by the binding requirements for the Norwegian anaphors *seg selv*, *sin*, and *seg* (Hellan 1988) and the Chinese anaphor *ziji* (Tang 1989); see Dalrymple (1993) for more discussion:

(17) Positive binding domains:

Coargument domain	Min. complete nucleus	Min. finite domain	Root domain
seg selv	*sin*	*seg*	*ziji*

These binding requirements are specified as part of the syntactic constraints given in the lexical entry of each anaphoric element (Dalrymple 1993). These syntactic requirements limit the possibilities for pronoun antecedency, which, as we will see in Section 2.2 and Section 2.3, may be further constrained by information at other levels.

Formally, we can define the syntactic domain in which an anaphor must find its antecedent by means of expressions involving *inside-out functional uncertainty*. As described in Chapter 6, Section 1.2, inside-out functional uncertainty allows reference to enclosing structures, those in which a particular f-structure is contained. In the case at hand, we can use inside-out functional uncertainty to define the binding domain relative to an anaphor, the f-structure domain within which the anaphor is required to find its antecedent.

Assuming that the f-structure for the pronoun is f, we can define each of the anaphoric binding domains as in (18):

(18) Coargument Domain: $(\underset{\neg(\rightarrow\ \mathrm{PRED})}{\mathrm{GF}^*}\quad \mathrm{GF}_{\mathrm{pro}}\ f)$

Minimal Complete Nucleus: $(\underset{\neg(\rightarrow\ \mathrm{SUBJ})}{\mathrm{GF}^*}\quad \mathrm{GF}_{\mathrm{pro}}\ f)$

Minimal Finite Domain: $(\underset{\neg(\rightarrow\ \mathrm{TENSE})}{\mathrm{GF}^*}\quad \mathrm{GF}_{\mathrm{pro}}\ f)$

Root Domain: $(\mathrm{GF}^*\ \mathrm{GF}_{\mathrm{pro}}\ f)$

These expressions constrain the f-structure domain within which the antecedent of the anaphor can appear. Recall that GF is an abbreviation for any grammatical function (Chapter 6, Section 1.1). In the expressions in (18), the path leading to the anaphor through the binding domain is:

(19) $\mathrm{GF}^*\ \mathrm{GF}_{\mathrm{pro}}$

For clarity, we use the abbreviation $\mathrm{GF}_{\mathrm{pro}}$ for the grammatical function borne by the pronoun, which can be any grammatical function. Constraints on the domain within which the anaphor must be bound are stated by means of *off-path constraints* on the path to the anaphor; off-path constraints are defined and discussed in Chapter 6, Section 1.4.

In the case of the Coargument Domain definition, for example, the path leading to the anaphor is:

(20) $\underset{\neg(\rightarrow\ \mathrm{PRED})}{\mathrm{GF}^*}\quad \mathrm{GF}_{\mathrm{pro}}$

This expression refers to a series of attributes GF, each of which must obey the off-path constraint $\neg(\rightarrow \mathrm{PRED})$. In this expression, the symbol $\rightarrow$ refers to the f-structure value of the attribute GF; the expression $\neg(\rightarrow \mathrm{PRED})$ represents a negative existential constraint preventing that f-structure from containing the attribute PRED. The effect of this constraint is that the path leading to the f-structure may not pass through an f-structure containing the attribute PRED. If the path passed through such an f-structure, the binding domain would incorrectly extend beyond the Coargument Domain.

The other binding domains are similarly constrained by off-path constraints that prevent the path from passing through an f-structure of a certain type. For anaphors subject to the Minimal Complete Nucleus constraint, the path from the pronoun to its antecedent may not pass through an f-structure with a SUBJ attribute, and for the Minimal Finite Domain constraint, an f-structure with a TENSE attribute may not be crossed. For the Root Domain, the path is unconstrained and may be of arbitrary length. In this way, anaphors that obey positive constraints are required to find an antecedent within a certain f-structural domain.

Although the antecedent of the anaphor must appear within its binding domain, antecedents that are too deeply embedded within the binding domain are not acceptable:

(21) **David$_i$'s father nominated himself$_i$.*

The f-structure for this example is:

(22)
$$\left[\begin{array}{ll} \text{PRED} & \text{`NOMINATE}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \left[\begin{array}{ll} \text{SPEC} & \left[\begin{array}{ll}\text{PRED} & \text{`DAVID'}\end{array}\right] \\ \text{PRED} & \text{`FATHER'}\end{array}\right] \\ \text{OBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`PRO'} \\ \text{PRONTYPE} & \text{REFL}\end{array}\right]\end{array}\right]$$

As noted in Chapter 6, Section 3.1, the antecedent of an anaphor is generally required to *f-command* the anaphor; that is, every f-structure that contains the antecedent must also contain the anaphor.[1] Although in example (21) the intended antecedent *David* of the reflexive *himself* appears in the proper binding domain, the Minimal Complete Nucleus, the f-command condition does not hold, and the sentence is ill-formed.

The syntactic relation between the anaphor and its antecedent is given by a constraint of the following form, where GF_{ante} is the grammatical function of the antecedent:

(23) $((\text{GF}^* \ \text{GF}_{\text{pro}} \ f) \ \text{GF}_{\text{ante}})$

In this expression, as above, f is the f-structure for the anaphor, and $(\text{GF}^* \ \text{GF}_{\text{pro}} \ f)$ defines the binding domain containing the anaphor and its antecedent. The expression in (23) refers to some f-structure bearing the unspecified grammatical function GF_{ante} within the binding domain. The f-command requirement follows from the form of this expression, since the expression in (23) picks out all and only the f-structures that f-command the anaphor within the binding domain.

In the expression in (23), the grammatical function of the antecedent GF_{ante} is unconstrained, and any grammatical function may be chosen. In some cases, however, the grammatical function of the antecedent may also be constrained in that the antecedent of the anaphor may be required to bear the SUBJ function within

[1] See Culy (1991) for discussion of the Fula pronominal *dum*, which is unusual in not obeying a command condition.

the binding domain. Hellan (1988) shows that the antecedent of the Norwegian possessive reflexive *sin* must be a subject:

(24) a. *Jon ble arrestert i sin kjøkkenhave.*
Jon was arrested in self's kitchen garden
'Jon$_i$ was arrested in his$_i$ kitchen garden.'

b. **Vi arresterte Jon i sin kjøkkenhave.*
we arrested Jon in self's kitchen garden
'We arrested Jon$_i$ in his$_i$ kitchen garden.'

In contrast, nonsubjects are acceptable antecedents for the Norwegian possessive pronoun *hans*:

(25) *Vi fant Jon under sengen hans.*
we found Jon under bed his
'We found Jon$_i$ under his$_i$ bed.'

Thus, it is necessary in some cases to constrain the grammatical function of the antecedent: it may be required to be a SUBJ. The following expression picks out all subjects within the Minimal Finite Domain relative to the pronoun f-structure *f*:

(26) $$((\underset{\neg(\rightarrow\ \text{TENSE})}{\text{GF}^{*}}\quad \text{GF}_{\text{pro}}\ f)\ \text{SUBJ})$$

F-structures with the SUBJ function in other domains are picked out similarly; see Dalrymple (1993) and Bresnan (2001b) for details.

Other conditions on the antecedent must also be met in some cases. Culy (1996) discusses the pronominal systems of several varieties of Fula in which certain pronouns place very specific syntactic requirements on their antecedents: for instance, that the antecedent must be a pronoun. Culy analyzes this as a type of agreement between the pronoun and its antecedent.

2.1.2. NEGATIVE BINDING CONSTRAINTS

Just as some anaphoric elements require their antecedent to appear within some syntactic domain, some elements can require noncoreference with every element within some domain. We call such noncoreference constraints *negative constraints*. For example, the pronoun *him* may not corefer with its coargument *Chris* in example (27):

(27) **Chris$_i$ nominated him$_i$.*

The f-structure for this sentence is:

(28)

$$\begin{bmatrix} \text{PRED} & \text{`NOMINATE}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`CHRIS'}\end{bmatrix} \\ \text{OBJ} & \begin{bmatrix}\text{PRED} & \text{`PRO'}\end{bmatrix} \end{bmatrix}$$

The constraint obeyed by *him* can be stated in the following way:

(29) The antecedent of the English pronoun *him* must not appear in the Coargument Domain containing the pronoun.

Different anaphoric elements in the same language may obey different negative constraints. Thus, as with the positive constraints, negative constraints must be lexically associated with each anaphoric element. Interestingly, the same domains that are relevant for defining positive constraints, described in Section 2.1.1 of this chapter, are also relevant for negative constraints, as Dalrymple (1993) shows for the Norwegian pronouns *ham selv* and *seg* (Hellan 1988), the Hindi possessive pronoun *uskaa* (Mohanan 1994), and the Yoruba pronoun *ó* (Pulleyblank 1986):

(30) Negative binding domains:

Coargument domain	Min. complete nucleus	Min. finite domain	Root domain
seg	*ham selv*	*uskaa*	*ó*

In the case of positive constraints, the anaphor is required to corefer with *some* element picked out by the constraint; in the case of negative constraints, the anaphor must be noncoreferent with *all* elements picked out by the constraint.

2.1.3. Positive and Negative Binding Constraints

Since positive and negative constraints are lexically associated with individual anaphors, we would expect to find anaphors that simultaneously obey both kinds of constraints. The Norwegian anaphoric element *ham selv* exemplifies this situation (Hellan 1988; Dalrymple 1993): *ham selv* must be bound to an argument in the Minimal Complete Nucleus (a positive constraint), but it is also required to be noncoreferent with a coargument SUBJ (a negative constraint).

In example (31a), the OBJ *Jon* and the oblique phrase *ham selv* are coarguments. Therefore, *ham selv* is coreferent with a coargument in example (31a); example (31b) shows that coreference with noncoarguments is also permitted:

(31) a. *Vi fortalte Jon om ham selv.*
we told Jon about self
'We told Jon$_i$ about self$_i$.'

b. *Jeg ga Jon en bok om ham selv.*
I gave Jon a book about self
'I gave Jon$_i$ a book about self$_i$.'

The antecedent of *ham selv* must appear in the Minimal Complete Nucleus containing it. This accounts for the unacceptability of example (32), since the intended antecedent *Jon* does not appear in the Minimal Complete Nucleus relative to *ham selv*:

(32) **Jeg lovet Jon å snakke om ham selv.*
I promised Jon to talk about self
'I promised Jon$_i$ to talk about self$_i$.'

However, *ham selv* may not corefer with a coargument subject, even one that is in its Minimal Complete Nucleus:

(33) **Jon snakker om ham selv.*
Jon talks about self
'Jon$_i$ talks about self$_i$.'

Thus, the Norwegian pronoun *ham selv* obeys two binding conditions: it must be noncoreferent with a SUBJ coargument, and it must be coreferent with an argument in the Minimal Complete Nucleus. Both of these requirements must be satisfied.

2.2. Binding and F-Precedence

In addition to constraints on anaphoric binding defined purely in terms of f-structure properties, there are constraints that are defined in terms of f-precedence relations holding between the anaphor and its antecedent. For example, Mohanan (1983) shows that overt pronouns in Malayalam cannot precede their antecedents:

(34) *[kuṭṭiyute ammaye] awan ṉuḷḷi*
child.GEN mother.ACC he pinched
'He$_i$ pinched the child$_i$'s mother.'

(35) **[awante ammaye] kuṭṭi ṉuḷḷi*
his mother.ACC child pinched
'The child$_i$ pinched his$_i$ mother.'

In (36), an overt pronoun is not acceptable if coreference with the matrix subject is intended:

(36) *[awan aanaye ṉuḷḷiyaṯinə s̄eeṣam] kuṭṭi uraŋŋi*
he elephant.ACC pinched after child slept
'The child$_i$ slept, after he$_{*i,j}$ pinched the elephant.'

However, pronominals that are not overtly realized at c-structure do not obey such ordering restrictions. The subordinate clause in example (37) contains a null pronoun that may corefer with the matrix subject *kuṭṭi* 'child':

(37) *[∅ aanaye ṉuḷḷiyat̲inə s̄eeṣam] kuṭṭi uraŋŋi*
elephant.ACC pinched after child slept
'The child$_i$ slept, after he$_{i,j}$ pinched the elephant.'

Kameyama (1985) examines similar data from Japanese, discussed in Chapter 6, Section 4.4, and proposes the following generalization, valid for Japanese, Malayalam, and many other languages:

(38) The antecedent of a pronoun must *f-precede* the pronoun.

As noted in Chapter 6, Section 4.4, constraining binding relations by f-precedence makes exactly the right predictions concerning overt and null elements.

Intuitively, an f-structure f f-precedes an f-structure g if the c-structure nodes corresponding to f precede the c-structure nodes corresponding to g. The formal definition of f-precedence provided by Kaplan and Zaenen (1989) is given in definition (103) of Chapter 6, Section 4.4, repeated here:

(39) F-precedence:

f *f-precedes* g ($f <_f g$) if and only if for all $n_1 \in \phi^{-1}(f)$ and for all $n_2 \in \phi^{-1}(g)$, n_1 c-precedes n_2.

This definition, together with the generalization in (38), predicts the patterns of acceptability for the Japanese and Malayalam examples examined above: null pronouns f-precede and are f-preceded by every element in the sentence, so no matter where the antecedent of a null pronoun appears in the sentence, the condition in (38) is satisfied. In contrast, overt pronouns are not permitted to f-precede their antecedents, accounting for the unacceptability of coreference in examples (35) and (36).

Bresnan (1995, 1998, 2001b) also discusses linear precedence conditions on anaphoric binding with particular attention to *weak crossover violations* in extraction, providing a different definition of f-precedence. We will discuss her findings in Chapter 14.

2.3. Binding and Argument Structure

As discussed in Chapter 2, Section 1.3, binding may also be constrained by argument structure relations: the antecedent of an anaphor may be required to outrank the anaphor or a phrase containing it on the *thematic hierarchy*. The thematic hierarchy presented in (21) of Chapter 8, Section 4.3 is repeated in (40):

(40) Thematic hierarchy:

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

Work by Sells (1988) shows that reference to the thematic hierarchy is necessary in an account of binding conditions in Albanian. A term argument in Albanian can antecede a term or oblique reflexive, while an oblique can only antecede another oblique. Among the term arguments, possible binding relations are constrained by the thematic hierarchy: if the antecedent and the anaphor are both terms, the antecedent must be higher on the thematic hierarchy than the anaphor.

Hellan (1988), Dalrymple and Zaenen (1991), and Dalrymple (1993) discuss Norwegian data that point to a similar conclusion. Hellan (1988) shows that some Norwegian verbs have two possibilities for passivization:

(41) a. *Vi overlot Jon pengene.*
we gave Jon money
'We gave Jon the money.'

b. *Jon ble overlatt pengene.*
Jon was given money
'Jon was given the money.'

c. *Pengene ble overlatt Jon.*
money was given Jon
'The money was given Jon.'

However, when the object contains a possessive reflexive whose antecedent is the subject of the passive verb, only one reading is possible (Hellan 1988, p. 160):

(42) *Barnet ble fratatt sine foreldre.*
child was taken self's parents
'The child$_i$ was deprived of self's$_i$ parents.'
* 'The child$_i$ was taken away from self's$_i$ parents.'

The only possible construal of this sentence is one where the subject *barnet* is the malefactive argument, and the object *sine foreldre* is the theme. Assuming that the malefactive and benefactive arguments occupy the same position on the thematic hierarchy, the malefactive argument outranks the theme argument. In the permissible reading of example (42), then, the antecedent *barnet* outranks the phrase containing the pronoun *sine foreldre* on the thematic hierarchy. The other reading, where the phrase containing the pronoun thematically outranks the antecedent, is not available.

Evidence of binding constraints defined at multiple levels of structure has also been explored by Dalrymple (1993) for English and Norwegian and by Arka and

Wechsler (1996) for Balinese. In particular, Arka and Wechsler show that constraints on Balinese binding relations depend on linear order, thematic prominence, and the term/oblique distinction.

3. ANAPHORA AND SEMANTIC COMPOSITION

We turn now to the central issue in the interpretation of anaphora: how the semantic relation is established between an anaphor and its antecedent. Within the glue semantics approach, several proposals have been made for the semantic treatment of anaphoric binding. The first proposal was made by Dalrymple et al. (1997b), whose approach correctly handles the interactions of anaphora and quantification within the sentence. However, that proposal only peripherally addresses issues that arise in analyzing intersentential anaphora, the interpretation of anaphors whose antecedents are not in the same sentence.

Subsequently, Crouch and van Genabith (1999) made the important observation that the glue approach is well-suited to handle the context-changing potential of sentences. They propose that the linear logic glue language not only manages the dynamics of meaning composition within a sentence, but also manages *context resources* and *context update*. In their analysis, Crouch and van Genabith adopt an *e-type* treatment of anaphoric binding (Evans 1980), where descriptions of entities relevant in the discourse are constructed in the course of the derivation of the meaning of an utterance and are used in anaphora resolution both within the sentence and in subsequent discourse.

Here, we adopt the insight of Crouch and van Genabith (1999) that the glue approach not only accounts for resource-sensitive aspects of meaning assembly within sentences, but also manages contextual contributions in discourse interpretation. In contrast to their approach, however, we provide a theory of coreference and anaphoric binding that is closer to Discourse Representation Theory (Kamp 1981; Kamp and Reyle 1993) and Dynamic Predicate Logic (Groenendijk and Stokhof 1991). As we explain in the following, we introduce discourse referents that are restricted in appropriate ways: these discourse referents can persist throughout the discourse and can participate in anaphor resolution. The analysis presented in this section is based on unpublished joint work by Martin van den Berg, Dick Crouch, John Lamping, and the author.[2]

[2] We are grateful to Jonas Kuhn for helpful discussion of these issues.

3.1. Anaphora in Context

It has long been clear that phrases like *Chris*, *someone*, or *a man* introduce new individuals into the context that can be referred to in later discourse. Karttunen (1976) was among the first to propose that indefinite noun phrases introduce a discourse referent representing some individual into the discourse context and that these discourse referents play an important role in anaphora resolution. Karttunen's basic idea was elaborated and refined in the work of Heim (1982) on File Change Semantics and by Kamp (1981) in his work on Discourse Representation Theory. In both of these theories, certain phrases, including indefinite noun phrases and names, introduce discourse referents into the discourse context. Pronouns in subsequent discourse can take these phrases as antecedents and so can be resolved to the discourse referents that correspond to them.

We assume that besides the meanings of utterances, the glue language deduction process manages a list of discourse referents corresponding roughly to the file cards of Heim (1982), the discourse referents of Kamp (1981) and Kamp and Reyle (1993), and the assignments to variables of Groenendijk and Stokhof (1991). These discourse referents are introduced by proper names and indefinite noun phrases.[3] Certain operators, such as negation and universal quantification, trap these discourse referents and do not allow them to contribute to the global context. Where the discourse referents are not trapped by such operators, they persist across the discourse and allow for the resolution of anaphors in subsequent sentences to a discourse referent introduced earlier in the discourse.

For instance, part of the semantic contribution of sentences like *David arrived*, *I met David yesterday*, or *I am going to give it to David* is to introduce a new discourse referent representing the individual David into the context. In the semantic representation of the short dialogue in (43), the pronoun in the second sentence picks up the discourse referent corresponding to the subject of the first sentence as its antecedent. Resolving the anaphor *he* in the second sentence to the subject *David* in the first sentence is permitted because it provides an antecedent for the pronoun and does not violate any syntactic or other constraints imposed by the pronoun:

(43) *David arrived. He yawned.*

$$arrive(David), yawn(David)$$

The same observations can be made for indefinite noun phrases like *a man*. The sentence *A man arrived* introduces a new discourse referent into the context, which can be used to resolve a pronoun in subsequent discourse. We assume the

[3] Of course, discourse referents can also be introduced by phrases of other types and can correspond to events, to plural individuals formed from the individuals relevant in the context, and so on; see Kamp and Reyle (1993) for discussion of these issues. Our approach can also manage these aspects of context update, but a full treatment is beyond the scope of this work.

meaning $\mathit{a}(X, \mathit{man}(X), \mathit{arrive}(X))$ for the sentence *A man arrived* (Chapter 9, Section 4.1.4). We further assume that this phrase introduces a discourse referent X into the context and that this discourse referent is available for anaphor resolution in subsequent discourse. The representation for the second sentence *He yawned* is $\mathit{yawn}(X)$:

(44) *A man arrived. He yawned.*

$$\mathit{a}(X, \mathit{man}(X), \mathit{arrive}(X)), \mathit{yawn}(X)$$

The expression X in this example should be viewed as similar to a variable in Dynamic Predicate Logic (Groenendijk and Stokhof 1991) or a discourse referent in Discourse Representation Theory (Kamp and Reyle 1993). When the pronoun in the second sentence is resolved to the discourse referent X in (44), X is associated with an individual that is a man, arrived, and yawned.

In contrast, these frameworks do not assume that a quantifier like *nobody* or *everyone* introduces a discourse referent into the context, since the mini-discourses in (45) are unacceptable:

(45) a. *Nobody arrived. *He yawned.*

b. *Everyone arrived. *He yawned.*

In fact, even a referent introduced by an indefinite in the scope of quantifiers like *nobody* or *everyone* or in the scope of negation is not assumed to persist (Karttunen 1976; Groenendijk and Stokhof 1991; Kamp and Reyle 1993), accounting for the unacceptability of the mini-discourse in (46) when *a mouse* is assumed to have narrow scope:

(46) *Nobody saw a mouse. *It squeaked.*

In the following, we present a theory of anaphora resolution that accounts for these facts.

3.2. Meaning and Context Change

Since we assume that certain phrases can introduce discourse referents into the context, we must provide a representation of the context and the discourse referents it contains. We propose that the full semantic contribution of a sentence is a *pair* consisting of the meaning of the sentence and the context that results from interpretation of the sentence. We write such pairs in square brackets, as in (47), where the first member of the pair is the meaning of the sentence and the second member is the list of context variables that are available after the sentence has been uttered:

(47) *David arrived.*

$$[\mathit{arrive}(\mathit{David}), \langle \mathit{David} \rangle]$$

In (48), the meaning of the sentence *David arrived* is $\mathit{arrive}(\mathit{David})$, the first member of the meaning-context pair. The second member of the pair is the context list, represented as:

(48) $\langle \mathit{David} \rangle$

The list of discourse referents available in a particular context appears in angled brackets. Since we assume that the sentence under discussion is uttered at the beginning of the discourse, the context after the sentence is uttered contains only the discourse referent David.

For a sentence with an indefinite noun phrase like *A man arrived*, we assume the meaning-context pair in (49):

(49) *A man arrived.*

$$[\mathit{a}(X, \mathit{man}(X), \mathit{arrive}(X)), \langle X \rangle]$$

Again, we assume that the sentence in (49) was uttered at the beginning of a discourse, so that after this sentence is uttered the context list contains only the discourse referent X representing a man who arrived.

Quantifiers like *nobody* and *everyone* are different from proper names and indefinites in their effect on the context: they do not introduce any new discourse referents into the discourse. When a sentence like *Nobody arrived* or *Everyone arrived* is uttered, no discourse referents are made contextually available:

(50) *Nobody arrived.*

$$[\mathit{no}(X, \mathit{person}(X), \mathit{arrive}(X)), \langle\rangle]$$

(51) *Everyone arrived.*

$$[\mathit{every}(X, \mathit{person}(X), \mathit{arrive}(X)), \langle\rangle]$$

In these examples, the context is empty, represented as $\langle\rangle$. This explains the infelicity of the mini-discourses in (45) earlier, as there are no discourse referents available to resolve any pronouns in subsequent discourse.

Additionally, indefinites in the scope of certain operators do not introduce discourse referents into the global discourse context. As shown in example (46) of this chapter, the reading of a sentence like *Nobody saw a mouse* in which *a mouse* has narrow scope does not allow reference to a mouse in subsequent discourse. Therefore the context after this sentence is uttered does not contain discourse

(52) *Nobody saw a mouse.*

$[\mathit{no}(X, \mathit{person}(X), \mathit{a}(Y, \mathit{mouse}(Y), \mathit{see}(X, Y))), \langle\rangle]$

3.3. Context: Meaning and Glue

3.3.1. Context and Meaning Constructors

The left-hand side of the meaning constructors we have proposed thus far represents a meaning, and the right-hand side represents the semantic structures associated with that meaning. Since we now propose that the meaning contribution of a sentence is a pair consisting of the sentential meaning and a list of discourse referents, we must enrich the right-hand side of our meaning constructors accordingly. We propose that the right-hand side of the meaning constructor for a sentence like *David arrived* is a pair consisting of the semantic structure associated with the meaning of the sentence and a list of semantic structures associated with the discourse referents that have been introduced:

(53) *David arrived.*

$$f\begin{bmatrix}\text{PRED} & \text{`ARRIVE}\langle\text{SUBJ}\rangle\text{'}\\ \text{SUBJ} & g\begin{bmatrix}\text{PRED} & \text{`DAVID'}\end{bmatrix}\end{bmatrix}$$

$[\mathit{arrive}(\mathit{David}), \langle \mathit{David} \rangle] : f_\sigma \otimes \langle g_\sigma \rangle$

In (53), the left-hand side of the meaning constructor is a pair: the first member of the pair is the meaning of the sentence $\mathit{arrive}(\mathit{David})$, and the second member of the pair is the list of available discourse referents $\langle \mathit{David} \rangle$.

The right-hand side of the meaning constructor contains a new expression, $\otimes$, the *multiplicative conjunction* operator of linear logic. We can think of this operator as similar to the English word *and*: the expression $f_\sigma \otimes \langle g_\sigma \rangle$ represents the availability of a pair of meaning resources f_σ and $\langle g_\sigma \rangle$.

The pair of meaning contributions on the left-hand side of the full meaning constructor in (53) is straightforwardly related to the pair of meaning resources on the right: the meaning $\mathit{arrive}(\mathit{David})$ is associated with the meaning resource f_σ, and the list of discourse referents $\langle \mathit{David} \rangle$ is associated with the corresponding list of semantic structures $\langle g_\sigma \rangle$.

Of course, in some cases more than one discourse referent is available. In such a case, each available discourse referent corresponds to a semantic structure in the context list on the right-hand side. In (54), the meaning of the sentence is $\mathit{select}(\mathit{David}, \mathit{Chris})$, and the list of available discourse referents is $\langle \mathit{David}, \mathit{Chris} \rangle$:

(54) *David selected Chris.*

$$f\begin{bmatrix}\text{PRED} & \text{`SELECT}\langle\text{SUBJ,OBJ}\rangle\text{'}\\ \text{SUBJ} & g\begin{bmatrix}\text{PRED} & \text{`DAVID'}\end{bmatrix}\\ \text{OBJ} & h\begin{bmatrix}\text{PRED} & \text{`CHRIS'}\end{bmatrix}\end{bmatrix}$$

$$[\mathit{select}(\mathit{David},\mathit{Chris}),\langle\mathit{David},\mathit{Chris}\rangle] : f_\sigma \otimes \langle g_\sigma, h_\sigma\rangle$$

These meanings are indexed by the semantic structures on the right-hand side of the meaning constructor: the meaning of the sentence corresponds to the semantic structure f_σ, and the list of discourse referents $\langle\mathit{David},\mathit{Chris}\rangle$ corresponds to the list of semantic structures $\langle g_\sigma, h_\sigma\rangle$.

The alignment between the elements of the context lists on the left and right sides of the meaning constructor in (54) is crucial: the first discourse referent *David* corresponds to the first semantic structure in the context list g_σ, and the second discourse referent *Chris* corresponds to the second semantic structure h_σ. Importantly, however, any order of elements in the context lists is permitted, as long as the correspondence between elements on the left and right sides is maintained. The meaning constructor displayed in (55) is exactly equivalent to the one shown in (54). Again, the discourse referent *David* corresponds to the semantic structure g_σ, and the discourse referent *Chris* corresponds to h_σ:

(55) $[\mathit{select}(\mathit{David},\mathit{Chris}),\langle\mathit{Chris},\mathit{David}\rangle] : f_\sigma \otimes \langle h_\sigma, g_\sigma\rangle$

This freedom of order in the context lists will be important below in our discussion of how the antecedent of an anaphor is determined.

3.3.2. Context and Meaning Assembly

We now turn to the issue of meaning derivation and the semantic and contextual contribution of a proper name like *David*. The lexical entry for *David* is given in (56):

(56) *David* $(\uparrow \text{PRED}) = \text{`DAVID'}$
$\lambda C.[\mathit{David},\langle\mathit{David},C\rangle] : \forall C.C \multimap [\uparrow_\sigma \otimes \langle\uparrow_\sigma, C\rangle]$

The variable C on the left and right sides of the meaning constructor in the second line of (56) represents a context list, which is updated by this meaning constructor by the addition of a discourse referent for *David* on the left and its corresponding semantic structure on the right. In (57), we instantiate the meaning constructor for *David* according to the f-structure labels given in (53):

(57) **[David]** $\lambda C.[\mathit{David},\langle\mathit{David},C\rangle] : \forall C.C \multimap [g_\sigma \otimes \langle g_\sigma, C\rangle]$

The right-hand side of this meaning constructor requires as its argument a context list C:

(58) $\forall C.C \multimap [g_\sigma \otimes \langle g_\sigma, C \rangle]$

When this input context list is available, it is consumed, and a pair of resources is produced: a semantic resource g_σ, and a new, updated context list that we write as $\langle g_\sigma, C \rangle$. The result is a context list that is exactly like the original context list C except that an additional element has been added: the semantic structure g_σ is the first element of the new context list.

The left-hand side of this meaning constructor takes as its argument a context list C:

(59) $\lambda C.[\mathit{David}, \langle \mathit{David}, C \rangle]$

When this argument is provided, a pair of meanings results. The first member of the pair represents the meaning of the proper name *David*, corresponding to the semantic resource g_σ on the right-hand side. The second member represents an updated context list, where the discourse referent *David* has been added as the first member, corresponding to the first element g_σ of the updated context list on the right-hand side.

In the analysis of example (53), we assume the standard meaning constructor for the intransitive verb *arrive*, labeled **[arrive]** in (60). Since this sentence is uttered at the beginning of a discourse, we also provide an empty context resource premise, labeled **[context]**:

(60) Meaning constructor premises for *David arrived*:

$$\begin{array}{lrcl}
\textbf{[context]} & \langle\rangle & : & \langle\rangle \\
\textbf{[arrive]} & \lambda X.\mathit{arrive}(X) & : & g_\sigma \multimap f_\sigma \\
\textbf{[David]} & \lambda C.[\mathit{David}, \langle \mathit{David}, C \rangle] & : & \forall C.C \multimap [g_\sigma \otimes \langle g_\sigma, C \rangle]
\end{array}$$

We will also make use of deduction rules for the multiplicative conjunction operator $\otimes$. In (61) and (62) we present the rules for $\otimes$ augmented with the corresponding operations on the associated context/meaning pairs on the left-hand side.[4] We use the subscript $\langle c \rangle$ as a special semantic type for context lists. The rule in (61) allows the context/meaning pairs to be split into two meaning constructors, a simple noncontextual meaning constructor $M : T$ and a context meaning constructor $MC : TC_{\langle c \rangle}$. Each of these meaning constructors can be used in subsequent deductive steps:

(61) **Context split** $$\frac{[M, MC] : T \otimes TC_{\langle c \rangle}}{M : T \otimes MC : TC_{\langle c \rangle}}$$

[4] See the appendix for the full set of linear logic deduction rules.

The rule in (62) allows the converse operation, where a context-meaning pair constructor $[M, MC] : T \otimes TC_{\langle c \rangle}$ is formed from a standard meaning constructor and a context meaning constructor:

(62) **Context merge** $$\frac{M : T, \quad MC : TC_{\langle c \rangle}}{[M, MC] : T \otimes TC_{\langle c \rangle}}$$

With the augmented meaning constructors that we now assume, the conclusion of a semantically complete and coherent deduction is a meaning-context pair. As usual, we assume that a semantically complete and coherent deduction results only when a nonimplicational conclusion is reached from the premises provided by the utterance with no additional unused premises present.

We begin the deduction by combining the premises in (60) labeled **[context]** and **[David]**, obtaining the meaning constructor labeled **[context-David]** in (63). The empty context is updated to include the semantic structure g_σ on the right-hand side and the discourse referent *David* on the left-hand side:

(63) **[context-David]** $[\mathit{David}, \langle \mathit{David} \rangle] : g_\sigma \otimes \langle g_\sigma \rangle$

We now use the deduction rule **Context split** to produce two meaning constructors, the updated context **[newcontext]** and the familiar resource for the proper name *David*, **[David2]**:

(64) **[newcontext]** $\langle \mathit{David} \rangle : \langle g_\sigma \rangle$

[David2] $\mathit{David} : g_\sigma$

We combine the resource **[David2]** with the verb meaning constructor **[arrive]** to produce the meaning constructor labeled **[David-arrive]** in (65):

(65) **[David-arrive]** $\mathit{arrive}(\mathit{David}) : f_\sigma$

The meaning constructor in (65) represents a semantically complete deduction for *David arrived*, but not a semantically coherent one, since the meaning constructor labeled **[newcontext]** in (64) has not been used. We make use of the deduction rule **Context merge** to combine **[David-arrive]** with **[newcontext]**, obtaining the semantically complete and coherent result in (66), as desired:

(66) **[newcontext], [David-arrive]** $\vdash [\mathit{arrive}(\mathit{David}), \langle \mathit{David} \rangle] : f_\sigma \otimes \langle g_\sigma \rangle$

3.4. Anaphora and Meaning Assembly

Section 2 of this chapter showed that syntactic and thematic constraints on anaphora resolution depend on the syntactic and thematic role of the antecedent. In fact, the grammatical function of an antecedent can play a role even in cross-sentential anaphor resolution, where the antecedent and anaphor are not in the same sentence (Grosz et al. 1995). We can exploit the correspondence function σ to relate the semantic structure of a discourse referent to its corresponding f-structure, so that syntactic or thematic constraints on anaphor resolution can be enforced.

3.4.1. Antecedents of Anaphors

We have seen that certain phrases, such as the proper name *David*, make available a discourse referent associated with a particular semantic structure. We now examine the semantic contribution of pronouns, the relation an anaphor bears to the discourse referent of its antecedent, and the role in constraining antecedent choice played by the other linguistic structures of LFG.

The precise nature of the relation between an anaphor and its antecedent has been the subject of much study. Kehler (1995) examines the nature of the anaphor-antecedent relation and proposes that an anaphor is annotated with information about its antecedent, providing evidence for this position from the interpretation of elliptical sentences as well as from other sources. We follow Kehler, Higginbotham (1983), Barwise (1987), Dalrymple et al. (1997b), and many others in representing the anaphor-antecedent relation by means of an annotation on the pronoun. In particular, we propose that the semantic structure of the antecedent of an anaphor appears as the value of the attribute ANTECEDENT in the semantic structure of the anaphor.

Consider a discourse whose first sentence is *David arrived*, with the f-structure given in (67):

(67) *David arrived.*

$$f\begin{bmatrix} \text{PRED} & \text{`ARRIVE}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & g\begin{bmatrix}\text{PRED} & \text{`DAVID'}\end{bmatrix}\end{bmatrix}$$

We can continue this discourse with the sentence *He yawned.* In this case, the most likely antecedent for the pronoun *He* is the subject of the first sentence, *David*, whose semantic structure is g_σ:

(68) *He yawned.*

$$h\begin{bmatrix} \text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & i\begin{bmatrix}\text{PRED} & \text{`PRO'}\end{bmatrix}\end{bmatrix} \dashrightarrow i_\sigma\begin{bmatrix}\text{ANTECEDENT} & g_\sigma\end{bmatrix}$$

The binding relation in (68) is syntactically wellformed, since it does not violate any constraints on antecedency imposed by the pronoun *he*.

With the binding relation between an anaphor and its antecedent made explicit in the semantic structure, we can impose the syntactic constraints on binding discussed in Section 2 of this chapter. For instance, we have seen that the antecedent of the English reflexive pronoun *himself* must appear in the Minimal Complete Nucleus relative to the pronoun, the minimal f-structure containing the pronoun and a SUBJ function. Example (5) of this chapter, repeated in (69), illustrates a wellformed binding relation between *himself* and its antecedent *David*:

(69) *David$_i$ compared Chris to himself$_i$.*

$$\left[\begin{array}{ll} \text{PRED} & \text{`COMPARE}\langle\text{SUBJ,OBJ,OBL}_{\text{GOAL}}\rangle\text{'} \\ \text{SUBJ} & d\left[\text{PRED} \quad \text{`DAVID'}\right] \\ \text{OBJ} & \left[\text{PRED} \quad \text{`CHRIS'}\right] \\ \text{OBL}_{\text{GOAL}} & p\left[\begin{array}{ll}\text{PRED} & \text{`PRO'} \\ \text{PRONTYPE} & \text{REFL}\end{array}\right] \end{array}\right] \dashrightarrow p_\sigma\left[\text{ANTECEDENT} \quad d_\sigma\right]$$

[redacted]

The relation between an anaphor like *himself* and its antecedent is given by an positive constraint of the general form in (70), repeated from (23) of this chapter, where p is the f-structure of the anaphor:

(70) $((\text{GF}^* \ \text{GF}_{\text{pro}} \ p) \ \text{GF})$

According to the Minimal Complete Nucleus binding condition, the path GF* may not pass through an f-structure with a SUBJ function:

(71) $((\underset{\neg(\rightarrow \text{SUBJ})}{\text{GF}^*} \quad \text{GF}_{\text{pro}} \ p) \ \text{GF})$

The expression in (71) represents the set of permissible f-structure antecedents of the reflexive pronoun *himself*, including *David* in example (69). We require the semantic structure of the antecedent, appropriately constrained, to appear as the value of the attribute ANTECEDENT in the pronoun's semantic structure. Thus, the following equation appears in the lexical entry of the reflexive pronoun *himself*:

(72) $(\uparrow_\sigma \ \text{ANTECEDENT}) = ((\underset{\neg(\rightarrow \text{SUBJ})}{\text{GF}^*} \quad \text{GF}_{\text{pro}} \uparrow) \ \text{GF})_\sigma$

The equation in (72) requires the antecedent of the pronoun ($\uparrow_\sigma$ ANTECEDENT) to be the semantic structure of some f-structure that appears within the Minimal Complete Nucleus relative to the pronoun. In example (69) this condition is met, and the binding relation is permitted.

Example (7) of this chapter, repeated in (73), illustrates a impermissible binding relation:

(73) **David*$_i$ *thought that Chris had seen himself*$_i$.

$$\left[\begin{array}{ll} \text{PRED} & \text{`THINK}\langle\text{SUBJ,COMP}\rangle\text{'} \\ \text{SUBJ} & d\left[\begin{array}{ll}\text{PRED} & \text{`DAVID'}\end{array}\right] \\ \text{COMP} & c\left[\begin{array}{ll} \text{PRED} & \text{`SEE}\langle\text{SUBJ,OBJ}\rangle\text{'}\\ \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`CHRIS'}\end{array}\right] \\ \text{OBJ} & p\left[\begin{array}{ll}\text{PRED} & \text{`PRO'}\\ \text{PRONTYPE} & \text{REFL}\end{array}\right]\end{array}\right]\end{array}\right] \dashrightarrow p_{\sigma}\left[\begin{array}{ll}\text{ANTECEDENT} & d_{\sigma}\end{array}\right]$$

The semantic structure d_σ cannot appear as the ANTECEDENT value of the reflexive pronoun *himself* because the f-structure labeled d does not stand in a syntactically permissible antecedent relation to the pronoun f-structure p: the Minimal Complete Nucleus for the pronoun p is the COMP f-structure, labeled c in (73). In other words, the binding equation given in (72) is not satisfied when the antecedent *David* is chosen for the reflexive pronoun *himself* in (73).

Other syntactic and thematic constraints are imposed similarly. Constraints on f-precedence and thematic relations between an anaphor and its antecedent, discussed in Section 2.2 and Section 2.3 of this chapter, also constrain the choice of ANTECEDENT value in a pronoun's semantic structure. Section 2.1.2 of this chapter discusses negative constraints, which rule out particular antecedents for a pronoun; negative constraints prevent syntactically impermissible antecedents from being chosen as the value of the ANTECEDENT attribute of a pronoun's semantic structure.[5]

3.4.2. Anaphora and Proper Names

Section 3.3 of this chapter showed that the meaning constructor for the sentence *David arrived* is as given in (74):

(74) *David arrived.*

$$f\left[\begin{array}{ll} \text{PRED} & \text{`ARRIVE}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & g\left[\begin{array}{ll}\text{PRED} & \text{`DAVID'}\end{array}\right]\end{array}\right]$$

$$[\mathit{arrive}(\mathit{David}), \langle \mathit{David}\rangle] : f_\sigma \otimes \langle g_\sigma \rangle$$

We now assume that this sentence is immediately followed by the sentence *He yawned* and that the antecedent of the pronoun *he* is the subject of the first sen-

[5] In fact, in a complete treatment of negative constraints a stronger condition is required. Negative constraints require *noncoreference* with all elements in the negative domain, as discussed by Dalrymple (1993, 2.2.2); constraints on antecedent choice rule out most but not all unwanted possibilities. We leave this aspect of the interpretation of negative constraints for future work.

tence *David*. Under this assumption, the meaning constructor for *He yawned* is given in (75):

(75) *He yawned.*

$$h\left[\begin{array}{ll}\textsc{pred} & \text{`}\textsc{yawn}\langle\textsc{subj}\rangle\text{'}\\ \textsc{subj} & i\left[\textsc{pred}\quad \text{`}\textsc{pro}\text{'}\right]\end{array}\right] \dashrightarrow i_\sigma\left[\textsc{antecedent}\quad g_\sigma[\;]\right]$$

$$[\mathit{yawn}(\mathit{David}), \langle \mathit{David}, \mathit{David}\rangle] : h_\sigma \otimes \langle i_\sigma, g_\sigma\rangle$$

In (75), we have chosen the semantic structure g_σ as the value of the ANTECEDENT attribute of the pronoun. In the following, we show how the pronoun updates and chooses an antecedent from the context.

We propose the lexical entry in (76) for the pronoun *he*:

(76) *he* $(\uparrow \textsc{pred}) = \text{`}\textsc{pro}\text{'}$

$$\lambda C.[\mathit{first}(C), \langle \mathit{first}(C), C\rangle] :$$
$$\forall C.\langle(\uparrow_\sigma \textsc{antecedent}), C\rangle \multimap [\uparrow_\sigma \otimes \langle \uparrow_\sigma, (\uparrow_\sigma \textsc{antecedent}), C\rangle]$$

Instantiating the meaning constructor in (76) according to the f-structure labels in (75), we have the following meaning constructor contribution:

(77) Meaning constructor for *he*:

$$\lambda C.[\mathit{first}(C), \langle \mathit{first}(C), C\rangle] \ : \ \forall C.\langle g_\sigma, C\rangle \multimap [i_\sigma \otimes \langle i_\sigma, g_\sigma, C\rangle]$$

We begin by examining the right-hand side of this meaning constructor:

(78) $\forall C.\langle g_\sigma, C\rangle \multimap [i_\sigma \otimes \langle i_\sigma, g_\sigma, C\rangle]$

This expression requires a context list of the form $\langle g_\sigma, C\rangle$ as an argument, meaning that the context list must include the semantic structure g_σ, the antecedent of the pronoun, as its first element. Requiring the semantic structure for the antecedent to appear as an element of the argument context list entails that only semantic structures that are available in the context list are suitable antecedents for a pronoun. Recall that all permutations of elements of the context list are possible; thus, any element of the list may appear in first position, so that any contextually available antecedent may be chosen, as long as syntactic and thematic constraints on antecedent choice imposed by the pronoun are met. If no syntactic, thematic, or information-structural constraints on antecedent choice are imposed, any element of the context set may be chosen as the antecedent.

When a semantic resource context list $\langle g_\sigma, C\rangle$ is found, it is consumed and a pair of resources $i_\sigma \otimes \langle i_\sigma, g_\sigma, C\rangle$ is produced. i_σ is the semantic structure of the pronoun, which becomes available as a semantic resource when the pronoun is resolved. Additionally, an updated context resource list $\langle i_\sigma, g_\sigma, C\rangle$ is produced, which is the same as the original context list $\langle g_\sigma, C\rangle$ except that the pronoun semantic structure i_σ is added. Thus, the semantic structure of the pronoun i_σ is made available as a potential antecedent in subsequent pronominal reference, as in an example like *He$_i$ selected himself$_i$*. The antecedent g_σ also remains contextually available as the context is monotonically enriched; this is necessary because in some cases the same antecedent is selected by more than one pronoun. In the Dutch example in (79), for example, the antecedent of both occurrences of the subject-oriented reflexive *zichzelf* is the subject, *Jan*:

(79) *Jan sprak met zichzelf over zichzelf.*
Jan talked with self about self
'Jan$_i$ talked with self$_i$ about self$_i$.'

The left-hand side of the meaning constructor in (78) is given in (80). The expression in (80) takes as an argument a context list C. We use the expression $\mathit{first}(C)$ for the first element of C, which is the meaning of the antecedent:

(80) $\lambda C.[\mathit{first}(C), \langle \mathit{first}(C), C\rangle]$

When the context argument C is provided, a pair of meanings is produced. The first member of the pair, $\mathit{first}(C)$, is the meaning of the pronoun's antecedent, which corresponds to the anaphor's semantic structure g_σ. The meaning of the antecedent is thereby assigned to the pronoun and becomes the meaning contribution of the pronoun. The second member of the pair is the updated context list, which differs from the original context list C in that the pronoun meaning $\mathit{first}(C)$ is added as the first element of the new context list.

An example may help to make the pronoun meaning constructor more clear. The meaning constructor premises for *He yawned* are given in (81):

(81) Meaning constructor premises for *He yawned*:

[context] $\langle \mathit{David}\rangle : \langle g_\sigma\rangle$

[yawn] $\lambda X.\mathit{yawn}(X) : i_\sigma \multimap h_\sigma$

[he] $\lambda C.[\mathit{first}(C), \langle \mathit{first}(C), C\rangle] : \forall C.\langle g_\sigma, C\rangle \multimap [i_\sigma \otimes \langle i_\sigma, g_\sigma, C\rangle]$

We first combine the premises labeled **[context]** and **[he]**. On the right-hand side, the context resource required by **[he]** must have g_σ as its first element, and this is true of **[context]**. A pair of resources is produced: a semantic resource i_σ for the pronoun, and an updated context $\langle i_\sigma, g_\sigma\rangle$.

On the left-hand side, the first element of **[context]**, $\mathit{first}(C)$, is the discourse referent *David*. The resulting expression is a pair consisting of the discourse referent *David*, which becomes the meaning of the pronoun, and the updated context $\langle \mathit{David}, \mathit{David}\rangle$, which is the same as the original context list except that the pronoun meaning *David* has been added as the first element. The result is the meaning constructor labeled **[context-he]** in (82):

(82) **[context-he]** $[\textit{David}, \langle \textit{David}, \textit{David} \rangle] \,:\, i_\sigma \otimes \langle i_\sigma, g_\sigma \rangle$

The meaning constructor **[context-he]** provides a meaning *David* for the pronoun; this is appropriate, since the antecedent of the pronoun is the subject *David* of the previous sentence. It also reflects an updated context: both the antecedent g_σ and the pronoun i_σ are elements of the context list, each associated with the discourse referent *David*.

We now combine the meaning constructor **[context-he]** with the verb meaning constructor **[yawn]** in the way described in Section 3.3.2 of this chapter, obtaining the semantically complete and coherent meaning constructor in (83), as desired:

(83) **[context], [he], [yawn]** $\vdash [\textit{yawn}(\textit{David}), \langle \textit{David}, \textit{David} \rangle] \,:\, h_\sigma \otimes \langle i_\sigma, g_\sigma \rangle$

3.5. Anaphora and Indefinites

3.5.1. Indefinites in Context

Like a proper name, an indefinite noun phrase like *a man* or *someone* introduces a discourse referent into the context. The f-structure and meaning constructor for the sentence *Someone arrived* are given in (84):

(84) *Someone arrived.*

$$f\begin{bmatrix} \text{PRED} & \text{`ARRIVE}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & g\begin{bmatrix} \text{PRED} & \text{`SOMEONE'} \end{bmatrix} \end{bmatrix}$$

$$[\textit{a}(X, \textit{person}(X), \textit{arrive}(X)), \langle X \rangle] : f_\sigma \otimes \langle g_\sigma \rangle$$

After this sentence is uttered, the context list contains the discourse referent X representing an individual who is a person and arrived, corresponding to the semantic resource g_σ.

We assume the lexical entry in (85) for *someone*:

(85) *someone* $(\uparrow \text{PRED}) = \text{`SOMEONE'}$

$$\lambda C.\lambda S.[\textit{a}(X, \textit{person}(X), \textit{sem}(S(X,C))), \textit{cxt}(S(X,C))] :$$
$$\forall C_{in}.\forall C_{out}.\forall H.C_{in} \multimap [[[\uparrow_\sigma \otimes \langle \uparrow_\sigma, C_{in} \rangle] \multimap [H \otimes C_{out}]] \multimap [H \otimes C_{out}]]$$

We first examine the right-hand side of the meaning constructor in (85). Instantiating the $\uparrow$ metavariables in this lexical entry according to the f-structure labels in (84), we have the following expression:

(86) $\forall C_{in}.\forall C_{out}.\forall H.C_{in} \multimap [[[g_\sigma \otimes \langle g_\sigma, C_{in} \rangle] \multimap [H \otimes C_{out}]] \multimap [H \otimes C_{out}]]$

We can compare this expression to the right-hand side of the corresponding simple noncontextual quantifier meaning constructor, as described in Chapter 9, Section 8:

(87) $\forall H.[g_\sigma \multimap H] \multimap H$

The obvious difference between (86) and (87) is the presence of the variables C_{in} and C_{out} representing context lists in (86). In the following, we justify each difference in turn.

In (86), the currently available context list C_{in} must be consumed:

(88) $\forall C_{in}.\forall C_{out}.\forall H.\underline{C_{in}} \multimap [[[g_\sigma \otimes \langle g_\sigma, C_{in}\rangle] \multimap [H \otimes C_{out}]] \multimap [H \otimes \; C_{out}]]$

Next, we note that the implicational meaning constructor $g_\sigma \multimap H$ in (87) represents the scope of the noncontextual quantifier. Since we now assume that meaning contributions consist of meaning-context pairs, we enrich each side of this implication accordingly. The antecedent of the implication in (87) consists of the meaning-context pair $g_\sigma \otimes \langle g_\sigma, C_{in}\rangle$, where the incoming context C_{in} has been augmented with the indefinite's semantic structure g_σ so that it is available for resolution of pronouns inside the scope. The consequent consists of the pair $H \otimes C_{out}$, where a new, possibly augmented context list C_{out} appears:

(89) $\forall C_{in}.\forall C_{out}.\forall H.C_{in} \multimap [[[\underline{g_\sigma \otimes \langle g_\sigma, C_{in}\rangle}] \multimap [\underline{H \otimes C_{out}}]] \multimap [H \otimes \; C_{out}]]$

C_{out} can differ from $\langle g_\sigma, C_{in}\rangle$ if an indefinite, proper name, or pronoun appears inside the scope of the indefinite and is added to the context list. If no additions to the context are made inside the scope, C_{out} is the same as $\langle g_\sigma, C_{in}\rangle$.

Finally, we assume that the resulting meaning-context pair is $H \otimes C_{out}$, where H corresponds to the scope semantic structure and C_{out} corresponds to the appropriately updated context list that is produced:

(90) $\forall C_{in}.\forall C_{out}.\forall H.C_{in} \multimap [[[g_\sigma \otimes \langle g_\sigma, C_{in}\rangle] \multimap [H \otimes C_{out}]] \multimap \underline{[H \otimes \; C_{out}]}]$

In sum, this meaning constructor consumes the currently available context list C_{in}, augments it with g_σ, and produces a modified and updated context list C_{out} as a result.

We now turn to an examination of the expression on the left-hand side of the meaning constructor for *someone* in (85), repeated in (91):

(91) $\lambda C.\lambda S.[\mathit{a}(X, \mathit{person}(X), \mathit{sem}(S(X,C))), \mathit{cxt}(S(X,C))]$

We can compare the expression in (91) to the noncontextual meaning contribution of a quantifier presented in Chapter 9, Section 8:

(92) $\lambda S.\mathit{a}(X, \mathit{person}(X), S(X))$

The expression in (92) differs from the one in (91) in two important ways. First, the expression in (91) requires a context argument C in addition to a scope meaning S, corresponding to the requirement for a context list C_{in} on the right-hand side.

Second, the derived scope reading $S(X,C)$ represents a meaning-context pair, unlike the meaning $S(X)$ in (92), which represents a proposition. The scope meaning S in (92) has type $\langle e \rightarrow t\rangle$, which is (for example) the type of an intransitive verb. In the current setting, pairs of meanings are derived; thus, S in (91) is of type $\langle\langle e,c\rangle \rightarrow \langle t,c\rangle\rangle$: a function from an individual/context pair $\langle e,c\rangle$ to a truth-value/context pair $\langle t,c\rangle$. This corresponds to the embedded implication $[g_\sigma \otimes \langle g_\sigma, C_{in}\rangle] \multimap [H \otimes C_{out}]$ on the right-hand side. Thus, when S is applied to a pair consisting of an individual X and a context C, a meaning-context pair of type $\langle t,c\rangle$ results.

The expressions $\mathit{sem}(S(X,C))$ and $\mathit{cxt}(S(X,C))$ refer to members of the meaning-context pair $S(X,C)$: the expression $\mathit{sem}(S(X,C))$ refers to the meaning (first) member of the pair, and the expression $\mathit{cxt}(S(X,C))$ refers to the context (second) member of the pair. In the example under discussion, the following facts are relevant:

(93) If $S(X,C) = [\mathit{arrive}(X), \langle X\rangle]$, then

$\mathit{sem}(S(X,C)) = \mathit{arrive}(X)$ and $\mathit{cxt}(S(X,C)) = \langle X\rangle$

In (91), the meaning $\mathit{sem}(S(X,C))$ becomes the scope of the indefinite, and the context $\mathit{cxt}(S(X,C))$ is the resulting context.

In the derivation of the meaning of *Someone arrived*, the premises in (94) are relevant:

(94) Meaning constructor premises for *Someone arrived*:

[context] $\langle\rangle : \langle\rangle$

[arrive] $\lambda X.\mathit{arrive}(X) : g_\sigma \multimap f_\sigma$

[someone] $\lambda C.\lambda S.[a(X, \mathit{person}(X), \mathit{sem}(S(X,C))), \mathit{cxt}(S(X,C))]$:
$\forall C_{in}.\forall C_{out}.\forall H.C_{in} \multimap [[[g_\sigma \otimes \langle g_\sigma, C_{in}\rangle] \multimap [H \otimes C_{out}]] \multimap [H \otimes C_{out}]]$

We begin the derivation by combining the meaning constructors labeled **[context]** and **[someone]**, producing the meaning constructor labeled **[cxt-someone]** in (95):

(95) **[cxt-someone]** $\lambda S.[a(X, \mathit{person}(X), \mathit{sem}(S(X,\langle\rangle))), \mathit{cxt}(S(X,\langle\rangle))]$:
$\forall C_{out}.\forall H.[[g_\sigma \otimes \langle g_\sigma\rangle] \multimap [H \otimes C_{out}]] \multimap [H \otimes C_{out}]$

Next, we observe that the meaning constructor labeled **[cxt-arrive]** in (96) is logically equivalent to **[arrive]**:

(96) **[arrive]** $\vdash$

[cxt-arrive] $\lambda C.\lambda X.[\mathit{arrive}(X), C] : \forall C.[g_\sigma \otimes C] \multimap [f_\sigma \otimes C]]$

Since **[cxt-arrive]** matches the requirements of **[cxt-someone]**, we can combine **[cxt-arrive]** and **[cxt-someone]** to obtain the complete and coherent meaning in (97), as desired:

(97) **[cxt-someone], [cxt-arrive]** $\vdash$

$$[\mathit{a}(X, \mathit{person}(X), \mathit{arrive}(X)), \langle X \rangle] : f_\sigma \otimes \langle g_\sigma \rangle$$

3.5.2. Indefinites as Antecedents

The context made available by the sentence *Someone arrived* contains the subject semantic structure g_σ, corresponding to the discourse referent X representing a person who arrived:

(98) Context of utterance:

$\langle X \rangle : \langle g_\sigma \rangle$

Interpretation of the sentence *He yawned* in this context proceeds as described in Section 3.4.2 of this chapter, since the incoming context in the two situations is very similar. Assuming that the antecedent of the pronoun *He* is the subject of the sentence *Someone arrived*, the f-structure, semantic structure, and meaning constructor for *He yawned* are:

(99) *He yawned.*

$$h\begin{bmatrix} \text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & i\begin{bmatrix}\text{PRED} & \text{`PRO'}\end{bmatrix} \end{bmatrix} \overset{\sigma}{\dashrightarrow} i_\sigma\begin{bmatrix}\text{ANTECEDENT} & g_\sigma\end{bmatrix}$$

$[\mathit{yawn}(X), \langle X, X \rangle] : h_\sigma \otimes \langle i_\sigma, g_\sigma \rangle$

The meaning constructor premises in (100) are involved in the derivation of the meaning constructor in (99):

(100) Meaning constructor premises for *He yawned*:

[context]	$\langle X \rangle$	$: \langle g_\sigma \rangle$
[yawn]	$\lambda X.\mathit{yawn}(X)$	$: i_\sigma \multimap h_\sigma$
[he]	$\lambda C.[\mathit{first}(C), \langle \mathit{first}(C), C \rangle]$	$: \forall C.\langle g_\sigma, C \rangle \multimap [i_\sigma \otimes \langle i_\sigma, g_\sigma, C \rangle]$

As shown for example (82), the meaning constructor **[context-he]** can be deduced from the meaning constructors **[context]** and **[he]** in (100):

(101) **[context-he]** $[X, \langle X, X \rangle] : i_\sigma \otimes \langle i_\sigma, g_\sigma \rangle$

From the meaning constructor **[context-he]** and **[yawn]**, we derive the following semantically complete and coherent meaning constructor for *He yawned*:

(102) **[context-he], [yawn]** $\vdash$ $[\mathit{yawn}(X), \langle X, X\rangle] : h \otimes \langle i_\sigma, g_\sigma\rangle$

3.6. Context and Quantifiers

Indefinite phrases like *someone* or *a man* are treated differently from quantified noun phrases like *nobody* or *every woman* in theories like Discourse Representation Theory (Kamp 1981; Kamp and Reyle 1993) and Dynamic Predicate Logic (Groenendijk and Stokhof 1991): as discussed in Section 3.1 of this chapter, quantified noun phrases do not introduce a discourse referent into the context. The f-structure and meaning constructor for the sentence *Nobody arrived* are given in (103):

(103) *Nobody arrived. (*He yawned.)*

$$f\begin{bmatrix} \text{PRED} & \text{‘ARRIVE}\langle\text{SUBJ}\rangle\text{’} \\ \text{SUBJ} & g\begin{bmatrix} \text{SPEC} & \text{‘NO’} \\ \text{PRED} & \text{‘PERSON’} \end{bmatrix} \end{bmatrix}$$

$$[\mathit{no}(X, \mathit{person}(X), \mathit{arrive}(X)), \langle\rangle] : f_\sigma \otimes \langle\rangle$$

No discourse referent is available in the context after this sentence is uttered; this explains the infelicity of a continuation like *He yawned*, which requires an antecedent for interpretation of its pronoun subject.

We assume the following lexical entry for the negative quantifier *nobody*:

(104) *nobody* $(\uparrow \text{PRED}) = \text{‘NOBODY’}$

$\lambda C.\lambda S.[\mathit{no}(X, \mathit{person}(X), \mathit{sem}(S(X, C))), C]$:

$\forall C_{in}.\forall C_{out}.\forall H.C_{in} \multimap [[[\uparrow_\sigma \otimes \langle \uparrow_\sigma, C_{in}\rangle] \multimap [H \otimes C_{out}]] \multimap [H \otimes\ C_{in}]]$

Instantiating the meaning constructor in (104) according to the f-structure labels in (103), we have the instantiated meaning constructor in (105):

(105) **[nobody]** $\lambda C.\lambda S.[\mathit{no}(X, \mathit{person}(X), \mathit{sem}(S(X, C))), C]$:

$\forall C_{in}.\forall C_{out}.\forall H.C_{in} \multimap [[[g_\sigma \otimes \langle g_\sigma, C_{in}\rangle] \multimap [H \otimes C_{out}]] \multimap [H \otimes\ C_{in}]]$

For comparison, we display the meaning constructor for *someone* in (106), with the crucial differences underlined:

(106) **[somebody]** $\lambda C.\lambda S.[a(X, \mathit{person}(X), \mathit{sem}(S(X, C))), \underline{\mathit{cxt}(S(X, C))}]$:

$\forall C_{in}.\forall C_{out}.\forall H.C_{in} \multimap [[[g_\sigma \otimes \langle g_\sigma, C_{in}\rangle] \multimap [H \otimes C_{out}]] \overline{\multimap [H \otimes}\ \underline{C_{out}}]]$

On the right-hand side of the meaning constructor for *someone* in (106), the context list that is made available is the context list C_{out}, which contains the semantic structure g_σ as well as any semantic structures contributed within the scope of *someone*. In contrast, on the right-hand side of the meaning constructor for *nobody* in (105), the context list that is made available is C_{in}, the incoming context list. Neither the semantic structure g_σ nor any semantic structures added within the scope of *nobody* are available outside its scope. In effect, the contextual contribution of *nobody* is to leave the incoming context unchanged. Thus, like the earlier proposal of Dalrymple et al. (1997b), this analysis has the desirable property of correctly characterizing interactions between quantifier scope and bound anaphora: any pronouns bound by a quantifier like *everyone* or *nobody* must appear within the scope of the quantifier, since the semantic structure for the quantifier is only available in contexts within its scope, not outside it.

For the sentence *Nobody arrived*, we assume the meaning constructor premises given in (107):

(107) Meaning constructor premises for *Nobody arrived*:

[context] $\langle\rangle : \langle\rangle$

[arrive] $\lambda X.\mathit{arrive}(X) : g_\sigma \multimap f_\sigma$

[nobody] $\lambda C.\lambda S.[\mathit{no}(X, \mathit{person}(X), \mathit{sem}(S(X,C))), C]$:
$\forall C_{in}.\forall C_{out}.\forall H.C_{in} \multimap [[[g_\sigma \otimes \langle g_\sigma, C_{in}\rangle] \multimap [H \otimes C_{out}]] \multimap [H \otimes C_{in}]]$

These premises are very similar to the ones relevant for the sentence *Someone arrived* given in (94), and the derivation of the meaning of the sentence proceeds analogously, except that the resulting context list corresponds to the empty context C_{in} rather than to the augmented context C_{out}:

(108) **[context], [nobody], [arrive]** $\vdash$

$$[\mathit{no}(X, \mathit{person}(X), \mathit{arrive}(X)), \langle\rangle] : f \otimes \langle\rangle$$

The difference between the context-changing potential of an indefinite like *someone* and a negative quantifier like *nobody* is represented explicitly in the linear logic derivation, in terms of the context that each makes available when it is evaluated.

Finally, we briefly examine the derivation of the meaning of the sentence *Nobody saw a mouse*. If the indefinite *a mouse* appears inside the scope of the quantifier *nobody*, *a mouse* does not contribute to the overall context, accounting for the infelicity of a sentence like *It squeaked* in subsequent discourse:

(109) *Nobody saw a mouse. (*It squeaked.)*

$$f\left[\begin{array}{ll} \text{PRED} & \text{`SEE}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & g\left[\begin{array}{ll} \text{PRED} & \text{`NOBODY'} \end{array}\right] \\ \text{OBJ} & h\left[\begin{array}{ll} \text{SPEC} & \text{`A'} \\ \text{PRED} & \text{`MOUSE'} \end{array}\right] \end{array}\right]$$

$$[\mathit{no}(X, \mathit{person}(X), a(Y, \mathit{mouse}(Y), \mathit{see}(X,Y))), \langle\rangle] : f_\sigma \otimes \langle\rangle$$

The derivation of the narrow-scope indefinite reading of this sentence proceeds on the basis of the following meaning constructors, which have been instantiated according to the f-structure labels in (109):

(110) Meaning constructor premises for *Nobody saw a mouse*:

[context] $\langle\rangle : \langle\rangle$

[see] $\lambda X.\lambda Y.\mathit{see}(X,Y) \;:\; g_\sigma \multimap [h_\sigma \multimap f_\sigma]$

[nobody] $\lambda C.\lambda S.[\mathit{no}(X,\mathit{person}(X),\mathit{sem}(S(X,C))),C] \;:$
$\forall C_{in}.\forall C_{out}.\forall H.C_{in} \multimap [[[g_\sigma \otimes \langle g_\sigma, C_{in}\rangle] \multimap [H \otimes C_{out}]] \multimap [H \otimes C_{in}]]$

[a-mouse] $\lambda C.\lambda S.[a(X,\mathit{mouse}(X),\mathit{sem}(S(X,C))),\mathit{cxt}(S(X,C))] \;:$
$\forall C_{in}.\forall C_{out}.\forall H.C_{in} \multimap [[[h_\sigma \otimes \langle h_\sigma, C_{in}\rangle] \multimap [H \otimes C_{out}]] \multimap [H \otimes C_{out}]]$

Combining the premises labeled **[a-mouse]** and **[see]**, we have the meaning constructor labeled **[see-a-mouse]** in (111):

(111) **[see-a-mouse]** $[\lambda C.\lambda X.a(Y,\mathit{mouse}(Y),\mathit{see}(X,Y)),\langle Y,C\rangle] :$
$\forall C.C \multimap [g_\sigma \multimap [f_\sigma \otimes \langle h_\sigma, C\rangle]]$

We combine this meaning constructor with **[nobody]**, producing the meaning constructor labeled **[nobody-see-a-mouse]** in (112):

(112) **[nobody-see-a-mouse]**
$\lambda C.[\mathit{no}(X,\mathit{person}(X),a(Y,\mathit{mouse}(Y),\mathit{see}(X,Y))),C] :$
$\forall C.C \multimap [f_\sigma \otimes C]$

Finally, we combine this meaning constructor with **[context]**, producing the semantically complete and coherent result in (113), as desired:

(113) **[context], [nobody-see-a-mouse]** $\vdash$
$[\mathit{no}(X,\mathit{person}(X),a(Y,\mathit{mouse}(Y),\mathit{see}(X,Y))),\langle\rangle] : f_\sigma \otimes \langle\rangle$

This reading of the sentence makes no discourse referents available for anaphora resolution in subsequent discourse.

In contrast, if the indefinite takes wide scope, a discourse referent corresponding to a mouse is available. Beginning with the premises in (110), we can combine the premises labeled **[nobody]** and **[see]**, producing the meaning constructor labeled **[nobody-see]** in (114):

(114) **[nobody-see]** $[\lambda C.\lambda X.\mathit{no}(X,\mathit{person}(X),\mathit{see}(X,Y)),C] :$
$\forall C.C \multimap [g_\sigma \multimap [f_\sigma \otimes C]]$

Combining this meaning constructor with **[a-mouse]**, we have the meaning constructor labeled **[nobody-saw-a-mouse2]** in (115):

(115) **[nobody-see-a-mouse2]**

$$\lambda C.[\mathit{a}(Y, \mathit{mouse}(Y), \mathit{no}(X, \mathit{person}(X), \mathit{see}(X,Y))), \langle Y, C\rangle] : \forall C.C \multimap [f_\sigma \otimes \langle h_\sigma, C\rangle]$$

Finally, we combine **[nobody-see-a-mouse2]** with **[context]**, producing the result in (116); this result differs from the one in (113) in making available a discourse referent corresponding to a mouse:

(116) **[context], [nobody-see-a-mouse2]** $\vdash$

$$[\mathit{a}(Y, \mathit{mouse}(Y), \mathit{no}(X, \mathit{person}(X), \mathit{see}(X,Y))), \langle Y\rangle] : f_\sigma \otimes \langle h_\sigma\rangle$$

We have shown that the glue approach is valuable not only in accounting for meaning contributions in the derivation of the meaning of a sentence, but also in managing contextual contributions across sentences in a discourse. The representation of context and the glue-theoretic treatment of anaphora resolution will be of crucial importance in our treatment of *anaphoric control* in Chapter 12. Elsewhere, however, when context and anaphora resolution are not relevant to the syntactic binding conditions. Antecedency conditions for some pronouns depend on information-structural properties of the sentence or discourse; Culy (2000) discusses *topic anaphora*, distinguishing among anaphors that refer to the topic of the sentence, paragraph, or story. Other conditions on pronoun antecedency are also found: a *logophoric pronoun* is one which is used to refer to "the author of a discourse or to a participant whose thoughts are reported" (Hagége 1974, translation by Stirling 1988). Bresnan (2001b, Chapter 11) discusses logophoricity and constraints on logophoric pronouns, and Culy et al. (1995) examine the pronominal systems of three Dogon languages and trace their evolution from a common ancestor, showing how logophoric pronouns can evolve in the course of language change.

Related to systems of pronominal binding, coreference, and logophoricity are systems of *obviation* and *switch reference*, where certain antecedents are allowed or disallowed for anaphors in certain syntactic positions. Simpson and Bresnan (1983) analyze control and obviation in Warlpiri; their analysis is reviewed by Dalrymple (1993), who provides a reanalysis using inside-out functional uncertainty.

12

FUNCTIONAL AND ANAPHORIC CONTROL

This chapter explores the syntax and semantics of functional and anaphoric control, constructions in which either syntactic or lexical constraints require coreference between an argument of the matrix clause and an argument of a subordinate or modifying adjunct clause. In English, such cases include the classes of "equi" and "raising" verbs. Crosslinguistically, descriptions of such constructions involve reference to functional syntactic properties such as term argument, SUBJ, OBJ, and so on; therefore, the syntactic discussion in the following is primarily centered around the f-structures of functional and anaphoric control constructions.

The open grammatical functions XCOMP and XADJ and the closed function COMP were first introduced by Bresnan (1982a) in a pioneering study of clausal relations and complementation. In the following, we will review this work and make some new proposals for the syntactic and semantic treatment of functional and anaphoric control constructions.

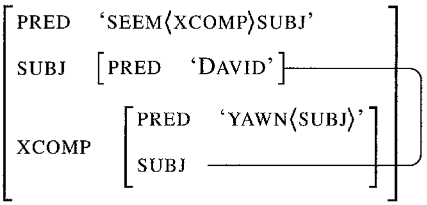
1. OPEN COMPLEMENTS AND FUNCTIONAL CONTROL

As an illustration of functional control, we first examine "raising" verbs,[1] verbs like *seem* in example (1):

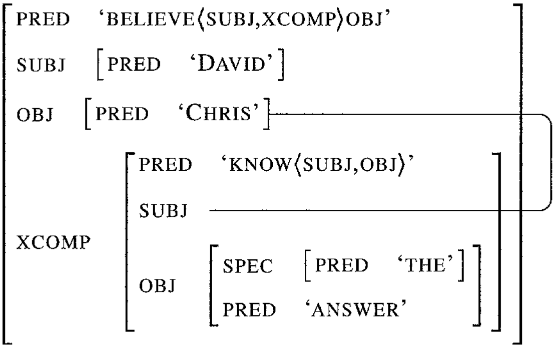
(1) *David seemed to yawn.*

Raising verbs are distinguished by the fact that the "raised" argument, the SUBJ *David* in example (1), is not a semantic argument of the raising verb. In other words, raising verbs impose no semantic constraints on the raised argument. Notationally, this is indicated by the position of the raised argument outside of the angled brackets in the semantic form of the raising verb, as discussed in Chapter 2, Section 3.6.1:

(2) *David seemed to yawn.*



(3) *David believed Chris to know the answer.*



Raising verbs in English and other languages exemplify *functional control*. Functional control verbs require as an argument an open complement XCOMP. In (2), the SUBJ of the raising verb *functionally controls* the SUBJ of the subordinate XCOMP. This means that the SUBJ of the verb *seemed* is required to be the same f-structure as the SUBJ of the subordinate XCOMP, as shown in (2). Other raising

[1] "Raising" verbs are so called because of their analysis in transformational grammar (Kiparsky and Kiparsky 1970; Postal 1974), in which the subject phrase of the subordinate clause was assumed to have raised, or moved up, from the subordinate clause to its final position in the matrix clause.

verbs exhibit functional control by an OBJ, as shown in example (3), where the OBJ of the matrix verb is the same as the SUBJ of the XCOMP.

1.1. Evidence for Functional Control

In constructions involving functional control, in which the same argument is both an argument of the matrix verb and the SUBJ of the subordinate XCOMP, any syntactic restrictions that are imposed on the SUBJ in the subordinate clause must also hold for the "raised" argument, since the same f-structure appears in both the matrix and subordinate clause.

Some English predicates require a semantically empty subject with a particular syntactic form. For example, the verb *rain* requires its subject to have the form *it*, not *there*, as shown in example (4).

(4) a. *It is raining.*

b. *There is a problem.*

The f-structure for example (4a) is:

(5)
$$\begin{bmatrix} \text{PRED} & \text{`RAIN}\langle\rangle\text{SUBJ'} \\ \text{SUBJ} & \begin{bmatrix}\text{FORM} & \text{IT}\end{bmatrix} \end{bmatrix}$$

Since only semantic constraints imposed by the XCOMP of a raising verb need be observed, we expect expletive arguments, arguments with no semantic content, to appear felicitously in a raising construction (Postal 1974; Bresnan 1982a,c). Syntactic requirements on the form of the "raised" argument must also be met in the functional control construction, since both matrix and subordinate clause requirements must be satisfied:

(6) a. *It seems to be raining.*

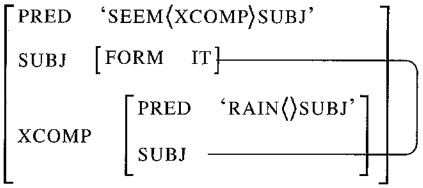
b. *There seems to be a problem.*

(7) a. *David believed it to be raining.*

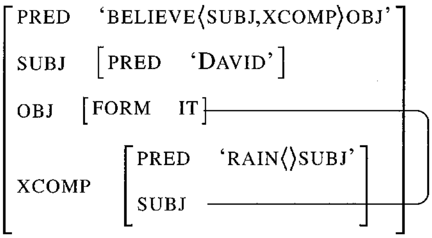
b. *David believed there to be a problem.*

In examples (6) and (7), the "raised" argument/controller is not a semantic argument either of the matrix or the subordinate predicate; syntactic requirements imposed by the subordinate clause verb are satisfied, no semantic requirements are violated, and the examples are wellformed. The f-structures for examples (6a) and (7a) are:

(8) a. *It seems to be raining.*



b. *David believed it to be raining.*



Case requirements imposed by both matrix and subordinate clause verbs must also be satisfied. Andrews (1982) discusses raising and functional control in Icelandic, exploring the behavior of "quirky case" verbs, verbs that lexically specify a particular case for their arguments. As discussed in Chapter 2, Section 4.1, Icelandic subjects of quirky case verbs can be marked with one of several cases, depending on requirements imposed by the verb:

(9) a. Accusative subject:

Drengina vantar mat.
boys.DEF.ACC lacks food.ACC
'The boys lack food.'

b. Dative subject:

Barninu batnaði veikin.
child.DEF.DAT recovered.from disease.DEF.NOM
'The child recovered from the disease.'

c. Genitive subject:

Verkjanna gætir ekki.
pains.DEF.GEN is.noticeable not
'The pains are not noticeable.'

Andrews shows that in a functional control construction, the case of the "raised" OBJ depends on the casemarking requirements on the SUBJ of the lower clause:

(10) a. Accusative "raised" object:

Hann telur mig (í barnaskap sínum) vanta peninga.
he believes me.ACC (in foolishness his) to.lack money.ACC
'He believes me (in his foolishness) to lack money.'

b. Dative "raised" object:

Hann telur barninu (í barnaskap sínum) hafa
he believes child.DEF.DAT (in foolishness his) to.have
batnað veikin.
recovered.from disease.DEF.NOM
'He believes the child (in his foolishness) to have recovered from the disease.'

c. Genitive "raised" object:

Hann telur verkjanna (í barnaskap sínum) ekki gæta.
he believes pains.DEF.GEN (in foolishness his) not noticeable
'He believes the pains (in his foolishness) not to be noticeable.'

The position of the parenthesized adjunct *í barnaskap sínum* 'in his foolishness', which is a matrix clause modifier, shows that the "raised" constituent does indeed appear as the OBJ of the matrix clause and not as a constituent of the subordinate clause. Since this argument is also the SUBJ of the subordinate XCOMP, the constraints on casemarking that the subordinate XCOMP verb imposes on this argument must be met for the sentence to be wellformed. Andrews (1990a) and Zaenen and Maling (1990) provide more discussion of default case, quirky case, and raising verbs in Icelandic.

Some syntactic tests discussed by Jacobson (1990, 1992) also demonstrate the syntactic characteristics of the open complement XCOMP in English. First, VP complement drop is never possible for the open complement XCOMP:

(11) a. *[Did David really yawn?] He seemed* $\left\{\begin{array}{l} \textit{to (yawn).} \\ *\emptyset \end{array}\right.$

b. *[Did Chris really know the answer?]*

David believed him $\left\{\begin{array}{l} \textit{to (know the answer).} \\ *\emptyset \quad \text{(wrong meaning)} \end{array}\right.$

Second, XCOMP is not among the syntactic categories that can appear in sentence-initial position in a long-distance dependency in English:

(12) a. **To yawn, David seemed.*

b. **To yawn, David believed Chris.*

As we will see in Section 3.2, the closed complement COMP behaves differently in each of these respects.

1.2. Constituent Structure and Functional Constraints

We propose the following annotated phrase structure rule for functional control constructions in English (only details relevant to this construction are displayed):

(13) $$\text{V}' \longrightarrow \begin{pmatrix}\text{V}\\ \uparrow=\downarrow\end{pmatrix} \begin{pmatrix}\text{NP}\\ (\uparrow\ \textsc{obj})=\downarrow\end{pmatrix} \begin{pmatrix}\text{VP}\\ (\uparrow\ \textsc{xcomp})=\downarrow\end{pmatrix}$$

This rule defines the constituent structure of functional control constructions and constrains the functional syntactic role of each constituent. Notice that the phrase structure rule does not specify the control relation between the matrix argument and the SUBJ of the XCOMP: the difference between verbs like *seemed*, whose SUBJ is also the SUBJ of its XCOMP, and *believed*, whose OBJ is the SUBJ of its XCOMP, is lexically specified, not given by constituent structure requirements.

The lexical entries for the English raising verbs *seemed* and *believed* contain the following syntactic information:

(14) *seemed* V (↑ PRED) = 'SEEM⟨XCOMP⟩SUBJ'
(↑ SUBJ) = (↑ XCOMP SUBJ)

(15) *believed* V (↑ PRED) = 'BELIEVE⟨SUBJ,XCOMP⟩OBJ'
(↑ OBJ) = (↑ XCOMP SUBJ)

These lexical entries contain a *control equation* specifying the relation between an argument of the matrix clause and the SUBJ of the subordinate XCOMP. The control equations for the above verbs are:

(16) *seemed* (↑ SUBJ) = (↑ XCOMP SUBJ)
believed (↑ OBJ) = (↑ XCOMP SUBJ)

These lexical entries and phrase structure rules give rise to the c-structures and f-structures shown in (17) (page 319) and (19) (page 320) for examples (2–3).[2] The theory of controller selection and how the control equation is determined is discussed in Section 6 of this chapter.

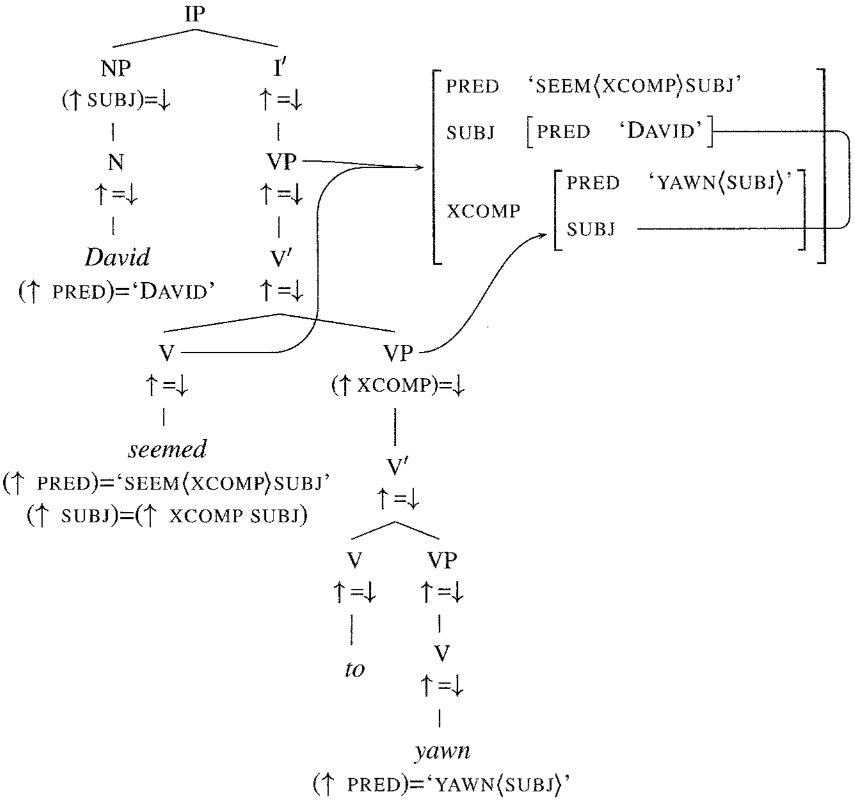
2. RAISING VERBS AND SEMANTIC COMPOSITION

2.1. Semantics of Raising Verbs

The semantic contribution of raising verbs like *seem* and *tend* has been widely studied. We propose the representation in (18) of the sentence *David seemed to yawn*:

[2] We follow Bresnan (2001b) in classifying infinitival *to* as the head of a verbal projection. However, see Falk (2001) for evidence that *to* in fact appears in C, as the head of CP.

(17) *David seemed to yawn.*



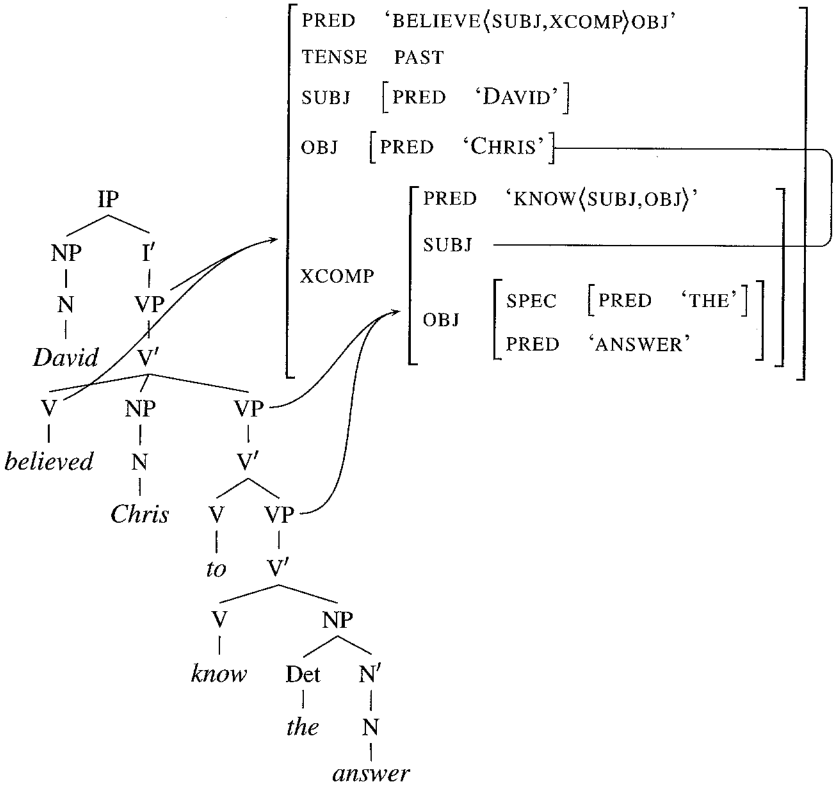
(18) *David seemed to yawn.*

seem(*yawn*(*David*))

In this example, the predicate *seem* holds of the proposition *yawn*(*David*), and the sentence has more or less the same meaning as the sentence *It seemed that David yawned.*

It has often been noted that scope ambiguities are available with the SUBJ argument of a subject raising verb like *seem* and with the OBJ argument of an object raising verb like *believe* (May 1977; Williams 1983; Halvorsen 1983; Jacobson 1990): a raising verb allows a narrow scope reading for its "raised" argument. For example, a sentence like *Someone seemed to yawn* has two readings, a narrow scope reading paraphrasable as *It seemed that someone yawned* and a wide scope reading where some particular person seemed to yawn, as shown in (20).

(19) *David believed Chris to know the answer.*



(20) *Someone seemed to yawn.*

narrow scope interpretation = *It seemed that someone yawned.*
wide scope interpretation = *There is someone who seemed to yawn.*

This ambiguity depends on the scope of *someone*, whether inside or outside *seem*:

(21) *Someone seemed to yawn.*

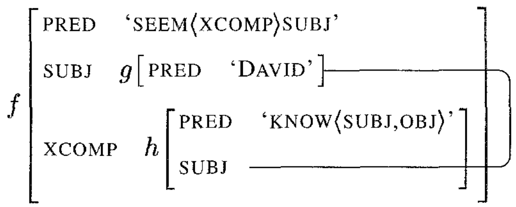
Narrow scope: $\mathit{seem}(\mathit{a}(X, \mathit{person}(X), \mathit{yawn}(X)))$
Wide scope: $\mathit{a}(X, \mathit{person}(X), \mathit{seem}(\mathit{yawn}(X)))$

In the next section, we will see how both readings for this sentence are derived.

2.2. Raising Verbs and Meaning Assembly

The sentence *David seemed to yawn* is associated with the f-structure and meaning constructor in (22):

(22) *David seemed to yawn.*



$\mathit{seem}(\mathit{yawn}(\mathit{David})) : f_\sigma$

Since our focus is not on anaphoric binding or context update, we provide simple noncontextual meaning constructors for example (22) rather than the context-meaning pair constructors presented in Chapter 11. To provide context-meaning pair constructors for this example, we would augment the context with the discourse referent *David* and its corresponding semantic structure g_σ, as described in Chapter 11.

The presence of the SUBJ argument outside the angled brackets in the semantic form in (22) indicates that the SUBJ of *seem* is not a semantic argument of the verb and that the sole semantic argument is the XCOMP. In keeping with this intuition, we follow Asudeh (2000a, 2001b) in providing the following lexical entry for the verb *seemed*:

(23) *seemed* (↑ PRED) = 'SEEM⟨XCOMP⟩SUBJ'
(↑ SUBJ) = (↑ XCOMP SUBJ)
$\lambda P.\mathit{seem}(P) : (\uparrow \text{XCOMP})_\sigma \multimap \uparrow_\sigma$

The meaning constructor in the third line of this lexical entry requires an XCOMP argument. A meaning contribution corresponding to the SUBJ is not required, since the SUBJ meaning is not a semantic argument of *seem*. If the SUBJ contributes a meaning resource, it must be consumed by the XCOMP verb for the sentence to be semantically complete and coherent.

We now instantiate the meaning constructor given in (23) according to the labels in (22). We also provide instantiated meaning constructors for the proper name *David* and the intransitive verb *yawn*:

(24) Meaning constructor premises for *David seemed to yawn*:

[seem] $\lambda P.\mathit{seem}(P) : h_\sigma \multimap f_\sigma$

[David] $\mathit{David} : g_\sigma$

[yawn] $\lambda X.\mathit{yawn}(X) : g_\sigma \multimap h_\sigma$

The meaning constructor labeled **[yawn]** requires a meaning for its subject, g_σ, to produce a meaning for h_σ. Thus, we first combine the meaning constructors **[David]** and **[yawn]** to produce the meaning constructor **[David-yawn]** for h_σ:

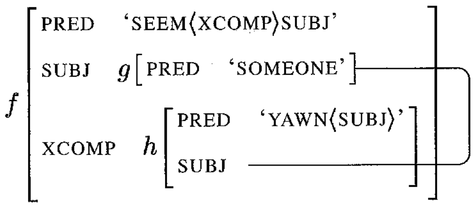
(25) **[David-yawn]** $\mathit{yawn}(\mathit{David}) : h_\sigma$

A meaning for h_σ is exactly what the meaning constructor **[seem]** requires, and a semantically complete and coherent meaning constructor for the sentence results:

(26) **[David-yawn], [seem]** $\vdash \mathit{seem}(\mathit{yawn}(\mathit{David})) : f_\sigma$

Meaning deduction from the premises contributed by the sentence *Someone seemed to yawn* yields two different conclusions, since the example is ambiguous with both a narrow and a wide scope reading:

(27) *Someone seemed to yawn.*



Narrow scope: $\mathit{seem}(a(X, \mathit{person}(X), \mathit{yawn}(X))) : f_\sigma$
Wide scope: $a(X, \mathit{person}(X), \mathit{seem}(\mathit{yawn}(X))) : f_\sigma$

The meaning constructor premises in (28) are relevant for this example. Again, since we are examining the semantics of raising verbs and are not concentrating on issues related to context update and anaphoric binding, we provide the simple meaning constructor discussed in Chapter 9, Section 8, rather than the context-meaning pair constructor discussed in Chapter 11, Section 3.6, for the quantifier *someone*:

(28) Meaning constructor premises for *Someone seemed to yawn*:

[seem]	$\lambda P.\mathit{seem}(P)$	$: h_\sigma \multimap f_\sigma$
[someone]	$\lambda S.a(X, \mathit{person}(X), S(X))$	$: \forall H.[g_\sigma \multimap H] \multimap H$
[yawn]	$\lambda X.\mathit{yawn}(X)$	$: g_\sigma \multimap h_\sigma$

For the narrow scope reading, we note that the quantifier meaning constructor **[someone]** requires as its argument a resource of the form $g_\sigma \multimap H$ for some semantic structure H. The meaning constructor **[yawn]** provides such a resource. Combining **[someone]** and **[yawn]**, we have the meaning constructor labeled **[someone-yawn]** in (29):

(29) **[someone-yawn]** $a(X, \mathit{person}(X), \mathit{yawn}(X)) : h_\sigma$

The meaning constructor in (29) provides a meaning resource h_σ, exactly what the meaning constructor labeled **[seem]** in (28) requires. Combining **[someone-yawn]** and **[seem]**, we obtain the semantically complete and coherent meaning constructor in (30), yielding the narrow scope reading:

(30) **[someone-yawn]**, **[seem]** $\vdash$ $\textit{seem}(a(X, \textit{person}(X), \textit{yawn}(X))) : f_\sigma$

To derive the wide scope reading for the example, we will make use of the abstraction rule given in example (36) of Chapter 9. Recall that this rule permits the introduction of a hypothetical premise on the glue side, which is discharged at a later point in the deduction; on the meaning side, hypothetical premise discharge corresponds to abstracting over the variable introduced by the premise. For this example, we hypothesize the premise $X : [g_\sigma]$ in the first line of (31):

(31)

$$\cfrac{\cfrac{\cfrac{X : [g_\sigma] \qquad [\textbf{yawn}]}{\textit{yawn}(X) : h_\sigma} \qquad\qquad [\textbf{seem}]}{\textit{seem}(\textit{yawn}(X)) : f_\sigma}}{[\textbf{seem-yawn}] \quad \lambda X.\textit{seem}(\textit{yawn}(X)) : g_\sigma \multimap f_\sigma}$$

We combine the hypothesized premise $X : [g_\sigma]$ with the premise **[yawn]**, producing the meaning constructor $\textit{yawn}(X) : h_\sigma$. This meaning constructor provides the semantic resource h_σ required by **[seem]**. Combining these two meaning constructors, we produce the meaning constructor $\textit{seem}(\textit{yawn}(X)) : f_\sigma$. Finally, the hypothesized premise $X : [g_\sigma]$ is discharged in the last line of (31), producing the meaning constructor labeled **[seem-yawn]**: the variable X is abstracted over on the left-hand side, producing the function $\lambda X.\textit{seem}(\textit{yawn}(X))$, and the implicational meaning constructor $g_\sigma \multimap f_\sigma$ is produced on the right-hand side.

The meaning constructor **[seem-yawn]** provides a resource of the form $g_\sigma \multimap H$, which is what the quantifier **[someone]** requires; for this reading, the semantic structure f_σ is chosen to provide the scope meaning of the quantifier. Combining the meaning constructors **[someone]** and **[seem-yawn]**, we obtain the semantically complete and coherent meaning constructor in (32), which provides the wide-scope reading for this example:

(32) **[seem-yawn]**, **[someone]** $\vdash$ $a(X, \textit{person}(X), \textit{seem}(\textit{yawn}(X))) : f_\sigma$

3. CLOSED COMPLEMENTS AND ANAPHORIC CONTROL

Anaphoric control contrasts with functional control in several interlinked ways. The subordinate complement in an anaphoric control construction is the closed function COMP, not the open function XCOMP. Some constraints on the controller in an anaphoric control construction are similar to those in functional control, but the nature of the control is different: the relation in anaphoric control is semantically much closer to an anaphoric binding relation and does not involve syntactic identity.

Anaphoric control constructions are of two types, obligatory anaphoric control and arbitrary anaphoric control (Bresnan 1982a; Zec 1987; Bresnan 2001b). In an obligatory anaphoric control construction, coreference is required between an argument of the matrix clause and the controlled position in the subordinate clause. In contrast, in an arbitrary anaphoric control construction, no coreference constraints are imposed by the control verb. Instead, the controlled argument in the subordinate clause finds its referent in a way very similar to an ordinary pronoun, and split antecedents and syntactically remote controllers are possible.

3.1. Obligatory Anaphoric Control

Obligatory anaphoric control constructions were first examined in LFG by Bresnan (1982a), who showed that an anaphor in an anaphoric control construction may be assigned an antecedent by the rules of sentence grammar. Further work exploring obligatory anaphoric control in Serbo-Croatian was done by Zec (1987). In the following, we explore the syntax of obligatory anaphoric control; the semantics of these constructions will be discussed in Section 4 of this chapter.

English equi verbs[3] exemplify obligatory anaphoric control of the SUBJ of a closed complement COMP:

(33) *David tried to leave.*

$$\begin{bmatrix} \text{PRED} & \text{`TRY}\langle\text{SUBJ,COMP}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`DAVID'} \end{bmatrix} \\ \text{COMP} & \begin{bmatrix} \text{PRED} & \text{`LEAVE}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

In example (33), the SUBJ of the obligatory anaphoric control verb *tried* anaphorically controls the SUBJ of the COMP, and the sentence is interpreted as meaning that David tried to bring about a situation where David leaves. The controller in an anaphoric control construction can also be a matrix clause object:

[3] "Equi" verbs are so called because of their participation in the "Equi-NP Deletion" transformation proposed for them by Postal (1972) and others in a transformational framework, in which an NP in subject position of the subordinate clause was assumed to be deleted under conditions of identity with an NP in the matrix clause.

(34) *David convinced Chris to leave.*

$$\begin{bmatrix} \text{PRED} & \text{`CONVINCE}\langle\text{SUBJ,OBJ,COMP}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`DAVID'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{`CHRIS'} \end{bmatrix} \\ \text{COMP} & \begin{bmatrix} \text{PRED} & \text{`LEAVE}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Here, the controller of the SUBJ of *leave* is the OBJ of the matrix verb, *Chris*, and the sentence means that David convinced Chris that Chris should leave. We will discuss syntactic and semantic constraints on the controller in Section 6 of this chapter.

3.2. Anaphoric vs. Functional Control

In an anaphoric control construction, the anaphorically controlled SUBJ of the subordinate COMP is syntactically independent from the matrix controller, although the two are semantically related by an anaphoric binding relation. Thus, unlike the situation with functional control, we do not expect syntactic restrictions imposed on the subject of the COMP to be relevant for the matrix clause controller. As Andrews (1982) shows for anaphoric control in Icelandic, the case restrictions found in Icelandic functional control constructions are not found in constructions involving anaphoric control.

As shown in example (9) of this chapter, subjects in Icelandic can bear a case that is idiosyncratically assigned by the verb. In constructions involving functional control, the case requirements of the subordinate clause verb must be satisfied, as shown in example (10) of this chapter. In contrast, in anaphoric control constructions in Icelandic, case requirements of the subordinate clause verb do not apply to the matrix controller. Example (35a) shows that the Icelandic verb *vanta* 'lack' requires an accusative SUBJ; in (35b), the subject of the verb *vanta* 'lack' is interpreted as coreferent with the subject of the equi verb *vonast* 'hope'. The subject of the control verb *vonast* 'hope' is marked with NOM case, not ACC, even though the subordinate verb *vanta* 'lack' specifies that its subject should be marked with ACC case:

(35) a. *Drengina vantar mat.*
boys.DEF.ACC lacks food.ACC
'The boys lack food.'

b. *Ég vonast til að vanta ekki efni í ritgerðina.*
I.NOM hope to to lack not material for thesis.DEF
'I hope to not lack material for the thesis.'

This contrasts with the situation with functional control, illustrated in (10) of this chapter, where the case specified by the subordinate clause verb must appear on the matrix subject.

Further evidence for the difference between anaphoric and functional control in English is discussed by Jacobson (1990, 1992) and, in connection with raising verbs, in Section 1.1 of this chapter. VP complement drop is a lexically governed option, impossible for the open function XCOMP, as shown in example (11) of this chapter, but possible for the closed COMP argument of many predicates:

(36) a. *[Did David really leave?] He tried.*
b. *[Will Chris leave?] If David can convince him.*

Evidence from topicalization is less clear. Jacobson (1990, 1992) claims that while the complement of a raising verb cannot appear at the beginning of the sentence, as shown in example (12) of this chapter, examples in which the infinitival complement of an equi verb appears in initial position are marginally acceptable:

(37) a. *?To leave, David (at least) tried.*
b. *?To leave, David (at least) convinced Chris.*

In fact, these examples are not acceptable for all speakers. Nevertheless, there does seem to be a contrast between the examples with sentence-initial infinitives in (37) and those in (12). The relatively low acceptability of the examples in (37) may in part be due to the unsuitability of the infinitive in the pragmatic/information structure role associated with the sentence-initial phrase (Tracy Holloway King, p.c.).

3.3. Constituent Structure and Functional Constraints

To analyze anaphoric control, we augment the phrase structure rule in (13) to allow for VP complement daughters bearing the COMP function:

(38)
$$\mathrm{V}' \longrightarrow \left(\begin{array}{c}\mathrm{V}\\ \uparrow=\downarrow\end{array}\right) \left(\begin{array}{c}\mathrm{NP}\\ (\uparrow \textsc{obj})=\downarrow\end{array}\right) \left(\begin{array}{c}\mathrm{VP}\\ (\uparrow \{\textsc{xcomp}|\ \textsc{comp}\})=\downarrow\end{array}\right)$$

The rule in (38) differs from the one in (13) in allowing the VP daughter of V′ to bear either the XCOMP or the COMP function. There is no constituent structure distinction between VP complements that are functionally controlled, bearing the XCOMP function, and those that are anaphorically controlled and bear the COMP function; XCOMP and anaphorically controlled COMP appear in the same position relative to adverbs and direct objects, for example:

(39) a. *The students seem clearly to be intelligent.* (XCOMP)
b. *The students tried hard to be on time.* (COMP)

(40) a. *The students believed David to have left.* (XCOMP)
b. *The students convinced David to leave.* (COMP)

Thus, it is only the functional annotations on the rule that distinguish the two cases.

We propose the following lexical entries for the English equi verbs *tried* and *convinced*:

(41) *tried* V (↑ PRED) = 'TRY⟨SUBJ,COMP⟩'
(↑ COMP SUBJ PRED) = 'PRO'

(42) *convinced* V (↑ PRED) = 'CONVINCE⟨SUBJ,OBJ,COMP⟩'
(↑ COMP SUBJ PRED) = 'PRO'

The second line in each of these lexical entries specifies a pronominal SUBJ for the subordinate COMP. Section 4 of this chapter discusses how the anaphoric control relation is established between the matrix controller and the subordinate clause SUBJ.

These lexical entries, together with the annotated phrase structure rule in (38), give rise to the anaphoric control structures in (44) (page 328) and (46) (page 329).

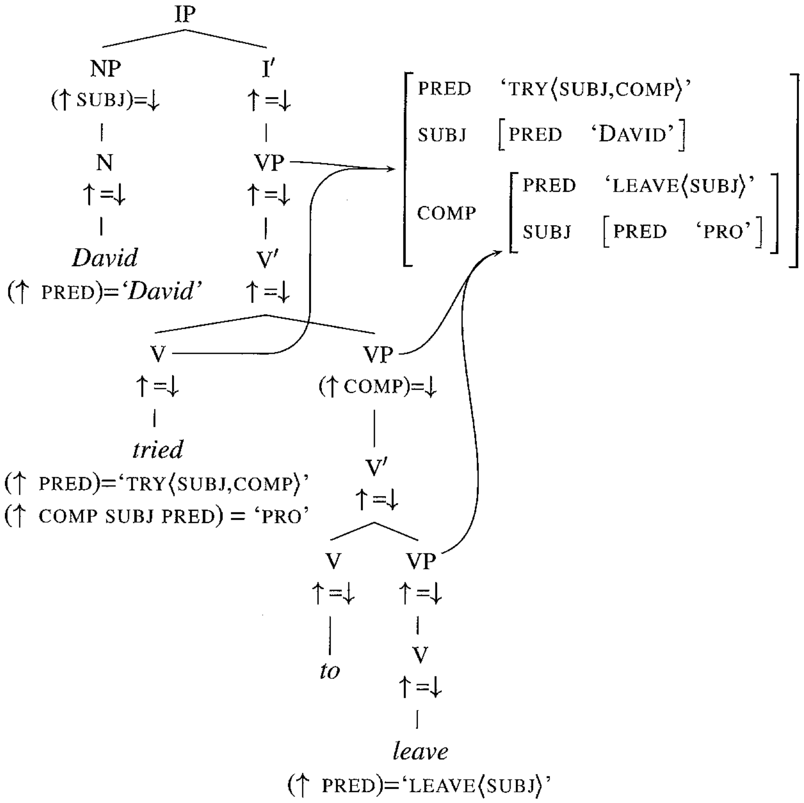
3.4. Equi Verbs and Control

We have seen that English equi verbs exemplify anaphoric control, in contrast with English raising verbs, which exhibit functional control. Equi and raising verbs differ in other ways as well: unlike raising verbs, the controller in an equi construction is semantically as well as syntactically selected by the verb. Notationally, this is reflected in the fact that the SUBJ of the equi verb *try* appears inside rather than outside the angled brackets in the semantic form (see Chapter 2, Section 3.6.1):

(43)
$$\begin{bmatrix} \text{PRED} & \text{`TRY}\langle\text{SUBJ,COMP}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`DAVID'}\end{bmatrix} \\ \text{COMP} & \begin{bmatrix} \text{PRED} & \text{`LEAVE}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{`PRO'}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

Since the controller is semantically selected, equi verbs cannot appear with a subordinate COMP verb that selects an expletive or semantically empty subject, since

(44) *David tried to leave.*

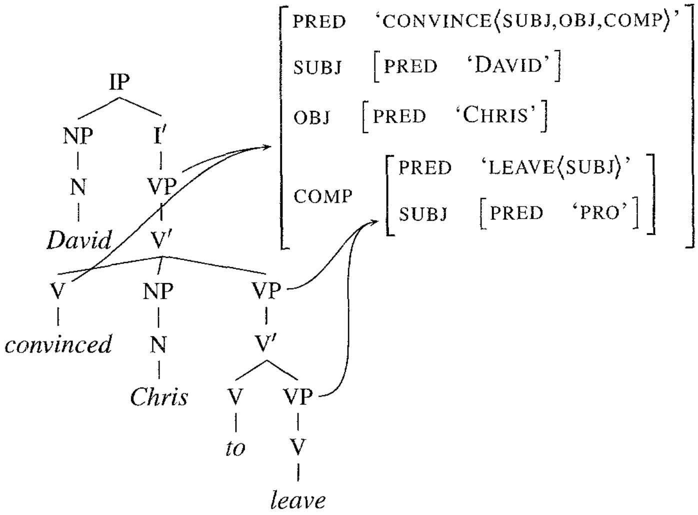


expletive subjects contribute no semantic content and therefore cannot enter into an anaphoric dependency with a controller in the higher clause:

(45) **There tried to be a problem.*

In principle, equi verbs can participate in either functional or anaphoric control, and it is sometimes assumed that equi constructions as well as raising constructions in English involve functional control (see, for example, Bresnan 1982a, 2001b; Asudeh 2000a, 2001b). The evidence presented above indicates that English equi verbs in fact participate in obligatory anaphoric control. However, examination of a larger range of languages shows that more variation is found: some languages have two types of equi verbs, some specifying anaphoric control and some specifying functional control. Falk (2001) provides an illuminating discus-

(46) *David convinced Chris to leave.*

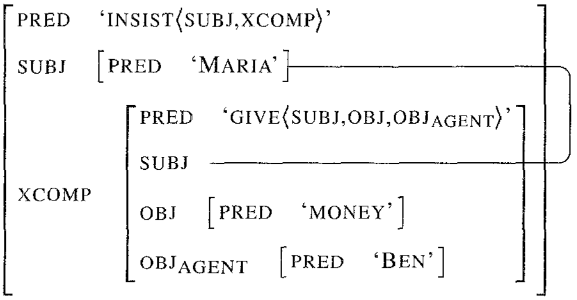


sion of the difference between anaphoric and functional control in equi constructions, though the conclusions he draws are different from those just presented.

Kroeger (1993, Chapter 4) shows that Tagalog has two different types of equi constructions. The first type involves functional control of a subject argument in the complement clause:

(47)	*nagpilit*	*si-Maria-ng*	*bigy-an*	*ng-pera*	*ni-Ben*
	PERFECT.ACTIVE.insist.on	NOM-Maria-COMP	give-DAT	GEN-money	GEN-Ben

‘Maria insisted on being given money by Ben.’



The second type involves anaphoric control of a term (possibly nonsubject) argument in the complement clause. In example (48), the matrix subject *Maria* anaphorically controls the OBJ$_{\text{AGENT}}$ argument in the subordinate clause:

(48)

nagpilit	*si-Maria-ng*	*bigy-an*	*ng-pera*	*si-Ben*
PERFECT.ACTIVE.insist.on	NOM-Maria-COMP	give-DAT	GEN-money	NOM-Ben

'Maria insisted on giving money to Ben.'

$$\left[\begin{array}{ll}
\text{PRED} & \text{`INSIST}\langle\text{SUBJ,COMP}\rangle\text{'}\\
\text{SUBJ} & \left[\text{PRED}\quad \text{`MARIA'}\right]\\
\text{COMP} & \left[\begin{array}{ll}
\text{PRED} & \text{`GIVE}\langle\text{SUBJ,OBJ,OBJ}_{\text{AGENT}}\rangle\text{'}\\
\text{SUBJ} & \left[\text{PRED}\quad \text{`BEN'}\right]\\
\text{OBJ} & \left[\text{PRED}\quad \text{`MONEY'}\right]\\
\text{OBJ}_{\text{AGENT}} & \left[\text{PRED}\quad \text{`PRO'}\right]
\end{array}\right]
\end{array}\right]$$

Thus, equi verbs involving functional as well as anaphoric control can be found, even within the same language.

4. EQUI VERBS AND SEMANTIC COMPOSITION

4.1. Semantics of Equi Verbs

Unlike raising verbs, equi verbs semantically as well as syntactically select for their controller argument. Thus, we treat an equi verb like *try* as denoting a two-place predicate, following Halvorsen (1983), Reyle (1988), and many others. The meaning for the sentence *David tried to leave* is shown in (49):

(49) *David tried to leave.*

$\textit{try}(\textit{David}, \textit{leave}(\textit{David}))$

In (49), *try* denotes a relation between an individual *David* and the proposition *leave*(*David*) that the individual is trying to bring about. The equi verb *tried* requires the pronominal subject of the subordinate verb *leave* to be coreferent with the controller subject of *tried*, *David*.

Asudeh (2000a, 2001b) proposes an analysis of English equi verbs within the glue framework that differs from the analysis presented here in its assumptions about the meaning of the complement of the equi verb. Following Chierchia (1984, 1985), Asudeh claims that the XCOMP complement of an equi verb denotes a property rather than a proposition. On this view, the meaning of a sentence like *David tried to leave* is represented as:

(50) *David tried to leave.*

$\textit{try}(\textit{David}, \lambda X.\textit{leave}(X))$ (Chierchia 1984, 1985; Asudeh 2000a, 2001b)

In (50), *try* denotes a relation between an individual and the property that the individual aspires to have. Chierchia (1984) argues for the property theory of control on the basis of entailments like the one shown in (51):

(51) *Nando tries anything/whatever Ezio tries.*
Ezio tries to jog at sunrise.
Nando tries to jog at sunrise.

On Chierchia's view, the validity of this argument is reflected in the following logical paraphrases:

(52)
$$\begin{array}{l} \forall X.\mathit{try}(\mathit{Ezio}, X) \rightarrow \mathit{try}(\mathit{Nando}, X) \\ \mathit{try}(\mathit{Ezio}, \lambda X.\mathit{jog\ at\ sunrise}(X)) \\ \hline \mathit{try}(\mathit{Nando}, \lambda X.\mathit{jog\ at\ sunrise}(X)) \end{array}$$

Chierchia argues that if the complement of the verb *try* is treated as a proposition roughly corresponding to the meaning of a sentence like *Ezio jogs at sunrise*, an unwanted entailment seems to follow, namely that Nando tries for Ezio to jog at sunrise:

(53)
$$\begin{array}{l} \forall X.\mathit{try}(\mathit{Ezio}, X) \rightarrow \mathit{try}(\mathit{Nando}, X) \\ \mathit{try}(\mathit{Ezio}, \mathit{jog\ at\ sunrise}(\mathit{Ezio})) \\ \hline \mathit{try}(\mathit{Nando}, \mathit{jog\ at\ sunrise}(\mathit{Ezio})) \end{array}$$

However, there are several difficulties with the property theory of control advocated by Chierchia and Asudeh.

First, Chierchia's proposal is based on the lack of availability of a strict reading for the conclusion of the argument in (51): the conclusion means that Nando tries for Nando (the sloppy reading) and not Ezio (the strict reading) to jog at sunrise.[4] However, other cases of sloppy-only readings for arguments similar to (51) are not amenable to a solution that, like Chierchia's, relies on the property theory of control. For example, the argument in (54) is also valid:

(54) *Nando does anything/whatever Ezio does.*
Ezio$_i$ *broke his*$_i$ *arm playing football.*
Nando$_j$ *broke his*$_j$ *arm playing football.*

However, there is no obvious way in which the canonical representation of the meaning of the sentence *Ezio broke his arm playing football* can be adjusted to predict the validity of the entailment in (54). Thus, whatever means accounts for

[4] The distinction between *strict* and *sloppy* readings is best known from analyses of ellipsis (see Dalrymple et al. 1991 and references cited there); in the following, (b) paraphrases the sloppy reading of (a), and (c) paraphrases the strict reading:

(a) *David rode his bike, and Chris did too.*
(b) *Chris rode Chris's bike.* (sloppy)
(c) *Chris rode David's bike.* (strict)

the sloppy-reading entailment in (54) can presumably account for the validity of the argument in (51) without assuming the property analysis of control.

Additionally, Higginbotham (1992) points out that in at least some cases, arguments analogous to the one in (51) do not provide evidence either for or against Chierchia's position. Higginbotham argues that the entailment in (55) should not be taken as a linguistic fact about the verb *practice*:

(55) *Nando does anything/whatever Ezio does.*
Ezio practices playing the piano.

Nando practices playing the piano.

Coreference between the subject of the verb *practice* and the understood subject of the gerund *playing* is enforced by real-world constraints on situations of practicing the piano. Regardless of whether the gerund complement of *practice* is taken to be a property or a proposition, it makes no sense to talk about Nando practicing Ezio's playing the piano. Therefore, Higginbotham argues, examples like this do not shed light on the issue of the type of the complement of equi verbs.

Higginbotham (1992) also discusses examples such as (56):

(56) *They expected to sit next to each other.*

As Higginbotham points out, this sentence has two readings, paraphrasable in the following way:

(57) a. *Each of them expects to sit next to the other one.*

b. *They expect that they will sit next to each other.*

On the proposition theory of control adopted here, both of these readings are readily available (we use the notation of Dalrymple et al. 1997a to represent reciprocal meaning):

(58) a. *Each of them expects to sit next to the other one:*

$\mathit{RECIP}(\mathit{they}, \lambda X.\lambda Y.\mathit{expect}(X, \mathit{sit\ next\ to}(X, Y)))$

b. *They expect that they will sit next to each other:*

$\mathit{expect}(\mathit{they}, \mathit{RECIP}(\mathit{they}, \lambda X.\lambda Y.\mathit{sit\ next\ to}(X, Y)))$

The reading paraphrased in (57a) is also readily available on the property theory:

(59) *Each of them expects to sit next to the other one:*

$\mathit{RECIP}(\mathit{they}, \lambda X.\lambda Y.\mathit{expect}(X, \lambda Z.\mathit{sit\ next\ to}(Z, Y)))$

However, the reading paraphrased in (57b) is difficult to account for on the property theory. It might be thought that the representation in (60) corresponds to the desired interpretation:

(60) *They expect that they will sit next to each other:*

$$\textit{expect}(\textit{they}, \lambda X.\textit{RECIP}(X, \lambda Z.\lambda Y.\textit{sit next to}(Z, Y)))$$

The problem with this representation is that a predicate like *expect* denotes a relation between an individual and the property that the individual expects to have. However, an individual cannot enter into a relationship involving a *RECIP* predicate, which must hold of a group and not an individual. These conflicting requirements make a coherent account of this reading difficult or impossible to obtain under the assumptions of the property theory.

In addition to these arguments, Higginbotham (1992) presents a number of other arguments supporting the proposition theory of control; we conclude with Higginbotham that the proposition theory has clear advantages over the property theory of control, and we adopt the proposition theory here.

4.2. Equi and Obligatory Anaphoric Control

Since we treat the subject of the COMP argument of an equi verb like *tried* as a pronoun whose antecedent is the controller, we reintroduce the context information and the context-meaning pair constructors used in our analysis of anaphoric binding and the syntax-semantics interface, as discussed in Chapter 11. The meaning contribution of a sentence like *David tried to leave* is a pair: the first member is the meaning of the sentence *try*(*David*, *leave*(*David*)), and the second member is the list of discourse referents introduced into the context when the sentence is evaluated. Correspondingly, the right-hand side of the meaning constructor for this example is a pair consisting of the semantic structure for the sentence and a list of semantic resources corresponding to the discourse referents that have been introduced:

(61) *David tried to leave.*

$$f\left[\begin{array}{ll}\text{PRED} & \text{`TRY}\langle\text{SUBJ,COMP}\rangle\text{'}\\ \text{SUBJ} & g\left[\begin{array}{ll}\text{PRED} & \text{`DAVID'}\end{array}\right]\\ \text{COMP} & h\left[\begin{array}{ll}\text{PRED} & \text{`LEAVE}\langle\text{SUBJ}\rangle\text{'}\\ \text{SUBJ} & i\left[\begin{array}{ll}\text{PRED} & \text{`PRO'}\end{array}\right]\end{array}\right]\end{array}\right]$$

$$[\textit{try}(\textit{David}, \textit{leave}(\textit{David})), \langle \textit{David}, \textit{David}\rangle] : f_\sigma \otimes \langle i_\sigma, g_\sigma \rangle$$

We assume that both the matrix subject *David* and the unexpressed pronominal subject of the complement verb *leave* contribute discourse referents to the context, so that both g_σ and i_σ are elements of the context list. The contribution of a discourse referent for the subject of the complement clause allows the resolution of bound anaphors in examples like (62), where the antecedent of the reflexive pronoun *himself* is the unexpressed pronominal subject of *vote*:

(62) *David tried to vote for himself.*

We propose the lexical entry given in (63) for the equi verb *tried*:

(63) *tried* $(\uparrow \text{ PRED}) = \text{`TRY}\langle\text{SUBJ,COMP}\rangle\text{'}$

$(\uparrow \text{ COMP SUBJ PRED}) = \text{`PRO'}$

$((\uparrow \text{ COMP SUBJ})_\sigma \text{ ANTECEDENT}) = (\uparrow \text{ SUBJ})_\sigma$

$\lambda X.\lambda P.\mathit{try}(X,P) : (\uparrow \text{ SUBJ})_\sigma \multimap [(\uparrow \text{ COMP})_\sigma \multimap \uparrow_\sigma]$

$$\begin{aligned}&\lambda C.[\mathit{first}(C), \langle \mathit{first}(C), C\rangle] \; : \\ &\quad \forall C.\langle(\uparrow \text{ SUBJ})_\sigma, C\rangle \multimap \\ &\qquad [(\uparrow \text{ COMP SUBJ})_\sigma \otimes \langle(\uparrow \text{ COMP SUBJ})_\sigma, (\uparrow \text{ SUBJ})_\sigma, C\rangle]\end{aligned}$$

The first two lines of this lexical entry are familiar from our syntactic discussion in Section 3.3 of this chapter. The third line establishes the antecedency of the pronominal subject of the COMP argument: semantically, the antecedent of the COMP SUBJ is the matrix clause controller, the SUBJ of the equi verb *tried*. In (64), we instantiate this constraint according to the f-structure labels in (61):

(64) $(i_\sigma \text{ ANTECEDENT}) = g_\sigma$

The meaning constructor in the fourth line of the lexical entry, labeled **[try]** in (66) below, provides the main predicate *try*. On the right-hand side of this meaning constructor, the meanings of the SUBJ and the COMP arguments are consumed, and a meaning for the sentence is produced. On the left-hand side, the predicate *try* is applied to the meaning X of the SUBJ and the meaning P of the COMP, producing the meaning $\mathit{try}(X, P)$ for the sentence.

The final part of the lexical entry is the pronominal meaning constructor supplied by the equi verb for the interpretation of the pronominal SUBJ of its complement clause, labeled **[pro]** in (66) below. Instantiating this meaning constructor according to the labels given in (61), we have the instantiated meaning constructor in (65):

(65) $\lambda C.[\mathit{first}(C), \langle \mathit{first}(C), C\rangle] : \forall C.\langle g_\sigma, C\rangle \multimap [i_\sigma \otimes \langle i_\sigma, g_\sigma, C\rangle]$

As discussed in Chapter 11, Section 3.4.2, the meaning constructor for a pronoun requires as its argument a context resource that contains its antecedent: in other words, the antecedent of the pronoun must be available in the context when the pronoun is resolved. Thus, on the right-hand side of this meaning constructor, we require a context argument of the form $\langle g_\sigma, C\rangle$, a context that includes the antecedent semantic resource g_σ. In resolving the pronoun, this context is consumed and a new context $\langle i_\sigma, g_\sigma, C\rangle$ is produced; the new context is identical to the old context except that the pronoun semantic resource i_σ is added as the first element.

The left-hand side of the meaning constructor in (65) is applied to a context list C, producing a meaning-context pair. The second member of the pair is an augmented context list in which the discourse referent corresponding to the antecedent, $\mathsf{first}(C)$, has been added to the context list as the discourse referent of the pronoun. The antecedent meaning also appears as the first member of the pair, associated with the semantic resource for the pronoun.

The instantiated meaning constructors given in (66) are relevant in the analysis of example (61). We assume that this sentence is uttered at the beginning of a discourse, so we also provide the empty context **[context]** as input:

(66) Meaning constructor premises for *David tried to leave*:

$$\begin{array}{lrcl}
\textbf{[context]} & \langle\rangle & : & \langle\rangle \\
\textbf{[try]} & \lambda X.\lambda P.\mathsf{try}(X,P) & : & g_\sigma \multimap [h_\sigma \multimap f_\sigma] \\
\textbf{[David]} & \lambda C.[\mathsf{David}, \langle \mathsf{David}, C\rangle] & : & \forall C.C \multimap [g_\sigma \otimes \langle g_\sigma, C\rangle] \\
\textbf{[leave]} & \lambda X.\mathsf{leave}(X) & : & i_\sigma \multimap h_\sigma \\
\textbf{[pro]} & \lambda C.[\mathsf{first}(C), \langle \mathsf{first}(C), C\rangle] & : & \forall C.\langle g_\sigma, C\rangle \multimap [i_\sigma \otimes \langle i_\sigma, g_\sigma, C\rangle]
\end{array}$$

We first combine the input context **[context]** with **[David]**, producing the meaning constructor labeled **[context-David]** in (67):

(67) **[context-David]** $[\mathsf{David}, \langle \mathsf{David}\rangle] : g_\sigma \otimes \langle g_\sigma\rangle$

We now split this resource into a context resource and a standard noncontextual meaning constructor by using the multiplicative conjunction deduction rule **Context split**, given in (61) of Chapter 11. This produces two meaning constructors, **[newcontext]** and **[David2]**:

(68)

$$\begin{array}{lrcl}
\textbf{[newcontext]} & \langle \mathsf{David}\rangle & : & \langle g_\sigma\rangle \\
\textbf{[David2]} & \mathsf{David} & : & g_\sigma
\end{array}$$

The new context **[newcontext]** is appropriate for resolving the pronoun **[pro]**, since it provides a resource for the antecedent g_σ of the pronoun. Combining **[pro]** and **[newcontext]**, we have the meaning constructor labeled **[resolved-pro]** in (69):

(69) **[resolved-pro]** $[\mathsf{David}, \langle \mathsf{David}, \mathsf{David}\rangle] : i_\sigma \otimes \langle i_\sigma, g_\sigma\rangle$

Again, we use the **Context split** deduction rule to split this meaning contribution into a context contribution **[newcontext2]** and a contribution **[resolvedpro2]** for the resolved pronoun:

(70)

$$\begin{array}{lrcl}
\textbf{[newcontext2]} & \langle \mathsf{David}, \mathsf{David}\rangle & : & \langle i_\sigma, g_\sigma\rangle \\
\textbf{[resolvedpro2]} & \mathsf{David} & : & i_\sigma
\end{array}$$

We can combine **[resolvedpro2]** with **[leave]**, producing the meaning constructor **[pro-leave]** in (71):

(71) **[pro-leave]** $\mathit{leave}(\mathit{David}) : h_\sigma$

We can also combine **[David2]** with the meaning constructor **[try]** to produce the meaning constructor **[David-try]** in (72):

(72) **[David-try]** $\lambda P.\mathit{try}(\mathit{David}, P) : h_\sigma \multimap f_\sigma$

The meaning constructor in (72) requires a meaning resource for h_σ, which is provided by **[pro-leave]**. Combining **[David-try]** and **[pro-leave]**, we have the meaning constructor labeled **[David-try-pro-leave]** in (73):

(73) **[David-try-pro-leave]** $\mathit{try}(\mathit{David}, \mathit{leave}(\mathit{David})) : f_\sigma$

We have now produced a complete meaning for example (61), but it is not yet a coherent meaning, since the meaning constructor **[newcontext2]** has not yet been integrated. We use the **Context merge** deduction rule given in (62) of Chapter 11 to combine **[newcontext]** with **[David-try-pro-leave]**, producing the semantically complete and coherent meaning constructor in (74), as desired:

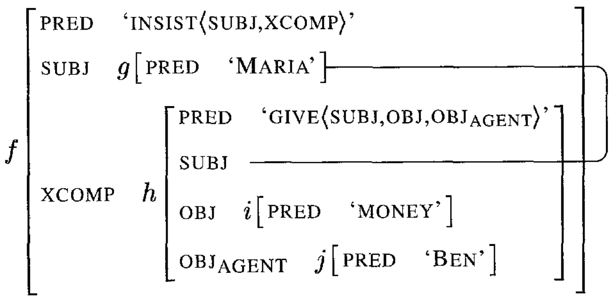
(74) **[newcontext2]**, **[David-try-pro-leave]** $\vdash$

$$[\mathit{try}(\mathit{David}, \mathit{leave}(\mathit{David})), \langle \mathit{David}, \mathit{David} \rangle] : f_\sigma \otimes \langle i_\sigma, g_\sigma \rangle$$

4.3. Equi and Functional Control

In Section 3.4 of this chapter, we saw that some equi verbs in Tagalog involve functional rather than anaphoric control. Example (47), repeated here, exemplifies functional control:

(75) *nagpilit* *si-Maria-ng* *bigy-an* *ng-pera* *ni-Ben*
PERFECT.ACTIVE.insist.on NOM-Maria-COMP give-DAT GEN-money GEN-Ben
'Maria insisted on being given money by Ben.'



$\mathit{insist}(\mathit{Maria}, \mathit{give}(\mathit{Ben}, \mathit{money}, \mathit{Maria})) : f_\sigma$

Since this example involves functional rather than anaphoric control, context management and anaphora resolution are not a central issue in meaning derivation, and we use simple meaning constructors rather than context-meaning pair constructors.

We assume that the XCOMP complement of the equi verb *nagpilit* 'insist on' patterns with other equi verbs in denoting a proposition, though we have not been able to verify the semantic patterns presented in Section 4.1 of this chapter with a native speaker of Tagalog. On this assumption, we propose the lexical entry in (76) for *nagpilit* 'insist on':

(76) *nagpilit* $(\uparrow \text{PRED}) = \text{'INSIST}\langle\text{SUBJ,XCOMP}\rangle\text{'}$
$(\uparrow \text{SUBJ}) = (\uparrow \text{XCOMP SUBJ})$

$$\lambda P.\lambda X.\mathit{insist}(X, P(X)) \ : \ [(\uparrow \text{SUBJ})_\sigma \multimap (\uparrow \text{XCOMP})_\sigma] \multimap [(\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma]$$

The first two lines of this lexical entry enforce the syntactic constraints appropriate for a subject raising verb. The rest of the entry consists of the meaning constructor for this verb, which requires two arguments: an implicational resource $[(\uparrow \text{SUBJ})_\sigma \multimap (\uparrow \text{XCOMP})_\sigma]$ corresponding to its XCOMP argument and a resource $(\uparrow \text{SUBJ})_\sigma$ corresponding to its subject. The XCOMP resource takes the form of an implication because it is an argument that is "missing" its subject (which is also the subject of the main verb). The subject's meaning is represented by X on the left-hand side of the meaning constructor in (76); the XCOMP's meaning is the property P, which is applied to the subject meaning X to produce the proposition $P(X)$ filling the second argument position of *insist*.

The meaning constructor in the lexical entry in (76) is shown in (77), instantiated with the f-structure labels given in (75); we also provide the standard meaning constructors for the proper names *Maria* and *Ben* and for the ditransitive verb *bigy-an* 'give'. Since the internal structure of the noun phrase *ng-pera* 'money' is not at issue here, we make the simplifying assumption that this noun phrase makes a contribution like that of a proper name and has the meaning *money*:

(77) Meaning constructor premises for *nagpilit si-Maria-ng bigy-an ng-pera ni-Ben*:

[insist]	$\lambda P.\lambda X.\mathit{insist}(X, P(X))$	$: [g_\sigma \multimap h_\sigma] \multimap [g_\sigma \multimap f_\sigma]$
[Maria]	Maria	$: g_\sigma$
[give]	$\lambda X.\lambda Y.\lambda Z.\mathit{give}(X, Y, Z)$	$: j_\sigma \multimap [i_\sigma \multimap [g_\sigma \multimap h_\sigma]]$
[money]	money	$: i_\sigma$
[Ben]	Ben	$: j_\sigma$

We first combine the premises labeled **[Ben]**, **[money]**, and **[give]** to produce the meaning constructor labeled **[give-Ben-money]** in (78):

(78) **[give-Ben-money]** $\lambda Z.\mathit{give}(\mathit{Ben}, \mathit{money}, Z) : g_\sigma \multimap h_\sigma$

This meaning constructor provides the implicational resource $g_\sigma \multimap h_\sigma$ required by the meaning constructor labeled **[insist]**. On the left-hand side, we obtain the corresponding meaning by applying the function $\lambda P.\lambda X.\mathit{insist}(X, P(X))$ to its argument $\lambda Z.\mathit{give}(\mathit{Ben}, \mathit{money}, Z)$. Combining **[give-Ben-money]** and **[insist]**, we have the meaning constructor labeled **[insist-give-Ben-money]** in (79):

(79) **[insist-give-Ben-money]** $\lambda X.\mathit{insist}(X, \mathit{give}(\mathit{Ben}, \mathit{money}, X)) : g_\sigma \multimap f_\sigma$

Finally, we combine this meaning constructor with the subject meaning constructor, labeled **[Maria]**. The resulting meaning constructor, displayed in (80), provides a semantically complete and coherent meaning for this example:

(80) **[insist-give-Ben-money], [Maria]** $\vdash$

$$\mathit{insist}(\mathit{Maria}, \mathit{give}(\mathit{Ben}, \mathit{money}, \mathit{Maria})) : f_\sigma$$

5. ARBITRARY ANAPHORIC CONTROL

5.1. Syntax of Arbitrary Control

As we have seen, English equi verbs such as *try* or *persuade* participate in the obligatory anaphoric control construction, where the referent of the controlled subordinate clause SUBJ is determined by constraints associated with the matrix verb. This situation contrasts with constructions involving *arbitrary control*, in which the reference of the pronominal element in the subordinate clause is not syntactically determined. In an arbitrary control construction, the reference of the controlled argument in the subordinate clause is resolved like any other pronoun, as described by Bresnan (1982a, 2001b). Example (81) involves arbitrary anaphoric control and means that David gestured for some unspecified individual or individuals to follow Chris:

(81) *David gestured to follow Chris.*

$$\begin{bmatrix} \text{PRED} & \text{`GESTURE}\langle\text{SUBJ,COMP}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`DAVID'} \end{bmatrix} \\ \text{COMP} & \begin{bmatrix} \text{PRED} & \text{`FOLLOW}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{`CHRIS'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

There are a number of differences between arbitrary control and functional or obligatory anaphoric control. Bresnan (1982a, 2001b) shows that for arbitrary anaphoric control, though not for functional or obligatory anaphoric control, a split antecedent (an antecedent that does not form a syntactic unit) is possible. In (82), the subject of *follow* can be interpreted as the group consisting of Chris and Matty, even though there is no single constituent representing this group in the matrix clause:

(82) *Chris told Matty that David had gestured to follow Ken.*
[possible interpretation: Chris and Matty follow Ken]

A syntactically remote antecedent is also possible. In (83), the subject of *follow* can be interpreted as coreferent with *Chris*, although the noun phrase *Chris* is not an argument of the immediately higher clause:

(83) *Chris thought that it was clear that David had gestured to follow Ken.*
[possible interpretation: Chris follows Ken]

And there may be no expressed antecedent in the same sentence at all, as in example (81).

These semantic differences will come out more clearly in the next section, when we discuss the interpretation of anaphoric control constructions. Syntactically, obligatory and arbitrary control constructions do not differ; the same phrase structure rule, given in (38) of this chapter, is used in deriving the two, and the syntactic portions of the lexical entries are similar. We propose the following lexical entry for a verb like *gesture*:

(84) *gesture* V ($\uparrow$ PRED) = 'GESTURE⟨SUBJ,COMP⟩'
($\uparrow$ COMP SUBJ PRED) = 'PRO'

Example (87) (page 340) shows the c-structure and f-structure for the example *David gestured to follow Chris.*

5.2. Semantics of Arbitrary Control

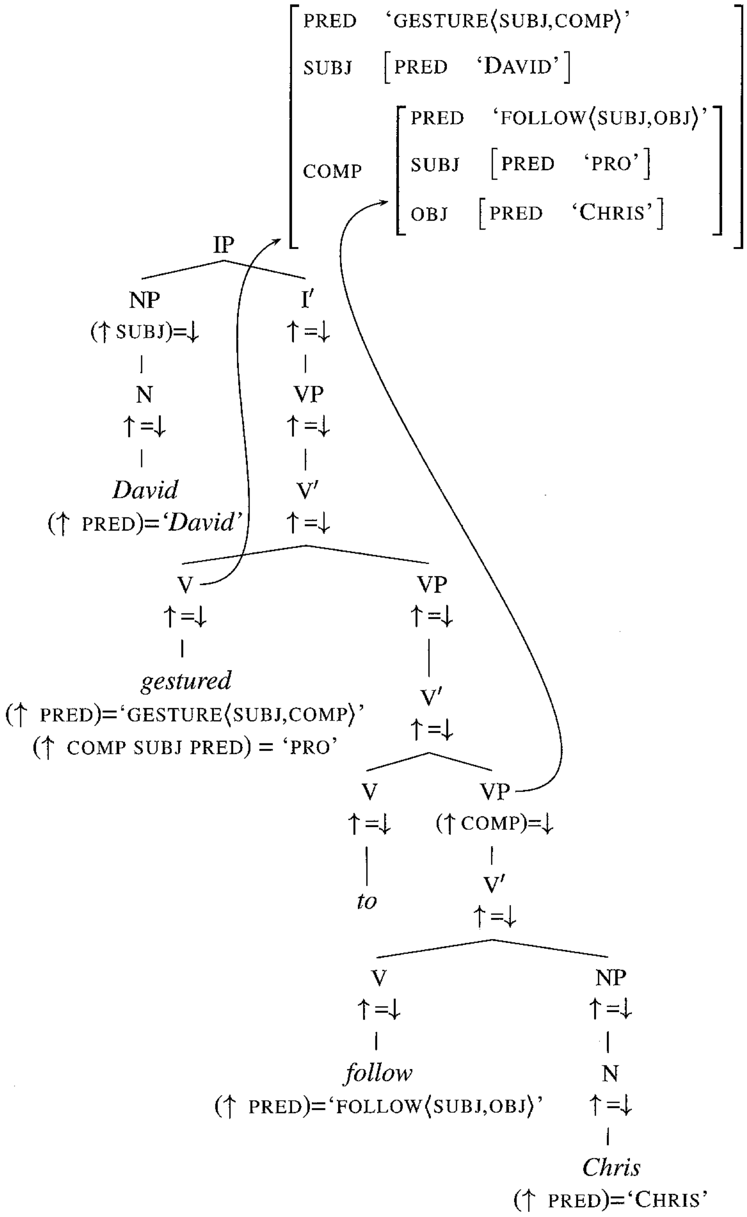
Consider the short discourse in (85):

(85) *Chris yawned. David gestured to leave.*

The most likely interpretation of the sentence *David gestured to leave* in (85) is that David gestured for Chris to leave:

(86) *Chris yawned. David gestured to leave.*
yawn(*Chris*), *gesture*(*David*, *leave*(*Chris*))

(87) *David gestured to follow Chris.*



In other contexts, other antecedents are possible. Unlike the situation with obligatory anaphoric control, lexical or syntactic constraints do not determine the referent of the complement subject. Instead, the pronominal subject of the complement *to leave* obeys the same constraints on pronoun resolution as overt pronouns.

However, lexical constraints can play an important role in constraining the range of possible referents for the controller in an arbitrary control construction. In (87), the matrix subject cannot corefer with the subject of the COMP: that is, a sentence like *David gestured to leave* cannot mean that David gestured for himself to leave. This is a *negative constraint* on anaphoric reference, which in this case is imposed by the verb *gesture*; negative constraints are discussed in Sections 2.1.2 and 3.4 of Chapter 11.

5.3. Arbitrary Control and Meaning Assembly

Since the unexpressed subject of the COMP of the verb *gestured* is interpreted as a pronominal whose antecedent must be resolved, we require a representation of the context in the derivation of the meanings of the sentences in the mini-dialogue in (85). The f-structure and meaning constructor for the first sentence in the dialogue are given in (88):

(88) *Chris yawned.*

$$f\begin{bmatrix} \textsc{pred} & \text{`}\textsc{yawn}\langle\textsc{subj}\rangle\text{'} \\ \textsc{subj} & g\begin{bmatrix}\textsc{pred} & \text{`}\textsc{Chris}\text{'}\end{bmatrix} \end{bmatrix}$$

$$[\mathit{yawn}(\mathit{Chris}), \langle \mathit{Chris}\rangle] : f_\sigma \otimes \langle g_\sigma \rangle$$

The left-hand side of the meaning constructor in (88) is a pair whose first member is the sentential meaning $\mathit{yawn}(\mathit{Chris})$ and whose second pair is the list of discourse referents $\langle \mathit{Chris}\rangle$ that are in context after this sentence is uttered. The right-hand side is also a pair: the first member is the semantic resource f_σ corresponding to the sentence meaning, and the second member is the list of semantic resources $\langle g_\sigma\rangle$ corresponding to the discourse referents in context. We assume the premises in (89) for the analysis of this sentence:

(89) Meaning constructor premises for *Chris yawned*:

[context]	$\langle\rangle : \langle\rangle$
[yawn]	$\lambda X.\mathit{yawn}(X) : g_\sigma \multimap f_\sigma$
[Chris]	$\lambda C.[\mathit{Chris}, \langle \mathit{Chris}, C\rangle] : \forall C.C \multimap [g_\sigma \otimes \langle g_\sigma, C\rangle]$

These premises closely resemble the premises given for the proper name *David*, the intransitive verb *arrive*, and the empty context in the analysis of the sentence *David arrived*, presented in Chapter 11, Section 3.3.2. The derivation of the meaning of the sentence proceeds analogously. Combining these premises produces the semantically complete and coherent meaning constructor in (90), as desired:

(90) **[context]**, **[yawn]**, **[Chris]** $\vdash$ $[\mathit{yawn}(\mathit{Chris}), \langle \mathit{Chris} \rangle] : f_\sigma \otimes \langle g_\sigma \rangle$

The second sentence in the mini-dialogue in (85), *David gestured to leave*, has the f-structure and meaning constructor given in (91):

(91) *David gestured to leave.*

$$h\left[\begin{array}{ll} \textsc{pred} & \text{`GESTURE}\langle\textsc{subj},\textsc{comp}\rangle\text{'} \\ \textsc{subj} & i\left[\begin{array}{ll}\textsc{pred} & \text{`DAVID'}\end{array}\right] \\ \textsc{comp} & j\left[\begin{array}{ll} \textsc{pred} & \text{`LEAVE}\langle\textsc{subj}\rangle\text{'} \\ \textsc{subj} & k\left[\begin{array}{ll}\textsc{pred} & \text{`PRO'}\end{array}\right] \end{array}\right] \end{array}\right]$$

$$[\mathit{gesture}(\mathit{David}, \mathit{leave}(\mathit{Chris})), \langle \mathit{Chris}, \mathit{Chris}, \mathit{David} \rangle] : h_\sigma \otimes \langle k_\sigma, g_\sigma, i_\sigma \rangle$$

The first member of the pair meaning constructor in (91) is the meaning of the sentence $\mathit{gesture}(\mathit{David}, \mathit{leave}(\mathit{Chris}))$, corresponding to the semantic resource h_σ. Its second member is the list of discourse referents that are available after this sentence is uttered: the discourse referents *David* and *Chris* and their corresponding semantic resources i_σ and k_σ are contributed by this sentence, and the discourse referent *Chris* and its corresponding semantic resource g_σ was contributed by the previous sentence *Chris yawned.*

The lexical entry for the arbitrary control verb *gestured* is given in (92):

(92) *gestured* $(\uparrow \textsc{pred}) = \text{`GESTURE}\langle\textsc{subj},\textsc{comp}\rangle\text{'}$

$$\lambda X.\lambda P.\mathit{gesture}(X, P) : (\uparrow \textsc{subj})_\sigma \multimap [(\uparrow \textsc{comp})_\sigma \multimap \uparrow_\sigma]$$

$$(\uparrow \textsc{comp subj pred}) = \text{`PRO'}$$
$$((\uparrow \textsc{comp subj})_\sigma\ \textsc{antecedent}) \neq (\uparrow \textsc{subj})_\sigma$$

$$\lambda C.[\mathit{first}(C), \langle \mathit{first}(C), C \rangle] :$$
$$\forall C.\langle((\uparrow \textsc{comp subj})_\sigma\ \textsc{antecedent}), C\rangle \multimap$$
$$[(\uparrow \textsc{comp subj})_\sigma \otimes \langle(\uparrow \textsc{comp subj})_\sigma, ((\uparrow \textsc{comp subj})_\sigma\ \textsc{antecedent}), C\rangle]$$

The first line of this lexical entry provides the PRED value for the verb *gestured.* The second line is the meaning constructor that introduces the predicate *gesture*, labeled **[gesture]** in (94) below: its first argument X represents the meaning of the SUBJ of *gesture*, and its second argument P represents the meaning of its COMP.

The third line of the lexical entry specifies a 'PRO' value for the PRED attribute of the complement clause. The fourth line introduces a constraint on its antecedent:

(93) $((\uparrow \textsc{comp subj})_\sigma\ \textsc{antecedent}) \neq (\uparrow \textsc{subj})_\sigma$

As noted earlier, the SUBJ of *gesture* cannot corefer with its COMP SUBJ: the sentence *David gestured to leave* cannot mean that David gestured for himself to leave. The constraint in (93) enforces this requirement by preventing the subject's semantic structure $(\uparrow \text{ SUBJ})_\sigma$ from being chosen as the ANTECEDENT value of $(\uparrow \text{ COMP SUBJ})_\sigma$. In the current context, we choose g_σ as the antecedent; this is acceptable because g_σ is accessible in the context and the constraint in (93) is not violated.

The final lines of this lexical entry provide a pronominal meaning constructor for the COMP SUBJ of *gesture*, labeled **[pro]** in (94). As with other pronominals, the right-hand side of this meaning constructor consumes a context resource that contains the semantic resource of its antecedent. Besides a resource for the pronoun itself, it produces a new, updated context resource to which the pronoun's semantic resource has been added. On the left-hand side, the meaning of the antecedent is assigned to the pronoun and added to the context list.

With the context **[context]** produced by uttering the first sentence in the discourse, *Chris yawned*, the premises in (94) are relevant for the deduction of the meaning of *David gestured to leave*:

(94) Meaning constructor premises for *David gestured to leave*:

[context]	$\langle g_\sigma \rangle$	:	$\langle \mathit{Chris} \rangle$
[gesture]	$\lambda X.\lambda Y.\mathit{gesture}(X,Y)$	:	$i_\sigma \multimap [j_\sigma \multimap h_\sigma]$
[David]	$\lambda C.[\mathit{David}, \langle \mathit{David}, C \rangle]$	:	$\forall C.C \multimap [i_\sigma \otimes \langle i_\sigma, C \rangle]$
[pro]	$\lambda C.[\mathit{first}(C), \langle \mathit{first}(C), C \rangle]$	:	$\forall C.\langle g_\sigma, C \rangle \multimap [k_\sigma \otimes \langle k_\sigma, g_\sigma, C \rangle]$
[leave]	$\lambda X.\mathit{leave}(X)$	:	$k_\sigma \multimap j_\sigma$

We first combine **[context]** with **[David]**, producing the meaning constructor labeled **[context-David]** in (95):

(95) **[context-David]** $[\mathit{David}, \langle \mathit{David}, \mathit{Chris} \rangle] : i_\sigma \otimes \langle i_\sigma, g_\sigma \rangle$

In (95), the discourse referent *David* is the first member of the context list. As noted in Chapter 11, Section 3.3.1, the meaning constructor in (96), where the context list has been reordered, is exactly equivalent to the one in (95):

(96) **[context-David]** $[\mathit{David}, \langle \mathit{Chris}, \mathit{David} \rangle] : i_\sigma \otimes \langle g_\sigma, i_\sigma \rangle$

We use the rule **Context split**, given in (61) of Chapter 11, to produce the two meaning constructors **[newcontext]** and **[David2]** from **[context-David]**:

(97)

[newcontext]	$\langle \mathit{Chris}, \mathit{David} \rangle$	:	$\langle g_\sigma, i_\sigma \rangle$
[David2]	David	:	i_σ

We combine **[David2]** with **[gesture]** to produce the meaning constructor labeled **[David-gesture]** in (98):

(98) **[David-gesture]** $\lambda Y.\mathit{gesture}(\mathit{David}, Y) : j_\sigma \multimap h_\sigma$

We can also combine the premises **[newcontext]** and **[pro]**, producing **[context-pro]**, given in (99):

(99) **[context-pro]** $[\mathit{Chris}, \langle \mathit{Chris}, \mathit{Chris}, \mathit{David} \rangle] : k_\sigma \otimes \langle k_\sigma, g_\sigma, i_\sigma \rangle$

We use the rule **Context split** to produce the two meaning constructors **[newcontext2]** and **[pro2]**:

(100) **[newcontext2]** $\langle \mathit{Chris}, \mathit{Chris}, \mathit{David} \rangle : \langle k_\sigma, g_\sigma, i_\sigma \rangle$

[pro2] $\mathit{Chris} : k_\sigma$

We combine the meaning constructor labeled **[pro2]** with **[leave]** to produce **[pro-leave]**, given in (101):

(101) **[pro-leave]** $\mathit{leave}(\mathit{Chris}) : j_\sigma$

We can now combine **[David-gesture]** with **[pro-leave]**, producing the meaning constructor labeled **[David-gesture-pro-leave]** in (102):

(102) **[David-gesture-pro-leave]** $\mathit{gesture}(\mathit{David}, \mathit{leave}(\mathit{Chris})) : h_\sigma$

Finally, we use the rule **Context merge**, given in (62) of Chapter 11, to combine **[David-gesture-pro-leave]** with **[newcontext2]**, producing the semantically complete and coherent meaning constructor in (103), as desired:

(103) **[newcontext2], [David-gesture-pro-leave]** $\vdash$

$$[\mathit{gesture}(\mathit{David}, \mathit{leave}(\mathit{Chris})), \langle \mathit{Chris}, \mathit{Chris}, \mathit{David} \rangle] : h_\sigma \otimes \langle k_\sigma, g_\sigma, i_\sigma \rangle$$

6. THE CONTROLLER IN ANAPHORIC OR FUNCTIONAL CONTROL

Determination of the controller in functional and obligatory control is constrained by both syntactic and semantic factors. In both kinds of control constructions, the controller must be a *term* (Chapter 2, Section 1.3). Further, as discussed in detail by Sag and Pollard (1991), the choice of controller in equi verbs is semantically constrained; equi verbs can be divided into semantic classes, and the semantic role of the controller can be predicted from these classes.

6.1. Syntactic Requirements

Bresnan (1982a) presents the following constraints on determination of the controller in control constructions:

(104) a. The controller must be a term (SUBJ, OBJ, or OBJ$_\theta$).

b. By default, the controller is the lowest available argument on the grammatical function hierarchy SUBJ > OBJ > OBJ$_\theta$.

The first requirement is that the controller is required to be a term: either SUBJ, OBJ, or OBJ$_\theta$. As demonstrated by Bresnan (1982a), this makes several strikingly correct predictions.

First, it accounts for what is known as *Visser's Generalization* (Visser 1963–1973; Bresnan 1982a), according to which there is no passive version of a verb involving subject control:[5]

(105) a. *John promised Mary to be on time.*

b. **Mary was promised by John to be on time.*

(106) a. *He strikes his friends as pompous.*

b. **His friends are struck by him as pompous.*

Visser's Generalization follows from the constraint in (104a) since an oblique or adjunct phrase such as *by John* is not a term and therefore cannot participate as a controller in a control construction.

Second, the constraint in (104a) accounts for what is known as *Bach's Generalization* (Bach 1980): there is no detransitivized version of a verb involving object control. Bresnan (1982a) presents the following illustrative examples, which involve both functional and anaphoric control:

(107) Anaphoric control by subject:

a. *Louise promised Tom to be on time.*

b. *Louise promised to be on time.*

[5] Annie Zaenen points out that example (a), in which a passive form of the German verb*versuchen* 'try' appears, constitutes an apparent counterexample to this generalization:

(a) *es wird versucht das Auto zu reparieren*
it is tried the car to repair
'It is tried to repair the car.'

Alternative analyses of this sentence have been proposed, however. Sentences with*versuchen* and an infinitival complement have been argued to be monoclausal ('*coherent*') constructions (Bech 1957; Berman 2000), which do not involve either anaphoric or functional control. Klaus Netter (p.c.) argues that in this example, the phrase *das Auto zu reparieren* 'to repair the car' is actually an extraposed subject, and neither functional nor anaphoric control is involved.

(108) Anaphoric control by object:

a. *Louise taught Tom to smoke.*

b. **Louise taught to smoke.*

(109) Functional control by object:

a. *Louise believed Tom to smoke.*

b. **Louise believed to smoke.*

Bach's Generalization follows from the constraint in (104a), since the controller must be present as a term argument in a control construction. Of course, Bach's Generalization does not apply to arbitrary anaphoric control constructions, since there are no syntactic constraints on the determination of the controller of the subject of the subordinate COMP in an arbitrary anaphoric control construction:

(110) a. *Louise gestured to Tom to follow.*

b. *Louise gestured to follow.*

The requirement in (104b) involves a violable syntactic hierarchy of default controllers in control constructions: Bresnan (1982a) claims that the unmarked choice for a controller is OBJ$_\theta$ if there is one, otherwise OBJ if there is one, and otherwise SUBJ. The following control constructions obey this rule:

(111) a. *David tried to leave.*

b. *David persuaded Chris to leave.*

Verbs like *promise* constitute exceptions to this default, since the SUBJ and not the OBJ is the controller:

(112) *Chris promised David to leave.*

Both of the generalizations in (104) are best thought of as constraining the determination of grammatical functions for verbs involving functional or anaphoric control; for discussion of the theory of mapping between semantic roles and grammatical functions, see Chapter 8.

6.2. Semantic Requirements

In addition to syntactic constraints on how the controller is realized, there are also semantic generalizations about controller choice in constructions involving equi verbs (which, as shown above, can involve either functional or anaphoric control). Sag and Pollard (1991) and Pollard and Sag (1994) provide a detailed exploration of different classes of equi verbs and propose semantic principles for determination of the controller in equi constructions. For example, they propose that verbs such as *order*, *persuade*, and *bid* are members of what they call the *order/permit* class of equi verbs. Verbs of this class refer to situations where a participant is influenced by another participant to perform a certain action, and the controller is always the OBJ argument of the active equi verb. This class contrasts with the *promise* class, containing verbs like *promise*, *agree*, and *demand*, in which the controller is always the SUBJ of the active equi verb. Thus, generalizations about linking in equi verbs — how the semantic arguments of these verbs are linked to the syntactic functions SUBJ, OBJ, OBJ_θ, and COMP, as described in Chapter 8 — are formulated with reference to these larger classes, not separately specified for each verb. Sag and Pollard's useful classifications are used in analyses of control in Balinese by Arka (1998) and Arka and Simpson (1998).

7. THE CONTROLLED FUNCTION

It is often assumed that only subjects can be controlled, and indeed this is true in English, Icelandic, Warlpiri, and many other languages. However, the grammatical function of the controllee is not always a SUBJ; there are also languages that allow nonsubjects to be controlled. As shown in example (48) of this chapter, nonsubjects can participate in anaphoric control in Tagalog. Other evidence for variation in controller choice can also be found. In a survey of raising constructions, Bader (1995) notes that a number of other languages allow nonsubject controllees, including Samoan, Blackfoot, and Kunuz Nubian.

Seiter (1983) presents a detailed and carefully argued discussion of raising constructions in Niuean, demonstrating convincingly that either the SUBJ or the OBJ of the subordinate clause can participate in functional control. Niuean is a VSO language with ergative case marking. In example (113), a construction that does not involve raising, ERGative casemarking appears on the complement clause SUBJ, and ABsolutive casemarking appears on the OBJ:

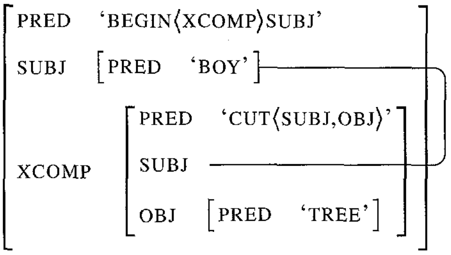
(113)	*kua*	*kamata*	*ke*	*hala*	*[he*	*tama]*	*[e*	*akau]*
	PERFECT	begin	SUBJUNCTIVE	cut	ERG	boy	ABS	tree

'The boy has begun to cut down the tree.'

In example (114), the SUBJ of the complement clause appears as the SUBJ of the matrix clause, similar to the raising construction in English:[6]

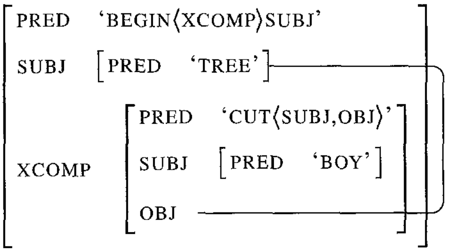
[6] The matrix clause subject *e tama* 'boy' in example (114) is marked with absolutive case, consistent with its role as the SUBJ of the matrix verb *kamata* 'begin'. This argument is also the SUBJ of the subordinate clause verb *hala* 'cut'. The casemarking pattern in (114) entails that the requirement for ergative casemarking on the subject of *hala* 'cut' in examples like (112) is not part of the lexical specifications of the verb, unlike the situation with the "quirky case" Icelandic verbs discussed in Section 1.1 of this chapter.

(114) *kua* *kamata [e* *tama] ke* *hala [e* *akau]*
PERFECT begin ABS boy SUBJUNCTIVE cut ABS tree
'The boy has begun to cut down the tree.'



Example (115) is similar to (114) except that the OBJ of the complement clause appears as the SUBJ of the matrix clause:

(115) *kua* *kamata [e* *akau] ke* *hala [he* *tama]*
PERFECT begin ABS tree SUBJUNCTIVE cut ERG boy
'The tree has begun to be cut down by the boy.'
('The tree has begun the boy to cut down ___.')



In examples (113–115), the subordinate clause verb *hala* 'cut' is transitive, subcategorizing for a SUBJ and an OBJ. In example (115), the phrase *he tama* 'boy' is marked with ERG case; it is the subordinate clause SUBJ. The "raised" argument *e akau* 'tree' is the SUBJ of the matrix clause and also the subordinate clause OBJ.

The casemarking of arguments in examples (113–115) constitutes one piece of evidence for this conclusion: the overt noun phrase in the subordinate clause in example (115) is ergative, the case used to mark subjects of transitive verbs. Thus, the argument that appears in the subordinate clause is the SUBJ, and the "raised" noun phrase must be the OBJ. Seiter (1983) provides further evidence that the OBJ as well as the SUBJ can participate as a controllee in raising, discussing evidence from subject-verb agreement, object-verb agreement, and quantifier float.

8. CONTROL IN ADJUNCTS

Thus far, we have examined functional and anaphoric control in the closed grammatical function COMP and the open function XCOMP. Modifiers may also participate in either functional or anaphoric control. The open function XADJ contains an open position functionally controlled by an argument of the matrix clause, and some modifying adjuncts ADJ participate in anaphoric control. Mohanan (1983) discusses control in modifier phrases, showing that control relations must be defined in functional and not phrasal terms: English and Malayalam are quite different in constituent structure, but obey similar functional constraints in control constructions.

8.1. Functional Control and XADJ

The open adjunct function XADJ is similar to the open function XCOMP in participating in functional control. The SUBJ of the XADJ is identified with an argument of the matrix clause, and the same f-structure fills both functions. Andrews (1990a) discusses the following Icelandic examples involving control of an XADJ:[7]

(116) *Njósnaranum var kastað einum út úr þyrlunni.*
spy.DEF.DAT was thrown alone.DAT out from helicopter.DEF
'The spy was thrown out of the helicopter alone.'

(117) *Ég mœtti Sveini drukknum.*
I met Svein.DAT drunk.DAT
'I met Svein drunk.'

In examples (116–117), the dative form of the open XADJ adjuncts *einum* 'alone' and *drukknum* 'drunk' appear. Andrews (1990a) shows that the subjects of these open adjuncts are functionally controlled by a term argument of the matrix verb. In particular, in (117) the object *Sveini* 'Svein' appears in dative case, as the verb *mœtti* 'met' requires, and the SUBJ of the dative adjunct *drukknum* 'drunk' is controlled by the matrix dative OBJ *Sveini*.

We propose the rule in (118) for XADJ in Icelandic, which allows for the XADJ phrase *drukknum* 'drunk' to be adjoined to a VP:[8]

(118)
$$\text{VP} \longrightarrow \begin{pmatrix}\text{VP}\\ \uparrow=\downarrow\end{pmatrix} \qquad \begin{array}{c}\text{AP}^*\\ \downarrow\in(\uparrow\ \text{XADJ})\\ (\uparrow\ \{\text{SUBJ}|\text{OBJ}|\text{OBJ}_\theta\}) = (\downarrow\ \text{SUBJ})\end{array}$$

[7] Andrews (1990a) attributes these examples to Rögnvaldsson (1984).

[8] It is unclear whether the XADJ is actually adjoined to VP in Icelandic or to a higher c-structure position. In either case, its functional annotations are as shown in (118).

According to this rule, the SUBJ of the open adjunct XADJ is identified with a term argument of the matrix clause: a SUBJ, OBJ, or OBJ_θ. We also propose the lexical entry in (119) for the adjective *drukknum* 'drunk', which requires a dative subject:

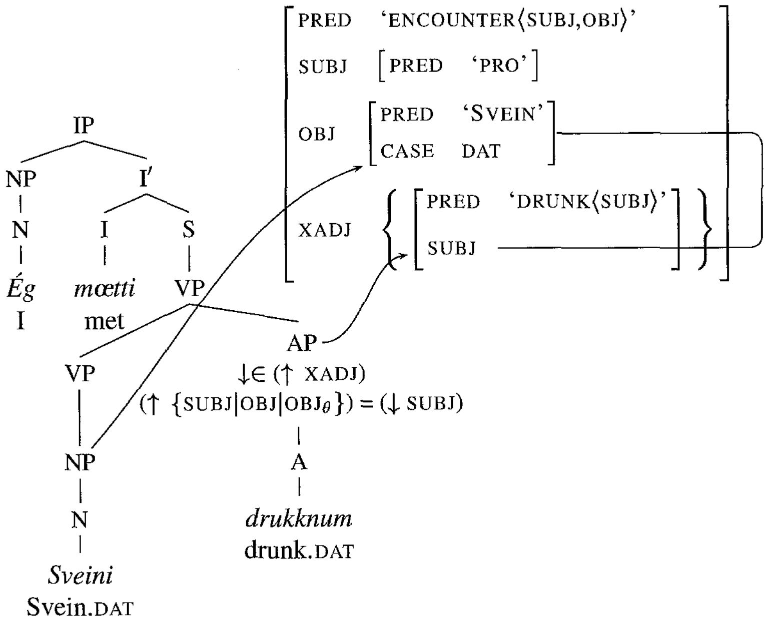
(119) *drukknum* A (↑ PRED) = 'DRUNK⟨SUBJ⟩'
(↑ SUBJ CASE) = DAT

The case constraint imposed by *drukknum* is satisfied by the dative noun phrase *Sveini*:

(120) *Sveini* N (↑ PRED) = 'SVEIN'
(↑ CASE) = DAT

For example (117), the c-structure and f-structure in (121) result:

(121) *Ég mœtti <u>Sveini</u> <u>drukknum</u>*
I met Svein.DAT drunk.DAT
'I met Svein drunk.'


PRED 'ENCOUNTER⟨SUBJ,OBJ⟩'
SUBJ [PRED 'PRO']
OBJ [PRED 'SVEIN' CASE DAT]
XADJ { [PRED 'DRUNK⟨SUBJ⟩' SUBJ] }
IP
NP
I′
N
I
S
Ég
mœtti
VP
I
met
VP
AP
↓∈ (↑ XADJ)
(↑ {SUBJ|OBJ|OBJθ}) = (↓ SUBJ)
NP
A
N
drukknum
drunk.DAT
Sveini
Svein.DAT

8.2. Open Adjuncts and Semantic Composition

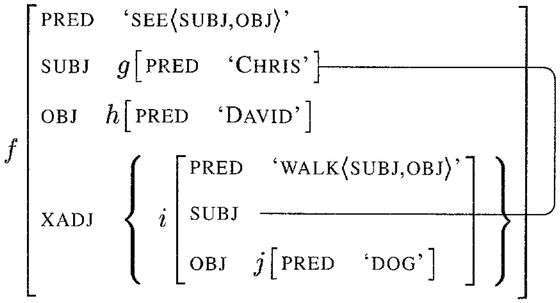
Examples of functional control in constructions involving XADJ are also found in English. In a sentence like *Walking the dog, Chris saw David*, the SUBJ of the adjunct *walking the dog* is functionally controlled by the SUBJ of the matrix clause *Chris* (Bresnan 1982a). We propose the representation in (122) of the meaning of this example:

(122) *Walking the dog, Chris saw David.*

$\mathit{during}(\mathit{walk}(\mathit{Chris}, \mathit{dog}))(\mathit{see}(\mathit{Chris}, \mathit{David}))$

Like other adjuncts, an open adjunct XADJ such as *walking the dog* combines with the clause it modifies, producing a new modified meaning of the same semantic type. In (122), the predicate *during* relates the interval at which the subordinate clause event occurs to the interval at which the main clause event occurs, and the sentence means that during the interval at which Chris was walking the dog, the event of Chris seeing David occurred. The f-structure and meaning constructor for example (122) are given in (123):

(123) *Walking the dog, Chris saw David.*

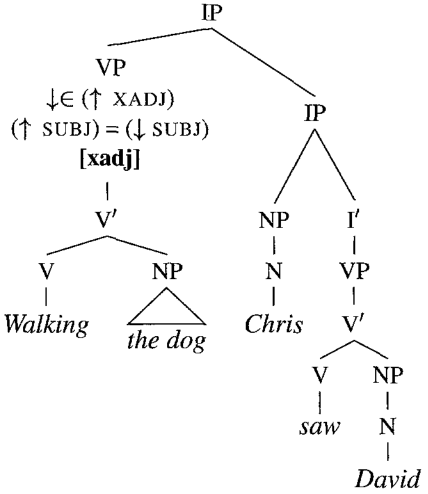


$\mathit{during}(\mathit{walk}(\mathit{Chris}, \mathit{dog}))(\mathit{see}(\mathit{Chris}, \mathit{David})) : f_\sigma$

What linguistic element contributes the *during* predicate in this example? As discussed in Chapter 9, Section 6, meanings can be contributed not only by the words of a sentence, but also by certain syntactic constructions. In this case, the information that the event of Chris seeing David occurred during the time at which Chris was walking the dog is not contributed by any of the words in either the subordinate or the main clause. Instead, the meaning is contributed by the phrase structure configuration associated with this construction: the meaning constructor associated with the *during* predicate appears on the c-structure rule associated with functionally controlled XADJ adjuncts. In other languages, this meaning might be expressed constructionally, as in English, or by morphological or lexical means.

The annotated c-structure tree for example (122) is given in (124):

(124) *Walking the dog, Chris saw David.*



The rule that gives rise to this c-structure is:

$$\text{(125) IP} \longrightarrow \begin{pmatrix} \text{VP} \\ \downarrow\in(\uparrow \textsc{xadj}) \\ (\uparrow \textsc{subj}) = (\downarrow \textsc{subj}) \\ \textbf{[xadj]} \end{pmatrix} \begin{pmatrix} \text{IP} \\ \uparrow=\downarrow \end{pmatrix}$$

The first daughter VP in the rule in (125) has three annotations that are crucial for our current discussion. The set-membership expression $\downarrow\in(\uparrow \textsc{xadj})$ requires the f-structure for the VP to appear as a member of the XADJ set of the mother IP. The equation $(\uparrow \textsc{subj})=(\downarrow \textsc{subj})$ means that the SUBJ of the XADJ phrase *walking the dog* is the same as the SUBJ of the matrix clause *Chris*. The third annotation, **[xadj]**, abbreviates the meaning constructor in (126):

$$\text{(126) } \textbf{[xadj]} \quad \lambda P.\lambda Q.\lambda X.\mathit{during}(P(X))(Q(X)) : \\ [(\downarrow \textsc{subj})_\sigma \multimap \downarrow_\sigma] \multimap [[(\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma] \multimap [(\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma]]$$

Instantiating the variables in this meaning constructor according to the f-structure labels in (124), we have the instantiated meaning constructor labeled **[xadj]** in (127):

$$\text{(127) } \textbf{[xadj]} \quad \lambda P.\lambda Q.\lambda X.\mathit{during}(P(X))(Q(X)) : \\ [g_\sigma \multimap i_\sigma] \multimap [[g_\sigma \multimap f_\sigma] \multimap [g_\sigma \multimap f_\sigma]]$$

The right-hand side of this meaning constructor requires two implicational resources, $[g_\sigma \multimap i_\sigma]$ and $[g_\sigma \multimap f_\sigma]$, to produce a meaning resource $[g_\sigma \multimap f_\sigma]$. The meaning resource $[g_\sigma \multimap i_\sigma]$ represents a resource i_σ for the XADJ that has not yet combined with its subject g_σ — in other words, $[g_\sigma \multimap i_\sigma]$ is a function from the subject meaning g_σ to the XADJ meaning i_σ. Similarly, $[g_\sigma \multimap f_\sigma]$ represents a main clause meaning "missing" its subject g_σ. When both of these resources are found, a new resource $[g_\sigma \multimap f_\sigma]$ is produced, reflecting a matrix clause meaning that is modified by the participial adjunct *walking the dog*. On the left-hand side, two arguments P and Q are required; each of these arguments is applied to the subject meaning X to produce a modified meaning $\mathit{during}(P(X))(Q(X))$ for the entire sentence.

The meaning constructor premises in (128) are relevant in the derivation of the meaning of this sentence; we have simplified the meaning contribution of the phrase *the dog*, representing it simply as the constant *dog*:

(128) Meaning constructor premises for *Walking the dog, Chris saw David*:

$$\begin{array}{lrl}
\textbf{[walk]} & \lambda X.\lambda Y.\mathit{walk}(Y,X) & : \; j_\sigma \multimap [g_\sigma \multimap i_\sigma] \\
\textbf{[dog]} & \mathit{dog} & : \; j_\sigma \\
\textbf{[see]} & \lambda X.\lambda Y.\mathit{see}(Y,X) & : \; h_\sigma \multimap [g_\sigma \multimap f_\sigma] \\
\textbf{[Chris]} & \mathit{Chris} & : \; g_\sigma \\
\textbf{[David]} & \mathit{David} & : \; h_\sigma \\
\textbf{[xadj]} & \lambda P.\lambda Q.\lambda X.\mathit{during}(P(X))(Q(X)) & : \\
 & \multicolumn{2}{r}{[g_\sigma \multimap i_\sigma] \multimap [[g_\sigma \multimap f_\sigma] \multimap [g_\sigma \multimap f_\sigma]]}
\end{array}$$

We begin by combining the premises labeled **[walk]** and **[dog]**, producing the meaning constructor labeled **[walk-dog]** in (129):

(129) **[walk-dog]** $\lambda Y.\mathit{walk}(Y, \mathit{dog}) : g_\sigma \multimap i_\sigma$

This meaning constructor provides the resource $g_\sigma \multimap i_\sigma$ required by the meaning constructor **[xadj]**. Combining **[xadj]** and **[walk-dog]**, we have the meaning constructor labeled **[xadj-walk-dog]** in (130):

(130) **[xadj-walk-dog]**

$$\lambda Q.\lambda X.\mathit{during}(\mathit{walk}(X, \mathit{dog}))(Q(X)) : [[g_\sigma \multimap f_\sigma] \multimap [g_\sigma \multimap f_\sigma]]$$

Next, we combine the premises labeled **[see]** and **[David]**, producing the meaning constructor **[see-David]** in (131):

(131) **[see-David]** $\lambda Y.\mathit{see}(Y, \mathit{David}) : g_\sigma \multimap f_\sigma$

This meaning constructor provides the resource needed by **[xadj-walk-dog]**. We combine **[xadj-walk-dog]** with **[see-David]** to produce the meaning constructor labeled **[xadj-walk-dog-see-David]** in (132):

(132) **[xadj-walk-dog-see-David]**

$$\lambda X.\mathit{during}(\mathit{walk}(X, \mathit{dog}))(\mathit{see}(X, \mathit{David})) : g_\sigma \multimap f_\sigma$$

Finally, we combine the meaning constructor in (132) with the remaining meaning constructor **[Chris]**, producing the semantically complete and coherent meaning constructor in (133), as desired:

(133) **[xadj-walk-dog-see-David]**, **[Chris]** $\vdash$

$$\mathit{during}(\mathit{walk}(\mathit{Chris}, \mathit{dog}))(\mathit{see}(\mathit{Chris}, \mathit{David})) : f_\sigma$$

8.3. Anaphoric Control and ADJ

Some adjuncts participate in *obligatory anaphoric control*, where an unexpressed pronominal argument of a clausal adjunct is anaphorically controlled by an argument of the matrix clause. Here we discuss the Warlpiri *kurra* and *karra* constructions, both of which exemplify anaphoric control. In both constructions, the SUBJ of a subordinate adjunct clause with complementizer *kurra* or *karra* is controlled by an argument of the matrix clause: the matrix SUBJ is the controller in the *kurra* construction, and the matrix OBJ is the controller in the *karra* construction.

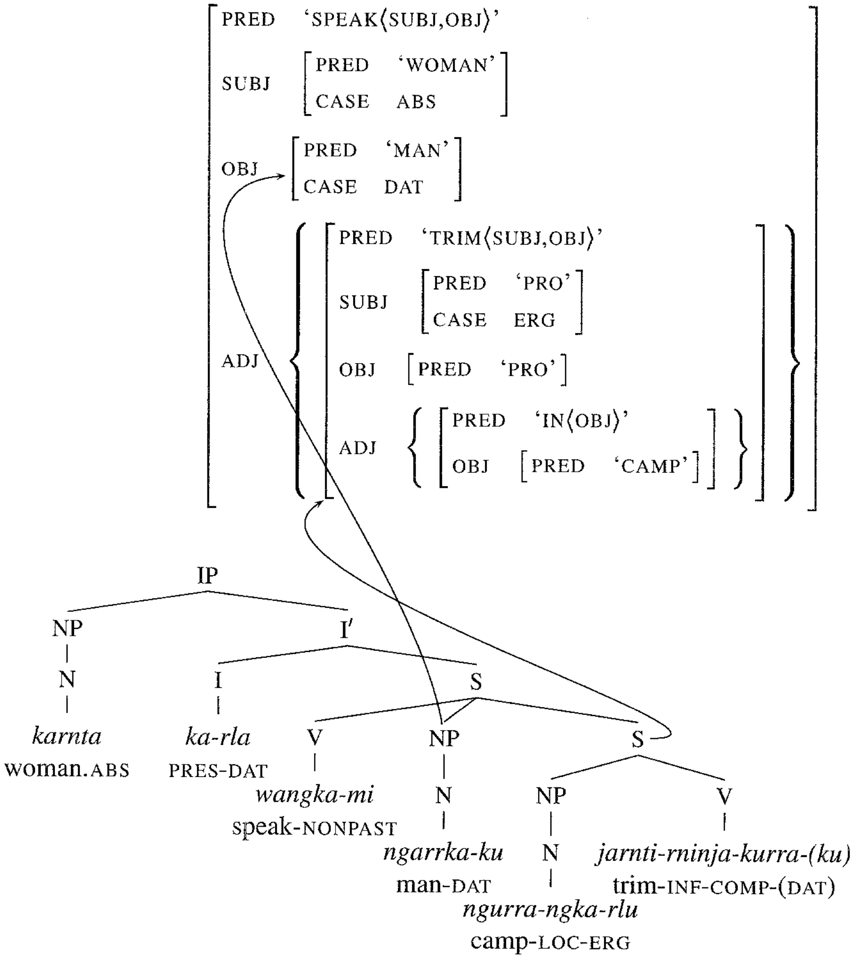
The sentence in (134) (page 355) exemplifies the *kurra*-construction, in which the OBJ of the matrix clause anaphorically controls the SUBJ of the adjunct clause. Simpson and Bresnan (1983) provide evidence from case agreement to demonstrate that the *kurra*-construction involves anaphoric rather than functional control and therefore that the modifying adjunct phrase *ngurra-ngka-rlu jarnti-rninja-kurra-(ku)* 'trimming it in camp' is an ADJ and not an XADJ phrase. As they point out, example (134) contains an adjunct phrase *ngurra-ngka-rlu* 'in camp' which has ERG casemarking *-rlu*. Such adjunct phrases agree in case with the SUBJ of the clause they modify. Here, the phrase *ngurra-ngka-rlu* 'in camp' modifies the subordinate adjunct clause *jarnti-rninja-kurra-(ku)* 'trimming it'. Therefore, the ERG casemarking on the modifier *ngurra-ngka-rlu* 'in camp' shows that the SUBJ of the subordinate clause *jarnti-rninja-kurra-(ku)* 'trimming it' is also ERG.

However, the matrix OBJ phrase *ngarrka-ku* 'man', which anaphorically controls the subordinate SUBJ, is DAT and not ERG, in accordance with the case requirements imposed by the matrix verb *wangka-mi* 'speak'. This difference in CASE requirements between the matrix controller and the subordinate clause controllee shows that the f-structures of the controller and the controllee are different and that anaphoric and not functional control is involved.

8.4. Controlled Adjuncts and Semantic Composition

Like the *kurra* construction, the Warlpiri *karra* construction involves anaphoric control of the subordinate clause SUBJ by an argument of the matrix clause. However, Simpson and Bresnan (1983) show that the *karra* construction contrasts with the *kurra* construction in its requirements on the controller-controllee relation: in

(134) *karnta* *ka-rla* *wangka-mi* *ngarrka-ku [ngurra-ngka-rlu*
woman.ABS PRES-DAT speak-NONPAST man-DAT camp-LOC-ERG
jarnti-rninja-kurra-(ku)]
trim-INF-COMP-(DAT)
'The woman is speaking to the man (while he is) trimming it in camp.'



the *karra* construction, the SUBJ of the adjunct clause is anaphorically controlled by the matrix SUBJ, not the OBJ:

(135) *ngarrka ka wirnpirli-mi [karli jarnti-rninja-karra]*
man.ABS PRES whistle-NONPAST boomerang.ABS trim-INF-COMP
'The man is whistling while trimming a boomerang.'

In example (135) the SUBJ of the adjunct phrase is anaphorically controlled by the matrix SUBJ *ngarrka* 'man', as required by the affix *-karra* on the subordinate clause verb *jarnti-* 'trim', and the example means that the man is whistling while he is trimming a boomerang. We represent the meaning of example (135) as in (136), using the same *during* predicate used in Section 8.2 of this chapter:

(136) $\mathit{during}(\mathit{trim}(\mathit{man}, \mathit{b\text{-}rang}))(\mathit{whistle}(\mathit{man}))$

The verbal affix *karra* contributes the meaning that the interval corresponding to the event of trimming the boomerang occurs during the interval corresponding to the whistling event. Since we are focusing on the semantics of control, we have simplified the representation in (136) of the meanings of the noun phrases *ngarrka* 'man' and *karli* 'boomerang', representing them as the individual constants *man* and *b-rang*.

The f-structure and meaning constructor for example (135) are given in (137). Since this example involves anaphoric control, we must provide contextual resources to allow the unexpressed pronominal subject of the verb *jarnti-rninja-karra* 'trim' to be resolved; therefore, we assume the context-meaning pair meaning constructors introduced in Chapter 11 in the analysis of this example:

(137) *ngarrka ka wirnpirli-mi [karli jarnti-rninja-karra]*
man.ABS PRES whistle-NONPAST boomerang.ABS trim-INF-COMP
'The man is whistling while trimming a boomerang.'

$$f\left[\begin{array}{ll}
\text{PRED} & \text{`WHISTLE}\langle\text{SUBJ}\rangle\text{'}\\
\text{SUBJ} & g\left[\begin{array}{ll}\text{PRED} & \text{`MAN'}\\ \text{CASE} & \text{ABS}\end{array}\right]\\
\text{ADJ} & \left\{ h\left[\begin{array}{ll}
\text{PRED} & \text{`TRIM}\langle\text{SUBJ,OBJ}\rangle\text{'}\\
\text{SUBJ} & i\left[\begin{array}{ll}\text{PRED} & \text{`PRO'}\\ \text{CASE} & \text{ERG}\end{array}\right]\\
\text{OBJ} & j\left[\begin{array}{ll}\text{PRED} & \text{`BOOMERANG'}\end{array}\right]
\end{array}\right]\right\}
\end{array}\right]$$

$$[\mathit{during}(\mathit{trim}(\mathit{man}, \mathit{b\text{-}rang}))(\mathit{whistle}(\mathit{man})), \langle \mathit{b\text{-}rang}, \mathit{man}, \mathit{man}\rangle] : \; f_\sigma \otimes \langle j_\sigma, i_\sigma, g_\sigma \rangle$$

The lexical entry for *jarnti-rninja-karra* ‘trim’ is given in (138):[9]

(138) *jarnti-rninja-karra* $(\uparrow \textsc{pred}) = \text{‘}\textsc{trim}\langle\textsc{subj,obj}\rangle\text{’}$

$$\lambda X.\lambda Y.\mathit{trim}(X,Y) : (\uparrow \textsc{obj})_\sigma \multimap [(\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma]$$

$$\lambda P.\lambda Q.\mathit{during}(P)(Q) : \uparrow_\sigma \multimap [(\textsc{adj} \in \uparrow)_\sigma \multimap (\textsc{adj} \in \uparrow)_\sigma]$$

$$(\uparrow \textsc{subj pred}) = \text{‘}\textsc{pro}\text{’}$$

$$((\uparrow \textsc{subj})_\sigma \textsc{ antecedent}) = ((\textsc{adj} \in \uparrow) \textsc{ subj})_\sigma$$

[pro]

The first line of this lexical entry specifies the PRED value for the verb *jarnti-rninja-karra* ‘trim’, and the second line provides the *trim* predicate: like other transitive verbs, this verb requires a meaning for its OBJ and a meaning for its SUBJ in order to produce a meaning for the entire sentence.

The third line of this lexical entry represents part of the meaning contribution of the affix *karra*, the *during* predicate:

(139) $\lambda P.\lambda Q.\mathit{during}(P)(Q) : \uparrow_\sigma \multimap [(\textsc{adj} \in \uparrow)_\sigma \multimap (\textsc{adj} \in \uparrow)_\sigma]$

On the right-hand side, this meaning constructor requires a meaning resource $\uparrow_\sigma$ (to be provided by the meaning constructor in the second line of the lexical entry when its requirements are satisfied) and a meaning $(\textsc{adj} \in \uparrow)_\sigma$ for the clause it modifies. When these are provided, a new resource $(\textsc{adj} \in \uparrow)_\sigma$ results, corresponding to the modified main clause meaning. On the left-hand side, P corresponds to the meaning of the ADJ clause, and Q corresponds to the unmodified meaning of the main clause.

The fourth line of this lexical entry provides the ‘PRO’ value for the PRED of the SUBJ. The fifth line establishes the meaning resource of the SUBJ of the matrix clause, $((\textsc{adj} \in \uparrow) \textsc{ subj})_\sigma$, as the antecedent of the pronominal SUBJ of the adjunct clause $(\uparrow \textsc{subj})_\sigma$:

(140) $((\uparrow \textsc{subj})_\sigma \textsc{ antecedent}) = ((\textsc{adj} \in \uparrow) \textsc{ subj})_\sigma$

Finally, the sixth line represents a pronominal meaning constructor **[pro]** for the SUBJ of *jarnti-rninja-karra* ‘trim’, defined in (141):

(141) **[pro]** $\lambda C.[\mathit{first}(C), \langle \mathit{first}(C), C\rangle] :$

$$\forall C.\langle((\uparrow \textsc{subj})_\sigma \textsc{ antecedent}), C\rangle \multimap$$
$$(\uparrow \textsc{subj})_\sigma \otimes \langle(\uparrow \textsc{subj})_\sigma, ((\uparrow \textsc{subj})_\sigma \textsc{ antecedent}), C\rangle$$

Instantiating this meaning constructor according to the f-structure labels in (137), we have:

[9] The expression $(\textsc{adj} \in \uparrow)$ exemplifies inside-out functional uncertainty, discussed in Chapter 6, Section 1.2; the use of the set membership symbol $\in$ as an attribute is discussed in Chapter 6, Section 2.1.

(142) **[pro]** $\lambda C.[\mathit{first}(C), \langle \mathit{first}(C), C\rangle] : \forall C.\langle g_\sigma, C\rangle \multimap [i_\sigma \otimes \langle i_\sigma, g_\sigma, C\rangle]$

Like the pronominal meaning constructors discussed in Chapter 11, Section 3.4.2, the right-hand side of this meaning constructor requires a context argument of the form $\langle g_\sigma, C\rangle$ — a context in which the antecedent g_σ is available. When this context is consumed, a meaning-context pair is produced: the meaning resource for the pronominal, i_σ, and a new augumented context to which the pronominal meaning resource i_σ has been added. On the left-hand side, a context C must be provided as an argument. The first member of the resulting meaning-context pair is the meaning $\mathit{first}(C)$ corresponding to the meaning of the antecedent, which is thereby assigned as the meaning of the pronominal; the second member of the pair is an updated context, where the pronoun's meaning $\mathit{first}(C)$ is added to the original context C.

The meaning constructors in (143) (page 358) are relevant in the analysis of example (135):

(143) Meaning constructor premises for *ngarrka ka wirnpirli-mi karli jarnti-rninja-karra*:

$$\begin{array}{lrcl}
\textbf{[context]} & \langle\rangle & : & \langle\rangle \\
\textbf{[whistle]} & \lambda X.\mathit{whistle}(X) & : & g_\sigma \multimap f_\sigma \\
\textbf{[man]} & \lambda C.[\mathit{man}, \langle \mathit{man}, C\rangle] & : & \forall C.C \multimap [g_\sigma \otimes \langle g_\sigma, C\rangle] \\
\textbf{[trim]} & \lambda X.\lambda Y.\mathit{trim}(Y,X) & : & j_\sigma \multimap [i_\sigma \multimap h_\sigma] \\
\textbf{[during]} & \lambda P.\lambda Q.\mathit{during}(P)(Q) & : & h_\sigma \multimap [f_\sigma \multimap f_\sigma] \\
\textbf{[pro]} & \lambda C.[\mathit{first}(C), \langle \mathit{first}(C), C\rangle] & : & \forall C.\langle g_\sigma, C\rangle \multimap [i_\sigma \otimes \langle i_\sigma, g_\sigma, C\rangle] \\
\textbf{[boomerang]} & \lambda C.[\mathit{b\text{-}rang}, \langle \mathit{b\text{-}rang}, C\rangle] & : & \forall C.C \multimap [j_\sigma \otimes \langle j_\sigma, C\rangle]
\end{array}$$

We begin by combining the premises labeled **[context]** and **[man]**, producing the meaning constructor labeled **[context-man]** in (144):

(144) **[context-man]** $[\mathit{man}, \langle \mathit{man}\rangle] : g_\sigma \otimes \langle g_\sigma\rangle$

We use the rule **Context split** (Chapter 11, page 297) to split this meaning constructor into a context resource **[newcontext]** and a meaning constructor **[man2]**:

(145) **[newcontext]** $\langle \mathit{man}\rangle : \langle g_\sigma\rangle$

[man2] $\mathit{man} : g_\sigma$

We combine the context resource **[newcontext]** with the pronoun resource **[pro]** for the anaphorically controlled subject of the XADJ clause, producing the meaning constructor **[context-pro]** in (146):

(146) **[context-pro]** $[\mathit{man}, \langle \mathit{man}, \mathit{man}\rangle] : i_\sigma \otimes \langle i_\sigma, g_\sigma\rangle$

We again use the deduction rule **Context split** to split the meaning constructor **[resolved-pro]** into the meaning constructors **[newcontext2]** and **[resolved-pro]** in (147):

(147) **[newcontext2]** $\langle \mathit{man}, \mathit{man} \rangle \ : \ \langle i_\sigma, g_\sigma \rangle$

[resolved-pro] $\mathit{man} \ : \ i_\sigma$

We combine **[resolved-pro]** with **[trim]**, producing **[pro-trim]**:

(148) **[pro-trim]** $\lambda X.\mathit{trim}(\mathit{man}, X) : i_\sigma \multimap h_\sigma$

Using the premises **[pro-trim]**, **[boomerang]**, and **[newcontext2]** along the lines described in Chapter 11, Section 3.4.2, we produce the meaning constructor labeled **[pro-trim-boomerang]** in (149):

(149) **[pro-trim-boomerang]**

$$[\mathit{trim}(\mathit{man}, \textit{b-rang}), \langle \textit{b-rang}, \mathit{man}, \mathit{man} \rangle] \ : \ f_\sigma \otimes \langle j_\sigma, i_\sigma, g_\sigma \rangle$$

Using the deduction rule **Context split**, we produce the meaning constructors **[newcontext3]** and **[pro-trim-boomerang2]** in (150):

(150) **[newcontext3]** $\langle \textit{b-rang}, \mathit{man}, \mathit{man} \rangle \ : \ \langle j_\sigma, i_\sigma, g_\sigma \rangle$

[pro-trim-boomerang2] $\mathit{trim}(\mathit{man}, \textit{b-rang}) \ : \ f_\sigma$

We also combine **[man2]** with **[whistle]** to produce the meaning constructor labeled **[man-whistle]** in (151):

(151) **[man-whistle]** $\mathit{whistle}(\mathit{man}) : f_\sigma$

Combining **[during]**, **[pro-trim-boomerang2]** and **[man-whistle]**, we have the meaning constructor labeled **[trim-whistle]** in (152):

(152) **[trim-whistle]** $\mathit{during}(\mathit{trim}(\mathit{man}, \textit{b-rang}))(\mathit{whistle}(\mathit{man})) : f_\sigma$

Finally, we use the deduction rule **Context merge** to combine **[trim-whistle]** with **[newcontext3]**, producing the desired result:

(153) **[trim-whistle]**, **[newcontext3]** $\vdash$

$$[\mathit{during}(\mathit{trim}(\mathit{man}, \textit{b-rang}))(\mathit{whistle}(\mathit{man})), \langle \textit{b-rang}, \mathit{man}, \mathit{man} \rangle] : f_\sigma \otimes \langle j_\sigma, i_\sigma, g_\sigma \rangle$$

In this way, constraints on the antecedent of the unexpressed pronominal subject of the ADJ clause are enforced.

9. FURTHER READING AND RELATED ISSUES

The syntax of functional and anaphoric control has been a central topic of LFG research from the inception of the theory; besides the work discussed in this chapter, important work exploring the nature of control crosslinguistically was done by Neidle (1982), Andrews (1982), and Mohanan (1983).

Research in LFG generally assumes that the open complement in a functional control construction bears the grammatical function XCOMP and that no other grammatical function is an open function. In a very interesting paper, Arka and Simpson (1998) provide evidence that this assumption cannot be universally maintained. As in English and many other languages, the controller in an equi or raising construction can bear the SUBJ role, while the controlled clause can be an XCOMP; alternatively, and unlike English, OBJ controllers are possible with open SUBJ complements. In the Balinese equivalent of a sentence like *Chris wants to leave*, then, the controller *Chris* can bear the OBJ function, with *to leave* bearing the SUBJ function. This unusual situation counterexemplifies previous claims about the syntactic role of the controller and the controlled clause: Jacobson (1990) claims that the controller in a raising construction must be higher on the grammatical function hierarchy than the controllee, precluding an analysis of the open complement in a Balinese control construction as a SUBJ. As Arka and Simpson (1998) show, control constructions in Balinese do not obey this generalization.

13

COORDINATION

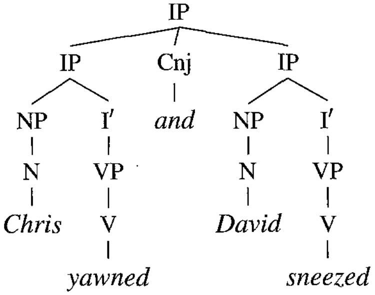
In this chapter, we examine constructions involving coordination. There are a number of important issues involved in the analysis of coordinate structures, and we will only be able to touch on some of the more complex issues. In Section 1 of this chapter, we examine simple sentential coordination; Section 2 discusses coordinate structures involving verbs and other argument-taking predicates. Section 3 discusses properties of coordinate structures, the *nondistributive features* introduced in Chapter 6, Section 2.2. Section 4 presents the theory of nonconstituent coordination. In Section 5, we turn to an examination of the semantics of coordination, which involves some notoriously difficult issues and problems. In some cases, such as sentential coordination, simple predicate conjunction is involved. In other cases involving sharing of arguments, the resource-sensitive nature of our glue language becomes an issue, and a theory of resource sharing is required.

Section 6 examines noun phrase coordination, which differs from sentential coordination in several respects. The coordinate noun phrase has its own syntactic and semantic properties, which may include person, number, and gender features. Semantically, noun phrase coordination involves group formation, and so we also briefly discuss the semantics of plurals, including a brief discussion of conjoined quantifiers.

1. SENTENTIAL COORDINATION

Coordination was first examined in an LFG setting by Bresnan et al. (1985b), and the formal properties of coordination were explored in detail by Kaplan and Maxwell (1988). We begin our discussion with the simple case of sentential coordination, as described by Kaplan and Maxwell (1988). Kaplan and Maxwell propose a constituent structure like the one in (1) for a coordinate sentence like *Chris yawned and David sneezed*:

(1) *Chris yawned and David sneezed.*



There is no limit to the number of conjuncts in a coordinate structure; therefore, as discussed in Chapter 2, Section 3.4, a coordinate structure is represented as a set whose members are the individual conjuncts. The c-structure in (1) corresponds to the following f-structure:

(2) *Chris yawned and David sneezed.*

$$\left\{\begin{array}{l}\begin{bmatrix}\text{PRED} & \text{'YAWN}\langle\text{SUBJ}\rangle\text{'}\\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{'CHRIS'}\end{bmatrix}\end{bmatrix}\\[2ex] \begin{bmatrix}\text{PRED} & \text{'SNEEZE}\langle\text{SUBJ}\rangle\text{'}\\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{'DAVID'}\end{bmatrix}\end{bmatrix}\end{array}\right\}$$

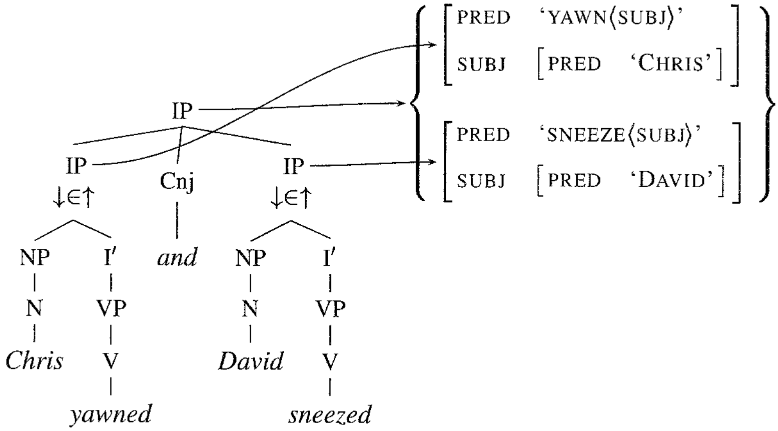
The c-structure and f-structure in (1–2) are constrained by the following rule for IP coordination:

(3) $$\text{IP} \longrightarrow \underset{\downarrow\in\uparrow}{\text{IP}^{+}} \quad \text{Cnj} \quad \underset{\downarrow\in\uparrow}{\text{IP}}$$

This rule makes use of the *Kleene plus operator* +, which licenses one or more occurrences of IP. Thus, this rule allows a coordinate IP to consist of one or more IPs, followed by a conjunction, followed by the final IP conjunct.[1]

The functional annotations on this rule specify that each f-structure corresponding to an IP conjunct is a member of the set of f-structures corresponding to the mother IP; *set descriptions* are discussed in more detail in Chapter 6, Section 2. The diagram in (4) shows the relation between the c-structure and the f-structure of the example under discussion:

(4) *Chris yawned and David sneezed.*

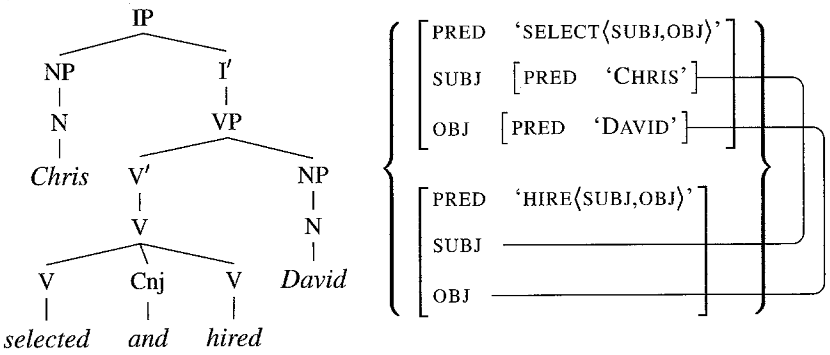


2. PREDICATE COORDINATION

When unsaturated predicates are coordinated, the situation is more complex: coordinated verbs often share some arguments, and Completeness and Coherence requirements must be satisfied for each verb. In example (5), the verbs *selected* and *hired* are transitive, and to meet completeness and coherence requirements each must have a SUBJ and an OBJ. In the c-structure and f-structure shown in (5), this requirement is met:

[1] This formulation of the coordination rule assumes that the daughters in a coordinate structure are not optional, unlike the situation with other phrase structure rules; at least one conjunct must appear before the conjunction, and one constituent appears after the conjunction.

(5) *Chris selected and hired David.*



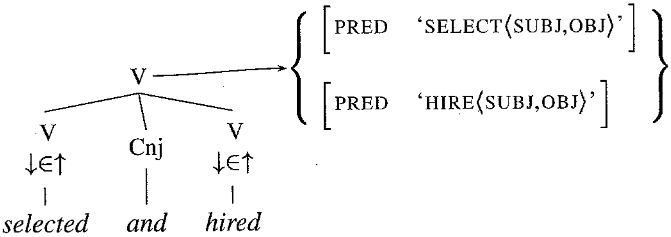
The phrase structure rule in (6) is used for verb coordination:

(6) $$\mathrm{V} \longrightarrow \underset{\downarrow\in\uparrow}{\mathrm{V}^{+}} \quad \mathrm{Cnj} \quad \underset{\downarrow\in\uparrow}{\mathrm{V}}$$

This rule is very similar to the one presented for IP coordination in example (3) of this chapter: these two rules represent a family of coordination rules allowing coordination of various categories.

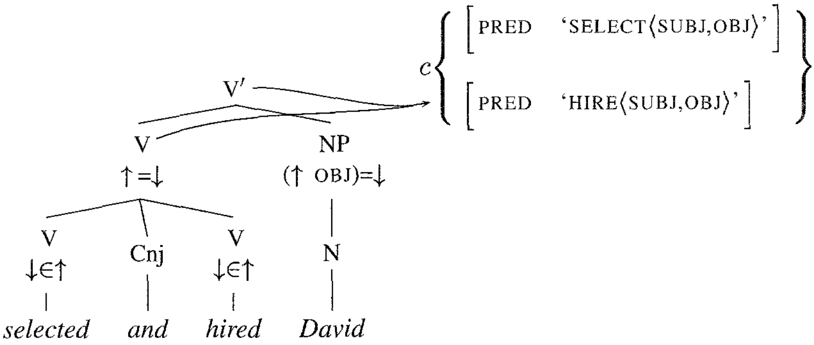
Given the rule in (6), the c-structure and f-structure for the coordinated verbs *selected and hired* are as shown in (7):

(7) *selected and hired*



Both of these verbs are transitive, requiring a SUBJ and an OBJ. In the example under discussion, the coordinate V is the head of V′, and the incomplete f-structure corresponding to V′ is the set labeled *c*, as the annotations in (8) indicate:

(8) *selected and hired David*



In (8), the equation on the NP node dominating *David* refers to the f-structure corresponding to the V′ node, the set *c*, and requires the f-structure for *David* to be the OBJ of that set. These requirements are summarized in (9), where the f-structure for *David* is labeled *d*, and the annotation on the NP node is instantiated to (c OBJ) = d:

(9) *selected and hired David*

$$c\left\{\begin{array}{l}\left[\text{PRED}\quad \text{`SELECT}\langle\text{SUBJ,OBJ}\rangle\text{'}\right]\\[2ex] \left[\text{PRED}\quad \text{`HIRE}\langle\text{SUBJ,OBJ}\rangle\text{'}\right]\end{array}\right\}$$

$$d\left[\text{PRED}\quad \text{`DAVID'}\right]$$

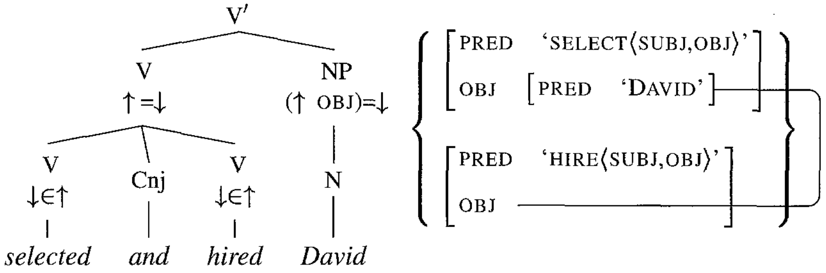
$$(c\ \text{OBJ}) = d$$

Bresnan et al. (1985b) and Dalrymple and Kaplan (2000) provide a definition of function application to sets that allows us to interpret an equation like (c OBJ) = d when c is a set: in such a situation, d is required to be the OBJ of each member of the set. This is because governable grammatical functions like OBJ are *distributive features*, as described in definition (59) of Chapter 6, Section 2.2, repeated here:

(10) If a is a *distributive* feature and s is a set of f-structures, then $(s\ a) = v$ holds if and only if $(f\ a) = v$ for all f-structures f that are members of the set s.

The constraints in (10) entail that d is the OBJ of each f-structure in c:

(11) *selected and hired David*



By the same reasoning, *Chris* is the SUBJ of the set *c* in example (5) and is thus the SUBJ of each member of *c*.

The definition in (10) makes a welcome prediction: two verbs can only be coordinated if they share the same syntactic argument structure. Hall (1965, page 66) provides the following example to illustrate this point:

(12) a. *John washes and polishes his car in the garage.*

b. **John washes and keeps his car in the garage.*

In example (12a), both *wash* and *polish* are transitive verbs, and *in the garage* is interpreted as a locative adjunct; the syntactic requirements of both verbs are satisfied, and the example is wellformed. In contrast, the verb *keep* in example (12b) requires both an OBJ and a locative oblique phrase OBL$_{\text{LOC}}$. According to the theory of feature distribution over sets of f-structures, the phrase *in the garage* cannot be analyzed as a adjunct modifier phrase in the first conjunct and an oblique argument in the second, but must bear the same grammatical function in both conjuncts. If the phrase is assigned a modifier role, the requirements of the verb *keep* are not satisfied, and the sentence is incomplete; if it is assigned the OBL$_{\text{LOC}}$ role, the verb *wash* acquires a locative oblique argument which it does not subcategorize for, and the sentence is incoherent. This accounts for the ungrammaticality of example (12b).

3. SYNTACTIC PROPERTIES OF COORDINATE STRUCTURES

Coordinate structures are special in that the coordinate structure as a whole may have its own properties distinct from the properties of its elements. The syntactic features that a set can have are *nondistributive features*. The behavior of nondistributive features is given in definition (59) of Chapter 6, Section 2.2, repeated here:

(13) If a is a *nondistributive* feature, then $(f\ a) = v$ holds if and only if the pair $\langle a, v\rangle \in f$.

We now propose a more complete version of the annotated phrase structure rule for coordinated verbs; this rule allows the analysis of sentences like *Chris both yawned and sneezed*, and associates the information contributed by the conjunction and any preconjunctions like *both* or *either* with the f-structure for the coordinate phrase:

(14) $$\text{V} \longrightarrow \begin{pmatrix}\text{PreCnj}\\ \uparrow=\downarrow\end{pmatrix} \quad \underset{\downarrow\in\uparrow}{\text{V}^{+}} \quad \underset{\uparrow=\downarrow}{\text{Cnj}} \quad \underset{\downarrow\in\uparrow}{\text{V}}$$

We propose the following lexical entries for *both* and *and*:

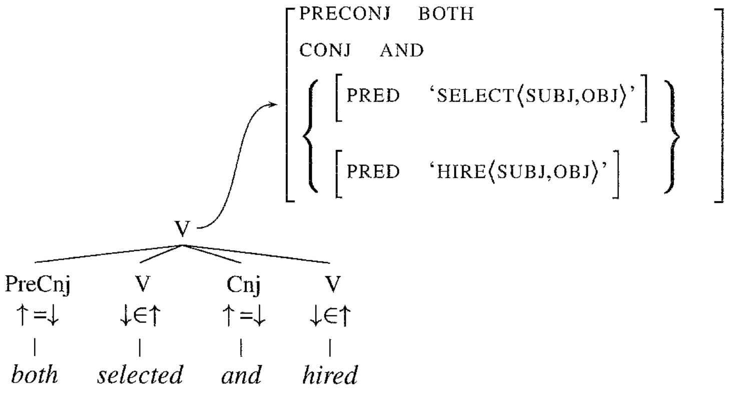
(15) *both* PreCnj $(\uparrow$ PRECONJ$)$ = BOTH
$(\uparrow$ CONJ$) =_c$ AND

and Cnj $(\uparrow$ CONJ$)$ = AND

The features PRECONJ and CONJ are classified as nondistributive features. The entry for *both* contains a constraining equation ensuring that it appears only when the CONJ value of the coordinate phrase is AND, accounting for the ungrammaticality of a phrase such as **both selected or hired.*

Given these lexical entries, the c-structure and f-structure for the phrase *both selected and hired* are displayed in (16):

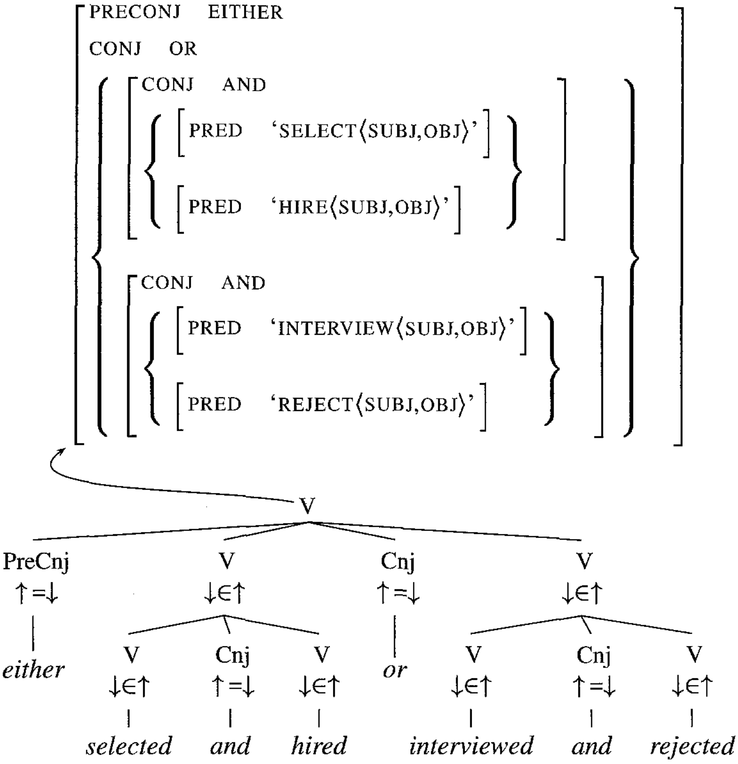
(16) *both selected and hired*



In (16), the PRECONJ and CONJ features are attributes of the coordinate structure, as required.

The coordination rule given in (14) also permits more complex nested coordinate structures like *either [selected and hired] or [interviewed and rejected]*, which has the following c-structure and f-structure analysis:

(17) *either selected and hired or interviewed and rejected*



4. NONCONSTITUENT COORDINATION

Linguists have often considered coordination to be a reliable indicator of constituenthood; as discussed in Chapter 3, Section 1, Radford (1981) proposes that a string of words is a constituent if "it can be coordinated with another similar string." However, constructions involving nonconstituent coordination show that the situation is actually more complex. Strings that are clearly not phrase structure constituents can be coordinated, as in example (18):

(18) *David introduced [[Chris] [to Tracy]] and [[Matty] [to Ken]].*

In example (18), the sequence *Chris to Tracy* is not a phrase structure constituent, but nevertheless it can be coordinated with the sequence *Matty to Ken*, which is also not a constituent.

Maxwell and Manning (1996) propose a theory of nonconstituent coordination that accounts for the grammaticality of examples like (18). The intuition behind their account is that example (18) is acceptable because *Chris to Tracy* constitutes a valid completion of the VP constituent beginning with *introduced*, and the phrase *Matty to Ken* is also a valid completion of such a VP. Maxwell and Manning's theory captures this intuition by allowing phrase structure rules to be split and rules of coordination to refer to the partial constituents that are described by these partial rules.

4.1. Constituent Structure Constraints

To illustrate Maxwell and Manning's approach, we provisionally assume the following simplified phrase structure rule for the English VP:

(19) VP $\longrightarrow$ V NP PP

Maxwell and Manning propose that we can think of the right side of this rule as being divided into two parts, which they label VP-x for the first part and x-VP for the second part:

(20) VP $\longrightarrow$ VP-x x-VP

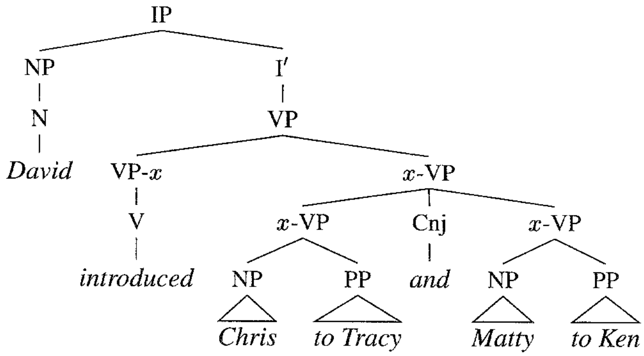
To analyze example (18), we assume that the first half of the VP rule analyzes V, and the second half analyzes the sequence NP PP:

(21) VP-x $\longrightarrow$ V

x-VP $\longrightarrow$ NP PP

Crucially, rules of coordination can refer to the partial phrase structure constituents that result from splitting rules in this way. This allows the following c-structure analysis of example (18):

(22) *David introduced Chris to Tracy and Matty to Ken.*



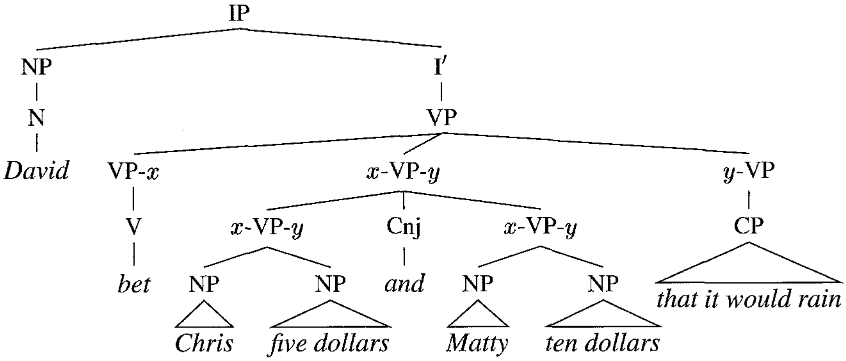
On Maxwell and Manning's theory, any phrase structure rule can be broken up into parts in this way. However, the only rules that can refer to these partial phrase structure constituents are rules of coordination. Therefore, the partial constituents that result from rule splitting play no other role in the grammar besides their role in the analysis of nonconstituent coordination.

In analyzing slightly more complex examples, a c-structure rule may be broken into more than two pieces. For instance, to analyze example (24), we assume the rule in (23):

(23) VP $\longrightarrow$ V NP NP CP

We then break the VP into three parts. The first part of the VP, VP-x, analyzes the V; the second part, x-VP-y, analyzes the sequence NP NP; and the third part, y-VP, analyzes CP:

(24) *David bet Chris five dollars and Matty ten dollars that it would rain.*



Other examples show greater degrees of complexity. Examples (22) and (24) involve partial constituents that are all dominated by a single mother; Maxwell and Manning also discuss examples in which more than one rule is involved in the split, treating these examples by the use of a stack of partial constituents that must be combined in a compatible way.

In formal terms, Maxwell and Manning (1996) state their proposal by reference to the state of the *finite-state automaton* that corresponds to the regular expression on the right-hand side of a phrase structure rule.[2] In constructions involving nonconstituent coordination, the automaton can stop in a particular state in a phrase

[2] A *finite-state automaton* is a formal machine that advances through a string, moving from state to state as the string is traversed. If the string is a member of the language of the regular expression corresponding to the automaton, the automaton will be in a final state when the end of the string is reached. An automaton corresponding to the right-hand side of an LFG phrase structure rule advances through the daughter categories it encounters, moving from state to state as the daughter phrases are analyzed.

structure rule and can then continue from that state to analyze each conjunct of a coordinate phrase. As Maxwell and Manning point out, another way of thinking of the theory is in terms of the regular expression that appears on the right-hand side of a phrase structure rule; on this view, the partial phrase structure constituents that are involved in nonconstituent coordination must be members of the suffix language of the regular expression representing the right-hand side of a phrase structure rule, where the prefix consists of the phrase structure categories that precede the coordinated subconstituent. In other words, each conjunct in a coordinate structure must constitute a valid expansion of the mother category.

Maxwell and Manning's analysis of nonconstituent coordination has a number of desirable properties. For instance, it allows a natural treatment of cases where each conjunct contains a different number of constituents. The only constituent structure requirement in an example like (25) is that each conjunct must constitute a valid completion of the VP rule, and different numbers of phrases are allowed:

(25) *You can call me [directly] or [[after three p.m.] [through my secretary]].*

As pointed out by Milward (1994), such cases are problematic for some other approaches to coordination, particularly the "3-D" approaches of Goodall (1987) and Moltmann (1992).

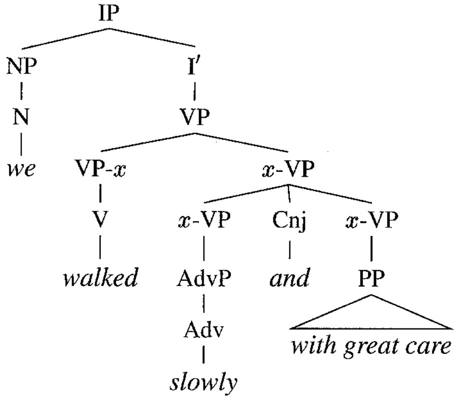
The approach also solves another long-standing problem in the syntax of coordination. Sag et al. (1985) discuss coordination of unlike constituents, providing examples like:

(26) a. *We walked [slowly] and [with great care].* [AdvP and PP]

b. *Terry turned out to be [longwinded] and [a bully].* [AP and NP]

In both of these examples, each conjunct of the coordinate phrase is an acceptable continuation of the VP in which it appears, and Maxwell and Manning's treatment of nonconstituent coordination extends unproblematically to such cases. An example like (26a) has the phrasal analysis given in (27):

(27) *We walked slowly and with great care.*



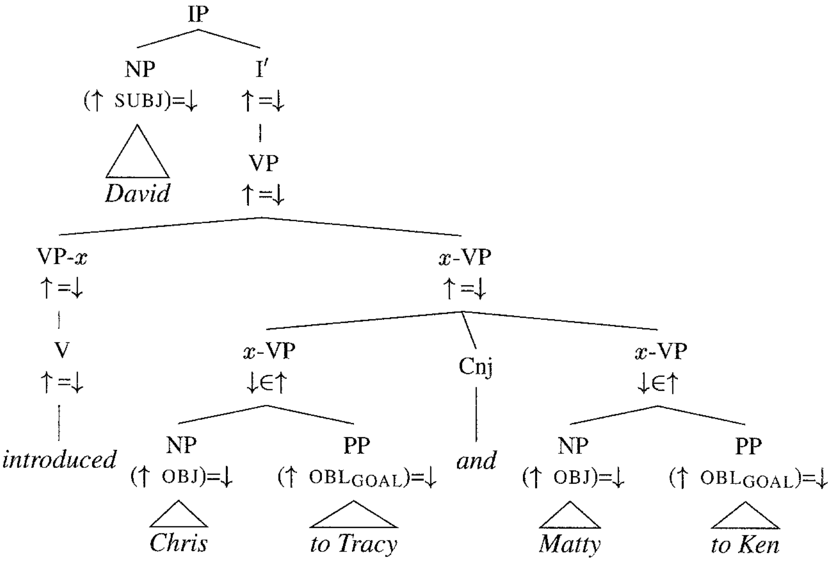
4.2. Functional Annotations

Another desirable property of Maxwell and Manning's analysis is that no special stipulations are required concerning the functional structures of constructions involving nonconstituent coordinations; the rules we have outlined so far give the desired result for all of the examples of nonconstituent coordination that we have discussed. We assume the standard functional annotations on phrase structure rules that we have discussed so far; we also impose the intuitively reasonable requirement that the f-structures of the subconstituent parts of a split constituent are the same as the f-structure for the full constituent, as the annotations in rule (28) indicate.

(28) $$\mathrm{VP} \longrightarrow \underset{\uparrow=\downarrow}{\mathrm{VP}\text{-}x} \quad \underset{\uparrow=\downarrow}{x\text{-VP}}$$

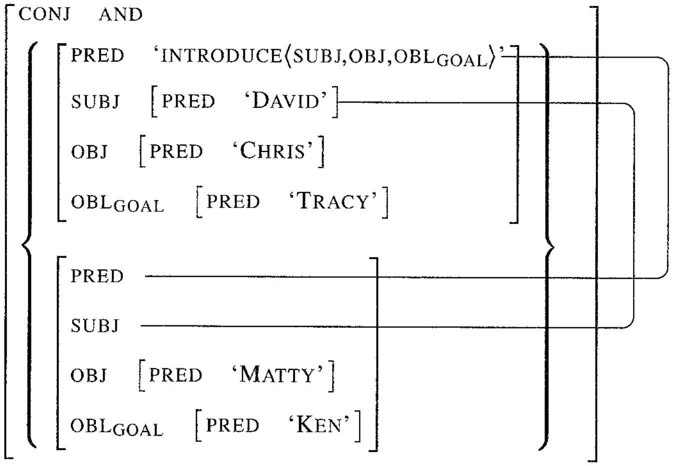
Under these assumptions, the annotated c-structure for example (18) is:

(29) *David introduced Chris to Tracy and Matty to Ken.*



Instantiating the annotations in (29), we have the f-structure in (30), as desired:

(30) *David introduced Chris to Tracy and Matty to Ken.*



5. COORDINATION AND SEMANTIC COMPOSITION

5.1. Sentential Coordination

The meaning of a coordinated sentence like *Chris yawned and David sneezed* is represented simply as the conjunction of the meanings of the conjuncts:

(31) *Chris yawned and David sneezed.*

$$\textit{yawn}(\textit{Chris}) \wedge \textit{sneeze}(\textit{David})$$

For the sentence *Chris yawned and David sneezed* to be true, it must be the case both that Chris yawned and that David sneezed. We assume the f-structure and meaning constructor given in (32) for this example:

(32) *Chris yawned and David sneezed.*

$$f\left[\begin{array}{ll}\text{CONJ} & \text{AND}\\ \multicolumn{2}{l}{\left\{\begin{array}{l} g\left[\begin{array}{ll}\text{PRED} & \text{`YAWN}\langle\text{SUBJ}\rangle\text{'}\\ \text{SUBJ} & h\left[\begin{array}{ll}\text{PRED} & \text{`CHRIS'}\end{array}\right]\end{array}\right]\\ \\ i\left[\begin{array}{ll}\text{PRED} & \text{`SNEEZE}\langle\text{SUBJ}\rangle\text{'}\\ \text{SUBJ} & j\left[\begin{array}{ll}\text{PRED} & \text{`DAVID'}\end{array}\right]\end{array}\right]\end{array}\right\}}\end{array}\right]$$

$$\textit{yawn}(\textit{Chris}) \wedge \textit{sneeze}(\textit{David}) : f_\sigma$$

In this example, the conjunction *and* contributes the requirement that both of the propositions corresponding to the conjuncts must hold. The meaning constructor in (33) imposes this requirement:

(33) **[and]** $\lambda X.\lambda Y.X \wedge Y : (\uparrow\ \in)_{\sigma\langle t\rangle} \multimap [(\uparrow\ \in)_{\sigma\langle t\rangle} \multimap \uparrow_{\sigma\langle t\rangle}]$

On the right-hand side of this expression, the set membership symbol $\in$ is used as an attribute (Chapter 6, Section 2.1); the expression requires as arguments two semantic resources corresponding to two elements of the set represented by $\uparrow$. These resources must be of semantic type t, as indicated by the $\langle t\rangle$ subscript annotation on the semantic structures (Chapter 9, Section 4.1.3). When those two resources are consumed, a semantic resource for the set $\uparrow_\sigma$ is produced. On the left-hand side, the meanings X and Y of the two elements of the set are conjoined.

The meaning constructor in (33) is used in the analysis of example (32), where the coordinate structure has two conjuncts. To analyze coordinate structures with more than two conjuncts, another rule is needed. Kehler et al. (1995) propose the following additional meaning constructor contribution for coordinate structures, also associated with the conjunction *and*:

(34) **[and2]** $\lambda X.\lambda Y.X \wedge Y : ![(\uparrow\ \in)_{\sigma\langle t\rangle} \multimap [\uparrow_{\sigma\langle t\rangle} \multimap \uparrow_{\sigma\langle t\rangle}]]$

The meaning constructor in (34) contains a new expression, the linear logic *of course* operator, written as an exclamation point. When this operator is prefixed to an expression, resource accounting is *turned off* for that expression, meaning that the expression may be used once, more than once, or not at all.

The right-hand side of the meaning constructor in (34) requires two arguments: the semantic resource corresponding to some conjunct in the coordinate structure and the semantic resource for the coordinate set itself. When these resources are consumed, a new semantic resource for the coordinate structure is produced. On the left-hand side, the meaning for the coordinate structure is obtained by conjoining the meaning of the consumed conjunct with the meaning of the previously analyzed coordinate structure.

In the analysis of example (32), **[and2]** will not be used, since the meaning constructor **[and]** is appropriate in constructions like (32) with two conjuncts. In a coordinate construction with three conjuncts, **[and]** must be used, since its meaning must be consumed for a semantically complete and coherent derivation to result; additionally, **[and2]** is used once, to combine the third conjunct with the first two. In a coordinate construction with four conjuncts, **[and2]** is used twice, and so on.

We propose the lexical entry in (35) for *and*, containing the two meaning constructors defined in (33) and (34):

(35) *and* Cnj $(\uparrow \text{CONJ}) = \text{AND}$
[and]
[and2]

The instantiated meaning constructor premise **[and]** is shown in (36), together with the meaning constructor contributions of the other words in the sentence:

(36) Meaning constructor premises for *Chris yawned and David sneezed*:

[yawn] $\lambda X.\mathit{yawn}(X) : h_\sigma \multimap g_\sigma$

[Chris] $\mathit{Chris} : h_\sigma$

[sneeze] $\lambda X.\mathit{sneeze}(X) : j_\sigma \multimap i_\sigma$

[David] $\mathit{David} : j_\sigma$

[and] $\lambda X.\lambda Y.X \wedge Y : g_\sigma \multimap [i_\sigma \multimap f_\sigma]$

We first combine the meaning constructors for **[yawn]** and **[Chris]** and for **[sneeze]** and **[David]**:

(37) **[yawn]**, **[Chris]** $\vdash \mathit{yawn}(\mathit{Chris}) : g_\sigma$

[sneeze], **[David]** $\vdash \mathit{sneeze}(\mathit{David}) : i_\sigma$

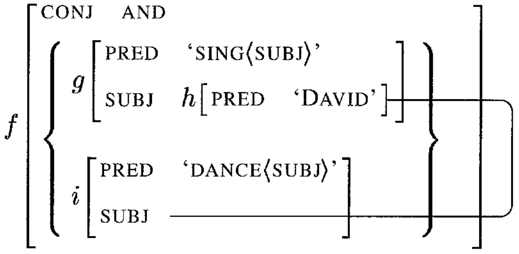
We then combine these results with **[and]**, obtaining the semantically complete and coherent meaning constructor shown in (38):

(38) $\mathit{yawn}(\mathit{Chris}) \wedge \mathit{sneeze}(\mathit{David}) : f_\sigma$

5.2. Subsentential Coordination

As with sentential coordination, coordination of subsentential units involves conjunction of the meanings of the conjunct phrases. The f-structure and meaning constructor for the sentence *David sang and danced* are given in (39):

(39) *David sang and danced.*



$\mathit{sing}(\mathit{David}) \wedge \mathit{dance}(\mathit{David}) : f_\sigma$

In this example, the subject *David* of the conjoined verbs *sang* and *danced* is shared across the coordinate structure, as described in Section 2 of this chapter. Argument sharing in subsentential coordination presents a special challenge for a resource-based account of the syntax-semantics interface.

In example (39), the subject *David* is shared by the verbs *sang* and *danced*, and each verb requires a meaning contribution from its subject. Our theory of meaning assembly and the syntax-semantics interface relies crucially on the assumption that the meaning constructor for *David* makes a single, unique meaning contribution. Clearly, however, the acceptability of example (39) entails that this single meaning contribution can satisfy the requirements imposed by each of the verbs in a coordinate structure. Therefore, in the analysis of examples like (39), we require a theory of resource management in argument sharing that accounts for the grammaticality of examples like *David sang and danced* while maintaining the desirable properties of our linear-logic-based glue approach.

In fact, reliance on a theory of resource management in the analysis of examples like (39) is of paramount importance: the acceptability of example (39) does *not* indicate that resource accounting is completely abandoned for the shared argument *David*. If resource accounting were switched off entirely for shared arguments — for example, by prefixing the meaning constructor for *David* with the linear logic *of course* operator — we would have no way of accounting for the unacceptability of examples like (40):

(40) **Chris selected and sang David.*

This example is syntactically and semantically incoherent. Syntactic requirements on verb coordination, discussed in Section 2 of this chapter, entail that *David* must appear as the OBJ argument of the verbs *selected* and *sang*. However, the intransitive verb *sang* does not require an OBJ resource, and so if an OBJ resource is provided, it will not be consumed in the meaning derivation and will remain unused at the end of the deduction, leading to semantic incoherence. If resource accounting were switched off for *David*, it could contribute one semantic resource to this sentence, and not the expected two. This would satisfy the requirements of the verb *selected*, violate no requirements imposed by the verb *sang*, and incorrectly result in a semantically coherent deduction. This example shows that resource accounting must in fact be enforced for example (40), as it is in all other cases. We require, then, a complete and explicit theory of resource accounting, argument sharing, and their interactions.

Particular care must be taken in the treatment of certain kinds of arguments shared across conjunctions. As noted by Partee (1970), a sentence like *Someone sang and danced* has the meaning represented in (41):

(41) *Someone sang and danced.*

$$\mathsf{a}(X, \mathsf{person}(X), \mathsf{sing}(X) \wedge \mathsf{dance}(X))$$

Here, the single quantifier *someone* scopes over the coordinate structure, and the variable X bound by the quantifier appears as an argument of both sing and dance. This meaning is not the same as the one for the sentence *Someone sang and someone danced*:

(42) *Someone sang and someone danced.*

$$\mathsf{a}(X, \mathsf{person}(X), \mathsf{sing}(X)) \wedge \mathsf{a}(Y, \mathsf{person}(Y), \mathsf{dance}(Y))$$

In (42), different people are involved in each activity, while example (41) requires that there is a single person who both sang and danced. This fact must also be captured by our theory of argument sharing and semantic composition.

Within the glue approach, two sorts of proposals have been made for the treatment of constructions involving argument sharing. In the following, we describe these two approaches and present some arguments bearing on the choice between them. As yet, we do not have a full understanding of the complete range of issues involved in choosing between these two approaches, and so a complete theory of subsentential coordination and sharing within the glue framework must be left for future research.

Kehler et al. (1995) were the first to propose a treatment of argument sharing and resource management within the glue approach. Their proposal appeals to the geometry of f-structures in constructions involving argument sharing: intuitively, their approach focuses on *occurrences* of f-structures, where an f-structure occurs more than once if there is more than one path leading to it. Semantic resources are associated with *paths* through the f-structure and thus with occurrences of f-structures; in essence, Kehler et al. (1995) provide a means for making one copy of a semantic resource for each f-structure path leading to its corresponding f-structure.

For example, in the analysis of example (39), there are two paths leading to the f-structure for *David*, since the f-structure for *David* appears as the value of two different SUBJ attributes. On Kehler et al.'s analysis, each verb requires a semantic resource associated with the path leading to its SUBJ. Since the f-structure for *David* appears at the end of each of these two paths, two copies of the meaning constructor for *David* are made available, and a semantically complete and coherent meaning deduction results.

Kehler et al. (1995) also discuss resource sharing and quantification, noting that their proposal produces the correct results for cases like (41), where a quantified noun phrase is shared. For more complicated cases of coordinate structures involving intensional verbs, which take quantifiers as arguments, they provide a special rule for quantifier duplication, whose use is restricted by a processing strategy along the lines suggested by Partee and Rooth (1983).

This approach successfully accounts for the acceptability of examples involving argument sharing by allowing the creation of as many semantic resources as needed to satisfy the requirements of each predicate involved in the sharing of a single argument. However, a major problem with this approach is that it also allows resource duplication in cases where such duplication is unwarranted. For example, constructions with raising verbs also exhibit argument sharing: the subject of a verb like *seem* is also the subject of its XCOMP (Chapter 12, Section 1). However, as pointed out by Asudeh (2000a), we do not wish to enforce resource duplication in this case; as discussed in Chapter 12, Section 2.2, the derivation of the meaning $\mathit{seem}(\mathit{yawn}(\mathit{David}))$ of a sentence like *David seemed to yawn* requires exactly one occurrence of the meaning resource contributed by *David*, not two. Similarly, resource duplication is not warranted in constructions in which an f-structure bears the TOPIC or FOCUS function as well as an argument function, even though the same f-structure appears at the end of the TOPIC or FOCUS path as well as a path associated with another grammatical function. In these cases, relying simply on the geometry of the f-structure to license feature duplication leads to the wrong result; a more constrained theory is needed.

A second approach to resource management in constructions involving argument sharing resembles approaches often adopted in Categorial Grammar (Partee and Rooth 1983; Steedman 1996); in a glue setting, the approach has been explored most extensively in unpublished work by Ash Asudeh and Dick Crouch (Asudeh 2000b provides a brief overview). This approach provides special rules for resource management in situations where arguments are shared. These rules combine the semantic requirements imposed by each predicate that shares an argument into a requirement for a single semantic resource.

For example, a rule like the one in (43) is proposed for coordinating verbs that share a SUBJ and each require a semantic contribution from their SUBJ. The rule combines the requirements of each conjunct into a requirement for a single semantic resource provided by the shared subject:

(43)
$$f\left[\begin{array}{ll}\text{CONJ} & \text{AND}\\ & \left\{\begin{array}{l} g\left[\text{SUBJ}\quad h[\;]\right]\\ i\left[\text{SUBJ}\quad\;\right]\end{array}\right\}\end{array}\right]$$

$$\lambda P.\lambda Q.\lambda X.P(X) \wedge Q(X) : [h_\sigma \multimap g_\sigma] \multimap [[h_\sigma \multimap i_\sigma] \multimap [h_\sigma \multimap f_\sigma]]$$

The right-hand side of this meaning constructor requires two arguments corresponding to the two conjuncts g_σ and i_σ, each "missing" its subject h_σ: the first argument is $[h_\sigma \multimap g_\sigma]$ and the second argument is $[h_\sigma \multimap i_\sigma]$. The result is a resource of the form $[h_\sigma \multimap f_\sigma]$, a resource for the coordinate structure still "missing" its subject. Crucially, a single semantic resource h_σ can satisfy this requirement. On the left-hand side, the meanings P and Q of the two conjuncts are applied to the subject meaning X. Thus, in contrast to the approach described above, this approach manages resource sharing by collapsing multiple requirements for a single resource into a single requirement, rather than by duplicating the single resource so that the multiple requirements can be satisfied. This approach is also successful in deriving the correct meaning for examples like (39) and (41).

This approach does not encounter the difficulties outlined above in cases where argument sharing does not require duplication of resources. No special provisions must be made for raising verbs, for example, as issues of resource management and resource duplication simply do not arise. However, a potential difficulty for the proposal is that it seems to require a large number of rules for handling different cases of argument sharing, where different numbers of arguments, or arguments with different grammatical functions, are shared. Furthermore, as pointed out by Kehler et al. (1995), resource sharing is at issue even in noncoordinate cases; Hudson (1976) provides example (44), where the verbs *support* and *oppose* each require an object, but only one object resource is provided by the noun phrase *two trade bills*:

(44) *Citizens who support, paraded against politicians who oppose, two trade bills.*

The challenge for the second approach, then, is to avoid positing a potentially unbounded number of rules for different circumstances in which resources are shared by determining a general typology and theory of resource sharing in cases where a meaning constructor like the one in (43) must be provided.

6. NOUN PHRASE COORDINATION

6.1. Noun Phrase Coordination and Feature Resolution

As discussed in Chapter 6, Section 2.2, a coordinate noun phrase can have features that are different from the features of the individual conjunct phrases. For example, a coordinate phrase like the Spanish phrase *José y yo* 'José and I' contains a first person conjunct and a third person conjunct, and behaves like a first person plural phrase. This is shown by the requirement for the use of the first person plural form of the verb in example (45):

(45) *José y yo* *hablamos/*habláis/*hablan*
José and I speak.1.PL/*speak.2.PL/*speak.3.PL
'José and I speak.'

The f-structure for the coordinate noun phrase *José y yo* 'José and I' is:

(46) *José y yo*

$$\left[\begin{array}{ll} \multicolumn{2}{l}{\left\{ \begin{array}{l} \left[\begin{array}{ll}\text{PRED} & \text{`JOSÉ'}\\ \text{PERS} & 3\\ \text{NUM} & \text{SG}\end{array}\right] \\ \left[\begin{array}{ll}\text{PRED} & \text{`PRO'}\\ \text{PERS} & 1 \\ \text{NUM} & \text{SG}\end{array}\right]\end{array}\right\}} \\ \text{PERS} & 1\\ \text{NUM} & \text{PL}\\ \text{CONJ} & \text{AND}\end{array}\right]$$

Features associated with the set representing a coordinate structure are *nondistributive* features. The features PERS, NUM, and GEND are nondistributive features, since they can be associated with coordinate as well as noncoordinate noun phrases. The problem of how to determine the PERS and other features of a coordinate phrase has been termed *feature resolution* and has been studied extensively (Corbett 1983, 1991). In many languages, including Spanish and English, the PERS feature resolves in the way indicated in (47):

(47) Person agreement:

first & second = first
first & third = first
second & third = second
third & third = third

As the table in (47) indicates, a coordinate structure with a first person conjunct and a second person conjunct behaves like a first person phrase; one with a second person conjunct and a third person conjunct behaves like a second person phrase; and so on.

Dalrymple and Kaplan (2000) propose a formal theory of feature resolution for the PERS and GEND features that predicts these facts. They represent the values of the PERS feature as sets:

(48) Person values:
{S,H}: first person
{H}: second person
{}: third person

Dalrymple and Kaplan propose that the PERS value of a coordinate set is determined by taking the set union of the PERS values of the conjuncts. The table in (49) represents exactly the same pattern of person agreement in Spanish and English as the table in (47):

(49)
$$\begin{array}{rl}
\{\text{S,H}\}\ \text{(first)} \cup \{\text{H}\}\ \text{(second)} & = \{\text{S,H}\}\ \text{(first)}\\
\{\text{S,H}\}\ \text{(first)} \cup \{\}\ \text{(third)} & = \{\text{S,H}\}\ \text{(first)}\\
\{\text{H}\}\ \text{(second)} \cup \{\}\ \text{(third)} & = \{\text{H}\}\ \text{(second)}\\
\{\}\ \text{(third)} \cup \{\}\ \text{(third)} & = \{\}\ \text{(third)}
\end{array}$$

Within LFG, there is a simple way to impose the requirement that the PERS value of a coordinate set is the union of the PERS values of the conjuncts, represented in the rule in (50):

(50)
$$\text{NP} \longrightarrow \begin{pmatrix}\text{PreCnj}\\ \uparrow=\downarrow\end{pmatrix} \quad \begin{array}{c}\text{NP}^{+}\\ \downarrow\in\uparrow\\ (\downarrow \text{PERS}) \subseteq (\uparrow \text{PERS})\end{array} \quad \begin{array}{c}\text{Cnj}\\ \uparrow=\downarrow\\ \\ \end{array} \quad \begin{array}{c}\text{NP}^{+}\\ \downarrow\in\uparrow\\ (\downarrow \text{PERS}) \subseteq (\uparrow \text{PERS})\end{array}$$

The annotations on the NP conjunct daughters of this rule require each conjunct NP to be a member of the conjunct set of the mother node (the $\downarrow\in\uparrow$ annotation). Additionally, the PERS value of each daughter node is required to be a subset of the PERS value of the mother node (the $(\downarrow \text{PERS}) \subseteq (\uparrow \text{PERS})$ annotation). Recall from our discussion in Chapter 5, Section 2.1 that the f-structure for an utterance is the *minimal solution* to its functional description. For the rule in (50), this means that the PERS value of the mother node is the *smallest* set that has the PERS value of each conjunct daughter as a subset, which corresponds exactly to the union of the PERS values of the conjunct daughters, as desired.

Verb agreement is enforced by means of *constraining equations* (Chapter 5, Section 2.8) like those in (51) for the Spanish first person verb *hablamos* '(we) speak' and third person verb *hablan* '(they) speak':

(51) a. *hablamos*: $(\uparrow \text{SUBJ PERS}) =_c \{\text{S,H}\}$

b. *hablan*: $(\uparrow \text{ SUBJ PERS}) =_c \{\}$

The constraint in (51) requires the subject of the first person plural form *hablamos* to bear the PERS value {S, H}; if the SUBJ of *hablamos* is a coordinate noun phrase, {S,H} must be the smallest set that contains the PERS sets of all of the conjunct daughters. Similarly, the subject of the third person plural form *hablan* must be the empty set {}.

Other languages have a richer system of personal pronouns. For example, Fula and many other languages exhibit a distinction between *inclusive* and *exclusive* first person pronouns: the referent of an inclusive first person pronoun includes the hearer, while the referent of an exclusive first person pronoun excludes the hearer. Dalrymple and Kaplan show that their theory of resolution for the PERS feature extends unproblematically to such languages.

6.2. Semantics of Noun Phrase Coordination

Much work has been done on the semantics of coordinated noun phrases. A clearly presented and useful overview is given by Winter (1998); as he notes, approaches to noun phrase coordination are generally of two types. The first type of analysis assumes the existence of two different lexical entries for coordinating conjunctions like *and*, while the second type (advocated by Winter) assumes that there is only one entry for *and*, whether it coordinates sentences or noun phrases. Our proposal is of the first type: we follow Hoeksema (1988) in assuming that the entry for *and* in noun phrase conjunction is different from the entry for *and* in sentential coordination.

The first type of coordinating conjunction, discussed in Section 5.1 of this chapter, coordinates sentences and other constituents of type t. We used the coordination operator $\wedge$ in the meaning constructor associated with this entry for *and*, often referred to as *Boolean 'and'*. The second type of coordinating conjunction coordinates noun phrases of type e; we will call this *group-forming 'and'*. Though we side with Hoeksema in assuming a distinction between these two types of *and*, an advantage of our proposal is that we do not require the complicated machinery for meaning assembly in quantifier coordination that his analysis requires.

For an example like *David and Chris met*, which involves the coordination of two names, we propose the meaning in (52):

(52) *David and Chris met.*

meet({*David*, *Chris*})

Here we assume that group-forming *and* combines *David* and *Chris*, individuals of type e, to form a group of type e consisting of these two individuals. We represent the group composed of the individuals *David* and *Chris* in curly brackets, as {*David,Chris*}. Research on plurals and group formation offers several possibilities for interpretation of this group-forming operation. If we adopt a lattice-theoretic treatment of plurals, as proposed by Link (1983) (see also Landman 1989), we would treat $\{\mathit{David},\mathit{Chris}\}$ as a special kind of plural individual formed from the atomic individuals *David* and *Chris*; Schwarzschild (1996) provides more discussion of this proposal and an alternative view.

As we will see, this proposal extends straightforwardly to cases in which indefinite singulars are coordinated. Following Hoeksema (1988), we propose the meaning in (53) for the sentence *A student and a professor met*:

(53) *A student and a professor met.*

$$a(X, \mathit{student}(X),\ a(Y, \mathit{professor}(Y), \mathit{meet}(\{X, Y\})))$$

This sentence means that a group consisting of the individuals X and Y met, where X is a student and Y is a professor. We can also coordinate a proper name with an indefinite:

(54) *David and a professor met.*

$$a(Y, \mathit{professor}(Y), \mathit{meet}(\{\mathit{David}, Y\}))$$

Here, the group that met consists of Y, a professor, and the individual *David.*

This approach works well for these simple cases. As noted by Schwarzschild (1996, page 22), complications arise in the treatment of other cases; in particular, the quantifier *no* is problematic:

(55) *No soldier and no officer met.*

$$\mathit{no}(X, \mathit{soldier}(X),\ \mathit{no}(Y, \mathit{officer}(Y), \mathit{meet}(\{X, Y\})))$$

The meaning representation given in (55) can be paraphrased as: ‘Every soldier met some officer’, clearly not a possible meaning for this sentence.[3] In fact, the scoping problem posed by this example can be circumvented if example (55) is analyzed as an instance of *branching quantification* (Barwise 1979), in which neither quantifier takes scope over the other; in fact, many examples that have been taken to exemplify branching quantification involve coordinated noun phrases. We will not attempt to provide a full analysis of these cases here, but will stick to providing an analysis of the simpler cases, in the belief that getting a firm handle on simpler cases of noun phrase coordination will enable a more revealing analysis of the more complex cases. For more discussion of the semantics of noun phrase coordination and plurality, see Krifka (1990), Lasersohn (1995), Schwarzschild (1996), Winter (1998), and the references cited in those works.

[3] A possibly more telling paraphrase of the meaning representation given in example (55) is: ‘There is no soldier that met no officer’.

6.3. Noun Phrase Coordination and Meaning Assembly

The f-structure and meaning constructor for the sentence *David and Chris met* are given in (56):

(56) *David and Chris met.*

$$f\left[\begin{array}{ll} \text{PRED} & \text{`MEET}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & g\left[\begin{array}{ll} \text{NUM} & \text{PL} \\ \text{CONJ} & \text{AND} \\ \multicolumn{2}{l}{\left\{\begin{array}{l} h\left[\text{PRED} \quad \text{`DAVID'}\right] \\ i\left[\text{PRED} \quad \text{`CHRIS'}\right] \end{array}\right\}} \end{array}\right] \end{array}\right]$$

$$\mathit{meet}(\{\mathit{David}, \mathit{Chris}\}) : f_{\sigma}$$

The premise labeled **[g-and]** participates in the analysis of example (56):

(57) **[g-and]** $\lambda X.\lambda Y.\{X, Y\} : (\uparrow\ \in)_{\sigma\langle e\rangle} \multimap [(\uparrow\ \in)_{\sigma\langle e\rangle} \multimap \uparrow_{\sigma\langle e\rangle}]$

The label **[g-and]** is chosen to indicate that this premise represents the contribution of *group-forming 'and'*, which differs from the premise labeled **[and]** given in (33) of this chapter in several respects: **[g-and]** requires arguments of type e, the type of individuals, whereas **[and]** requires arguments of type t, the type of propositions; and **[g-and]** combines these arguments to form a group of individuals which we represent as $\{X, Y\}$, whereas **[and]** conjoins its arguments to form a conjunctive proposition $X \wedge Y$.

In Section 5.1 of this chapter, we noted that a conjunction like *and* not only contributes the premise **[and]** to a meaning deduction, but may also contribute an additional premise **[and2]** in cases of coordination with more than two conjuncts. Analogously, we provide the meaning constructor labeled **[g-and2]** in (58) for cases of noun phrase coordination with more than two conjuncts:

(58) **[g-and2]** $\lambda X.\lambda Y.\{X\} \cup Y : ![(\uparrow\ \in)_{\sigma\langle e\rangle} \multimap [\uparrow_{\sigma\langle e\rangle} \multimap \uparrow_{\sigma\langle e\rangle}]]$

Like the meaning constructor **[and2]**, given in (34) of this chapter, the right-hand side of this meaning constructor is prefixed with the linear logic *of course* operator !, indicating that it can be used any number of times and need not be used at all. In noun phrase coordination constructions with two conjuncts, it will not be used. In constructions with three conjuncts, it will be used once: on the right-hand side, it consumes the resource corresponding to the third conjunct and the resource corresponding to the coordinate structure, producing a new resource for the coordinate structure; on the left-hand side, it adds the meaning of the third conjunct to the group of individuals denoted by the coordinate noun phrase. The lexical entry for *and* contributes both of these meaning constructors to a meaning deduction:

(59) *and* Cnj $(\uparrow \text{CONJ}) = \text{AND}$
[g-and]
[g-and2]

The meaning deduction for example (56) involves the meaning constructor premises given in (60):

(60) Meaning constructor premises for *David and Chris met*:

[meet]	$\lambda X.meet(X)$	$:\ g_{\sigma\langle e\rangle} \multimap f_{\sigma\langle t\rangle}$
[David]	$David$	$:\ h_{\sigma\langle e\rangle}$
[Chris]	$Chris$	$:\ i_{\sigma\langle e\rangle}$
[g-and]	$\lambda X.\lambda Y.\{X,Y\}$	$:\ h_{\sigma\langle e\rangle} \multimap [i_{\sigma\langle e\rangle} \multimap g_{\sigma\langle e\rangle}]$

The premise **[g-and]** requires as its arguments a meaning for h_σ and a meaning for i_σ. These resources are provided by the premises labeled **[David]** and **[Chris]**. Combining **[David]**, **[Chris]**, and **[g-and]**, we obtain the premise labeled **[David-and-Chris]** in (61):

(61) **[David-and-Chris]** $\{David, Chris\} : g_{\sigma\langle e\rangle}$

This meaning constructor provides the semantic resource g_σ of type e, which is exactly what the meaning constructor labeled **[meet]** in (60) requires. Combining **[David-and-Chris]** and **[meet]**, we have the semantically complete and coherent meaning constructor in (62), as desired:

(62) **[David-and-Chris], [meet]** $\vdash$ $meet(\{David, Chris\}) : f_\sigma$

Next we consider example (63), which differs from the example just discussed in that a quantifier is coordinated with a proper name:

(63) *David and a professor met.*

$$f\left[\begin{array}{ll} \text{PRED} & \text{`MEET}\langle\text{SUBJ}\rangle\text{'} \\ \text{SUBJ} & g\left[\begin{array}{l} \begin{array}{ll}\text{NUM} & \text{PL}\end{array} \\ \begin{array}{ll}\text{CONJ} & \text{AND}\end{array} \\ \left\{\begin{array}{l} h\left[\begin{array}{ll}\text{PRED} & \text{`DAVID'}\end{array}\right] \\ i\left[\begin{array}{ll}\text{SPEC} & \left[\begin{array}{ll}\text{PRED} & \text{`A'}\end{array}\right] \\ \text{PRED} & \text{`PROFESSOR'}\end{array}\right]\end{array}\right\}\end{array}\right]\end{array}\right]$$

$a(Y, professor(Y), meet(\{David, Y\})) : f_\sigma$

The meaning constructor premises in (64) are relevant for this example:

(64) Meaning constructor premises for *David and a professor met*:

$$\begin{array}{lrl}
\textbf{[meet]} & \lambda X.\mathit{meet}(X) & : g_\sigma \multimap f_\sigma \\
\textbf{[David]} & \mathit{David} & : h_{\sigma\langle e\rangle} \\
\textbf{[a-professor]} & \lambda T.\mathit{a}(Y, \mathit{professor}(Y), T(Y)) & : \forall H.[i_\sigma \multimap H] \multimap H \\
\textbf{[g-and]} & \lambda X.\lambda Y.\{X, Y\} & : h_\sigma \multimap [i_\sigma \multimap g_\sigma]
\end{array}$$

We first combine the premises **[David]** and **[g-and]**, producing the meaning constructor labeled **[David-and]** in (65):

(65) **[David-and]** $\lambda Y.\{\mathit{David}, Y\} : i_\sigma \multimap g_\sigma$

We now use the abstraction rule given in example (36) of Chapter 9, which permits the introduction and subsequent discharge of a hypothetical premise in the deduction; on the meaning side, hypothetical premise discharge corresponds to abstracting over the variable introduced by the premise. We hypothesize the premise $X : [i_\sigma]$ in the first line of (66), which allows us to perform a deduction using the premises **[David-and]** and **[meet]** to produce the meaning constructor labeled **[David-and-X-meet]** in the final line of (66):

(66)

$$\dfrac{\dfrac{\dfrac{X : [i_\sigma] \qquad \textbf{[David-and]}}{\{\mathit{David}, X\} : g_\sigma} \qquad \textbf{[meet]}}{\mathit{meet}(\{\mathit{David}, X\}) : f_\sigma}}{\textbf{[David-and-X-meet]} \quad \lambda X.\mathit{meet}(\{\mathit{David}, X\}) : i_\sigma \multimap f_\sigma}$$

The meaning constructor **[David-and-X-meet]** provides the implicational resource required by the quantifier **[a-professor]**. Combining **[David-and-X-meet]** and **[a-professor]**, we have the semantically complete and coherent meaning constructor given in (67):

(67) **[David-and-X-meet], [a-professor]** $\vdash$

$$\mathit{a}(Y, \mathit{professor}(Y), \mathit{meet}(\{\mathit{David}, Y\})) : f_\sigma$$

The semantic deduction of an example such as (53), in which two quantifiers are coordinated, proceeds similarly: a hypothetical premise is introduced and discharged in the derivation of the scope meaning for each quantifier.

7. FURTHER READING AND RELATED ISSUES

Besides the seminal work of Bresnan et al. (1985b), Andrews (1983a) contributed early and influential (although unfortunately unpublished) work on coordination in LFG. The theory of nonconstituent coordination developed by Maxwell and Manning (1996) was extended in work by Brun (1996a,b).

Besides a theory of PERS resolution in coordination, Dalrymple and Kaplan (2000) also present a theory of gender resolution; Vincent and Börjars (2000) present an alternative view. Resolution in coordinate structures is also examined by Sadler (1999) in a study of *asymmetric agreement*, constructions in which agreement is determined by features of a single conjunct of a coordinate phrase.

Some LFG work on coordination is based on syntactic assumptions that have since been abandoned. Kaplan and Maxwell (1988) proposed a theory of function application to sets that does not rely on a distinction between distributive and nondistributive features or on the definition presented in (10) for application of a function involving a distributive feature to a set; instead, they propose that the properties of a set are defined as the *generalization* of the properties of its elements (see Chapter 6, Section 3.3 for definition and discussion of generalization). Many of the predictions of this theory are indistinguishable from the theory presented in this chapter. However, Kaplan and Maxwell's theory makes unwanted predictions involving constraining properties of sets and also has difficulty in cases where a set has a property that is different from its conjuncts; see Dalrymple and Kaplan (2000) for more discussion.

14

LONG-DISTANCE DEPENDENCIES

Topicalization constructions, relative clauses, and wh-questions in English and many other languages exemplify *long-distance dependencies*, constructions in which a displaced constituent bears a syntactic function usually associated with some other position in the sentence. For example, in an English topicalization construction like *Chris, David likes*, the displaced initial constituent *Chris* plays two roles: it is the TOPIC of the sentence and the OBJ of the verb *likes*. Section 1 of this chapter discusses the syntax of long-distance dependencies, showing how the syntactic relation is established between the fronted phrase and its within-clause grammatical function. Section 2 discusses cases in which a long-distance dependency is signaled by special morphological marking: in particular, we discuss and analyze long-distance dependencies in Kikuyu, where sentences with long-distance dependencies exhibit a special tonal change.

Since many syntactic constraints in long-distance dependency constructions are definable in terms of the grammatical function of the displaced phrase, f-structural constraints on the relation between a displaced constituent and its within-clause functional role will feature heavily in our syntactic discussion. In earlier LFG work (for example, Kaplan and Bresnan 1982) it was assumed that the relation between a displaced constituent and its corresponding within-clause "gap" was definable in constituent structure terms, using the double arrow notation $\Uparrow$ and $\Downarrow$ to relate two positions in the constituent structure tree. In later work, Kaplan and Zaenen (1989) showed that this treatment made it difficult to account for functional constraints on long-distance dependencies, and the original analysis based on c-structure relations and defined in terms of the $\Uparrow$ and $\Downarrow$ notation was subsequently abandoned; see Dalrymple et al. (1995d) for more discussion of the history of the analysis of long-distance dependencies in LFG. However, even though the primary constraints on long-distance dependencies are formulated in f-structure terms, some LFG analyses assume the existence of *traces*, phonologically null c-structure elements corresponding to the within-clause position of the displaced constituent. In Section 3 we discuss evidence for and against traces, with particular attention to the phenomenon of *weak crossover*.

Finally, we turn to a discussion of the semantics of constructions involving long-distance dependencies. Section 4 discusses the semantics of relative clauses and meaning composition, and Section 5 discusses issues that arise in the semantic treatment of wh-questions.

1. SYNTAX OF LONG-DISTANCE DEPENDENCIES

1.1. Topicalization

We begin our discussion of long-distance dependencies with the English topicalization construction, in which a constituent appears at the beginning of the sentence and is interpreted as the TOPIC of the sentence.[1] Following early transformational analyses of this construction, the displaced TOPIC constituent is sometimes spoken of as having been "fronted" or "extracted," and we will also use this terminology in our discussion. The TOPIC phrase also plays a grammatical role in the clause, according to the *Extended Coherence Condition* (Chapter 7, Section 3), originally proposed by Zaenen (1980) (see also Fassi-Fehri 1988):

(1) Extended Coherence Condition:

FOCUS and TOPIC must be linked to the semantic predicate argument structure of the sentence in which they occur, either by functionally or by anaphorically binding an argument.

[1] In accord with general practice in discussions of topicalization in LFG, we represent theTOPIC as a syntacticized discourse function in the functional structure. Of course, the topicalization construction also has particular discourse effects that are represented at the level of*information structure* (Chapter 7, Section 3).

In the following, we will examine the syntax of the topicalization construction in English. Examination of this construction reveals c-structural constraints on the permitted constituent structure categories of the fronted constituent as well as f-structural constraints on the path relating the TOPIC to its within-clause grammatical function. Constraints on the topicalization construction in other languages may differ: as we will see, other languages may allow a different set of phrasal categories to appear in TOPIC position or may place different constraints on the f-structural relation between the TOPIC and its within-clause role.

1.1.1. Category of the Topicalized Phrase

The fronted phrase in the English topicalization construction may be one of several possible phrase structure categories, as shown in example (2):

(2) a. NP: *Chris, I like.*

b. PP: *To Chris, I gave a book.*

c. AP: *Happy, Chris will never be.*

d. CP: *That Chris was a movie star, I never would have guessed.*

e. VP: *?To leave, we convinced Chris.* (acceptable for some speakers; see Chapter 12, Section 3.2)

The sentences in (2) exemplify the permitted range of topicalized phrase structure categories:

(3) Phrasal category of TOPIC phrases in English: NP, PP, AP, CP, VP

1.1.2. Grammatical Function of the Topicalized Phrase

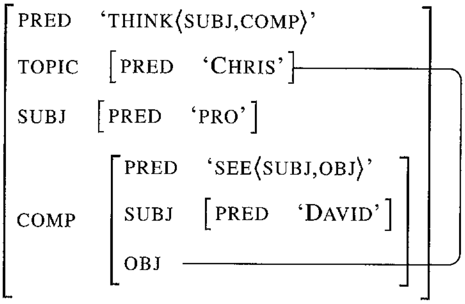
The within-clause grammatical function of the fronted phrase is also constrained: some functions can be related to the TOPIC, whereas others cannot. For instance, the TOPIC can also fill the role of OBJ:

(4) *Chris, we like.*

$$\left[\begin{array}{ll} \text{PRED} & \text{`LIKE}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{TOPIC} & \left[\begin{array}{ll}\text{PRED} & \text{`CHRIS'}\end{array}\right] \\ \text{SUBJ} & \left[\begin{array}{ll}\text{PRED} & \text{`PRO'}\end{array}\right] \\ \text{OBJ} & \text{———} \end{array}\right]$$

In this sentence, *Chris* is the TOPIC of the sentence and also the OBJ of the verb *like*. Longer paths are also acceptable; in example (5), the TOPIC is the OBJ of the subordinate COMP, so that the path through the f-structure leading to the within-clause function of the TOPIC is COMP OBJ:

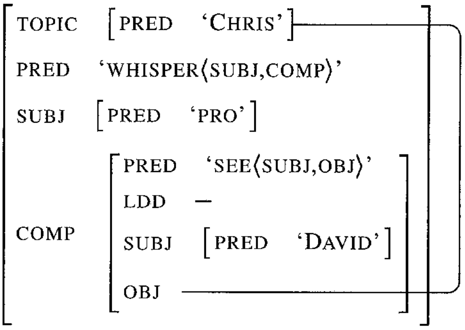
(5) *Chris, we think that David saw.*



Not all within-clause functions can be related to the TOPIC, however. As discussed in Chapter 6, Section 1.4, it is possible for the TOPIC to be related to a position within the COMP of a so-called "bridge verb" like *think*, but not to a position within the COMP of a nonbridge verb like *whisper*. The distinction between bridge and nonbridge verbs is not reflected in the grammatical function of the sentential complement: in both cases, the path to the within-clause argument is COMP OBJ. Instead, this requirement is stated as an additional condition on the f-structures in the extraction domain.

We represent this aspect of the syntax of nonbridge verbs by means of the f-structure attribute LDD (for "long-distance dependency") with value −, which is lexically specified by a nonbridge verb like *whisper* as appearing in its COMP. Such f-structures cannot participate in a long-distance dependency. This accounts for the unacceptability of example (6), since the COMP f-structure contains the attribute-value pair ⟨LDD, −⟩:

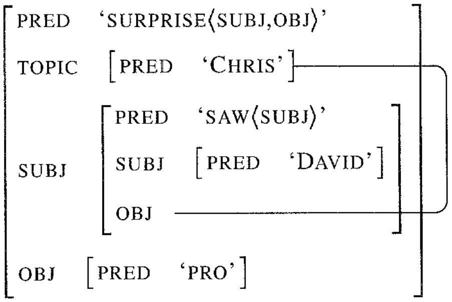
(6) * *Chris, we whispered that David saw.*



In (6), the path relating the TOPIC to its within-clause function goes through the COMP f-structure of the verb *whisper*, which has the value – for the feature LDD. We demonstrate below how this situation is disallowed.

A number of other constraints on long-distance dependencies were originally explored by Ross (1967) and have since been the subject of intense scrutiny and debate. Among these constraints is the Sentential Subject Constraint, according to which a long-distance dependency cannot involve a position inside a sentential subject:

(7) **Chris, that David saw ___ surprised me.*



This constraint is simply stated: the path to the within-clause function of the TOPIC may not include SUBJ. Other constraints on long-distance dependencies are characterized similarly, either as constraints on grammatical functions permitted on the path or as constraints on attributes and values in f-structures through which the path passes.

There is little consensus on the proper characterization of long-distance dependencies involving modifying adjuncts. For example, Williams (1992) claims that examples like (8), which involve a relation between a fronted wh-phrase and a position inside an adverbial modifier, are "marginal though possible":

(8) *?Who did John go to New York after talking to?*

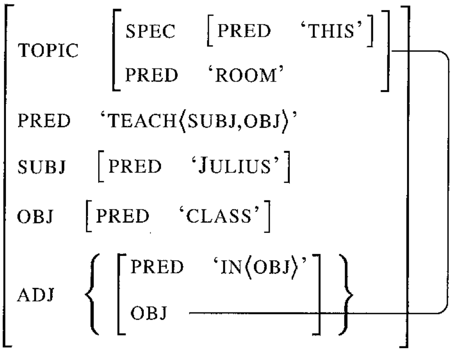
However, Cinque (1990) and Hornstein and Weinberg (1995) count as ungrammatical examples that are very similar to (8):

(9) a. **To whom did you leave without speaking?*

b. **What did John drink cognac after singing?*

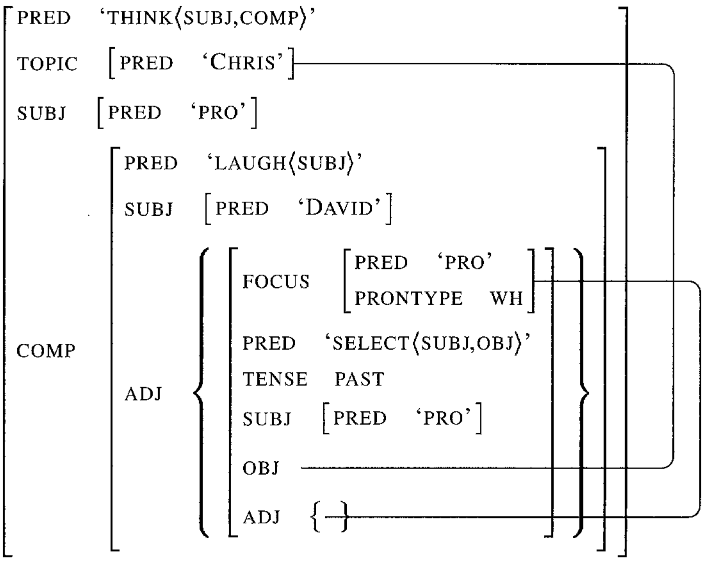
Constraints on long-distance dependencies involving modifying adjuncts are difficult to characterize, and judgements vary idiosyncratically. However, we believe that some basic conclusions can be drawn. First, some dependencies involving modifying adjuncts are acceptable, including example (10):

(10) *This room, Julius teaches his class in.*



Other examples involving a dependency between a TOPIC and a position inside an ADJ are not acceptable. For example, the unacceptability of example (10) shows that extraction from a tensed sentential modifier is not permitted:

(11) **Chris, we think that David laughed when we selected.*



We propose the general characterization of the path to the grammatical function of the TOPIC in the English topicalization construction that is given in (12):

(12) In English, the TOPIC phrase can be related to a grammatical function that is embedded inside any number of XCOMP, COMP, or tensed sentential OBJ functions, as long as the COMP function is governed by a bridge verb, or to a grammatical function inside a possibly embedded nontensed ADJ function.

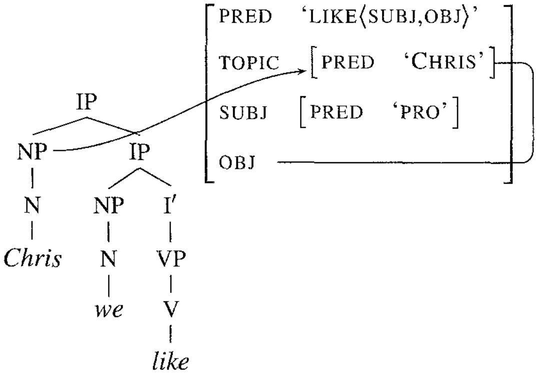
Other languages place different constraints on the topicalization path. For example, Kroeger (1993, Chapter 7) shows that in Tagalog, the topicalization path must consist exclusively of subjects: only a SUBJ may be extracted, and the only function out of which extraction is permitted is SUBJ. This is true not only for the topicalization construction but also for other long-distance dependencies as well; the path in the wh-question construction and the relative clause construction must also contain only SUBJ attributes.

1.1.3. Constituent Structure and Functional Constraints

In Chapter 4, Section 2.2, we discussed the relation between constituent structure specifier positions and discourse functions: in many languages, the discourse functions TOPIC and FOCUS are required to appear in specifier positions of functional categories. King (1995) analyzes the configurational encoding of TOPIC and FOCUS in Russian, showing that the Russian TOPIC appears in the specifier position of IP, and Kroeger (1993) shows that the same is true in Tagalog.

Interestingly, however, the English TOPIC phrase does not appear in specifier position. Bresnan (2001b, Chapter 9) shows that English topics are adjoined to IP, as shown in (13):

(13) *Chris, we like.*



To analyze examples like (13), we propose the phrase structure rule in (14):

(14)
$$\text{IP} \longrightarrow \begin{pmatrix} \text{TopicP} \\ (\uparrow \text{TOPIC}) = \downarrow \\ (\uparrow \text{TOPIC}) = (\uparrow \text{TOPICPATH}) \end{pmatrix} \begin{pmatrix} \text{IP} \\ \uparrow = \downarrow \end{pmatrix}$$

Here we use the *constituent structure metacategory abbreviation* TopicP for the phrase structure category of the fronted phrase; the role of metacategories in syntactic description is discussed in Chapter 5, Section 1.2. We also use the functional abbreviation TopicPath for the path through the f-structure to the within-clause function of the fronted phrase.

The set of phrasal categories that can participate in the topicalization construction in English is given in (3) of this chapter. On this basis, we define TopicP for English in rule (14) as the following disjunction of categories:

(15) TopicP $\equiv$ {NP | PP | VP | AP | CP}

We must also properly constrain the path through the f-structure that relates the TOPIC to its within-clause function. Formally, this relation involves *functional uncertainty*, discussed in Chapter 6, Section 1.1. Example (12) of this chapter outlined a set of constraints on TopicPath, the long-distance path for topicalization in English. We can formally characterize these constraints as in (16):

(16) English TopicPath:

$$\{\textsc{xcomp} \mid \underset{(\rightarrow \textsc{ldd}) \neq -}{\textsc{comp}} \mid \underset{(\rightarrow \textsc{tense})}{\textsc{obj}}\}^* \;\; \{(\textsc{adj} \underset{\neg(\rightarrow \textsc{tense})}{\in}) \;\; (\textsc{gf}) \mid \textsc{gf}\}$$

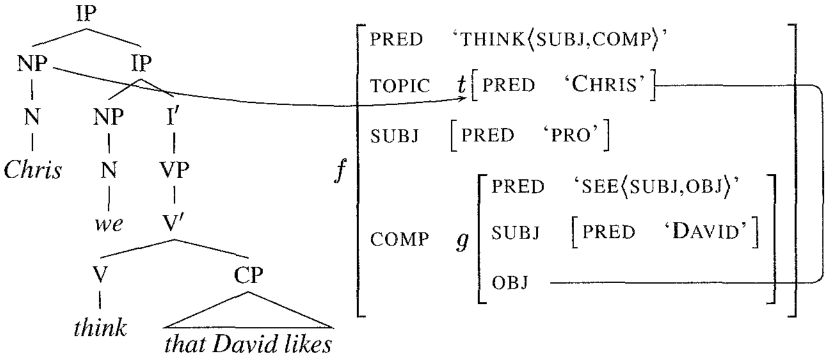
This expression allows the within-clause grammatical function of the TOPIC to be arbitrarily deeply embedded inside any number of properly constrained XCOMP, COMP, or OBJ functions, and optionally to appear as an untensed member of the ADJ set of such a function, or as an argument of the ADJ. The possibility for deeply embedded TOPICS is represented by the *Kleene star operator* * permitting any number of XCOMP, COMP, or OBJ attributes on the path.

In the expression $\underset{(\rightarrow \textsc{ldd}) \neq -}{\textsc{comp}}$, the off-path constraint $(\rightarrow \textsc{ldd}) \neq -$ ensures that the path to the within-clause function of the TOPIC phrase does not involve a nonbridge verb. Recall the use of symbols like $\rightarrow$, defined in (37) of Chapter 6, Section 1.4: the symbol $\rightarrow$ in an off-path constraint on an attribute refers to the f-structure that contains the attribute.

(17) In an expression like $\underset{(\rightarrow s)}{a}$, $\rightarrow$ refers to the value of the attribute a.

The constraint in (16) requires the f-structure in the wellformed example in (18) that is labeled g not to contain the attribute LDD with value $-$:

(18) *Chris, we think that David saw.*



In the analysis of example (18), the following specially tailored version of the annotated rule in (14) is relevant:

(19) $$\text{IP} \longrightarrow \left(\begin{array}{c} \text{NP} \\ (\uparrow \text{TOPIC}) = \downarrow \\ (\uparrow \text{TOPIC}) = (\uparrow \underset{(\rightarrow \text{LDD}) \neq -}{\text{COMP}} \quad \text{OBJ}) \end{array}\right) \left(\begin{array}{c} \text{IP} \\ \uparrow = \downarrow \end{array}\right)$$

In (20), the f-structure metavariables on the NP node in (19) are instantiated appropriately: ↑ is the f-structure f corresponding to the mother IP node in (18); ↓ is the f-structure t corresponding to the daughter NP node; and → in the off-path constraint $(\rightarrow \text{LDD}) \neq -$ is g, the value of the COMP attribute on which the off-path constraint appears. Thus, we have the following constraints:

(20) $$\left(\begin{array}{c} \text{NP} \\ (f\ \text{TOPIC}) = t \\ (f\ \text{TOPIC}) = (f \underset{(g\ \text{LDD}) \neq -}{\text{COMP}} \quad \text{OBJ}) \end{array}\right)$$

This condition is satisfied by the f-structure in (18), and the sentence is grammatical. The definition of TOPICPATH in (16) permits other such paths relating the TOPIC to its within-clause grammatical function, as long as they do not pass through a COMP f-structure bearing the feature LDD with value −.

Similarly, the off-path constraint in the expression $(\text{ADJ} \underset{\neg(\rightarrow \text{TENSE})}{\in})$ rules out tensed adjunct phrases on the path, accounting for the ungrammaticality of examples like (11). In Chapter 6, Section 2.1, we discussed the use of expressions like (ADJ ∈), in which the set membership symbol ∈ is used as an attribute to allow reference to some member of a set of modifying adjuncts ADJ. The off-path constraint ¬(→ TENSE) requires the member of the ADJ set that contains the TOPIC not to contain the attribute TENSE. In other words, the topicalization path can go through a member of the modifier set, but only if the adjunct is not tensed.

As noted earlier, different languages impose different constraints on the topicalization path: Kroeger (1993, Chapter 7) shows that only subjects can participate in long-distance dependencies in Tagalog:

(21) Tagalog TopicPath:

SUBJ^{+}

This expression uses the *Kleene plus* operator + to indicate that TopicPath in Tagalog consists of at least one occurrence of SUBJ: only a SUBJ may be topicalized, and only a SUBJ may contain a TOPIC. LFG research has not yet established a complete typology of possible paths in long-distance dependencies: future research will no doubt reveal more about the possible range of crosslinguistic and cross-constructional variation in long-distance paths like TopicPath as well as universal generalizations concerning constraints on long-distance paths.

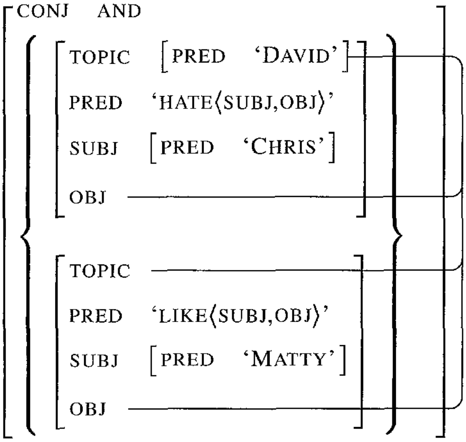
1.1.4. Topicalization in Coordinate Structures

It has long been noted that coordinate structures obey the *across-the-board constraint* (Ross 1967; Williams 1978), according to which a topicalized phrase bears a grammatical function inside one conjunct of a coordinate structure only if it also bears a grammatical function inside the other conjuncts. Example (22) is ungrammatical because the topicalized phrase *David* bears the OBJ function only in the first conjunct of the coordinate sentence:

(22) **David, [Chris hates* ___ *and Matty likes Ursula].*

Instead, the topicalized phrase must bear a grammatical function "across the board," within each conjunct in a coordinate construction, as in example (23):

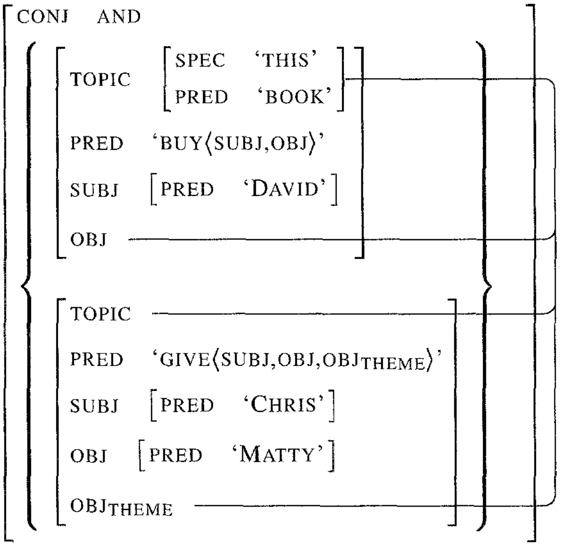
(23) *David, [Chris hates* ___ *and Matty likes* ___*].*



In example (23), the topicalized phrase is the OBJ of both *hates* and *likes.* The across-the-board constraint falls out as a consequence of our theory of coordination and long-distance dependencies. Since grammatical functions are *distributive features* (Chapter 6, Section 2.2), asserting that an f-structure is the OBJ of a set means that it is the OBJ of each member of the set. That is, resolving the path to the within-clause grammatical function of the topic correctly entails that the TOPIC must bear a grammatical function within each conjunct of the coordinate phrase.

A slight refinement of our definition of functional uncertainty is needed, however. Kaplan and Maxwell (1988) discuss examples like (24), in which the topic *David* is the OBJ of the first conjunct and the $\text{OBJ}_{\text{THEME}}$ of the second conjunct:

(24) *This book, [David bought and Chris gave Matty].*



These examples show that the TOPIC can bear different grammatical functions within each conjunct.[2] To analyze these examples, Kaplan and Maxwell (1988) propose a variant definition of functional uncertainty (the original definition was given in (8) of Chapter 6). Kaplan and Maxwell's new definition, presented in (25), accounts for the grammaticality of examples like (24):

[2] Kaplan and Zaenen (1989) discuss constraints on the long-distance path in coordinate structures in Japanese, based on work by Saiki (1985): the grammatical function of theTOPIC must either be SUBJ in all conjuncts, or a nonsubject function in all conjuncts. Similar constraints hold in English:

(a) **Who did Chris think [David met ___] and [___ saw Matty]?*

Kaplan and Zaenen show how TOPICPATH can be defined so as to predict these facts. Falk (2000) also discusses these patterns and provides an alternative analysis.

(25) Functional uncertainty:

If η is a regular expression, then $(f\ \eta) = v$ holds if and only if

$$(f\ a\ \textsc{suff}(a,\eta)) = v$$

for some symbol a, where $\textsc{suff}(a,\eta)$ is the set of suffix strings s such that $as\ \in \eta$.

The effect of this definition is to allow a regular expression representing a path through the f-structure to be resolved differently in each conjunct of a coordinate structure. The definition gives exactly the same result as the one presented in (8) of Chapter 6 for constructions not involving coordination and sets. When a long-distance dependency involves positions inside a set representing a coordinate structure, the definition in (25) allows the as yet unexplored suffix of the regular expression representing the path to be expanded independently within each element of the set. Thus, a valid path must be taken within each conjunct, but a different path may be taken for each element.

1.2. Relative Clauses

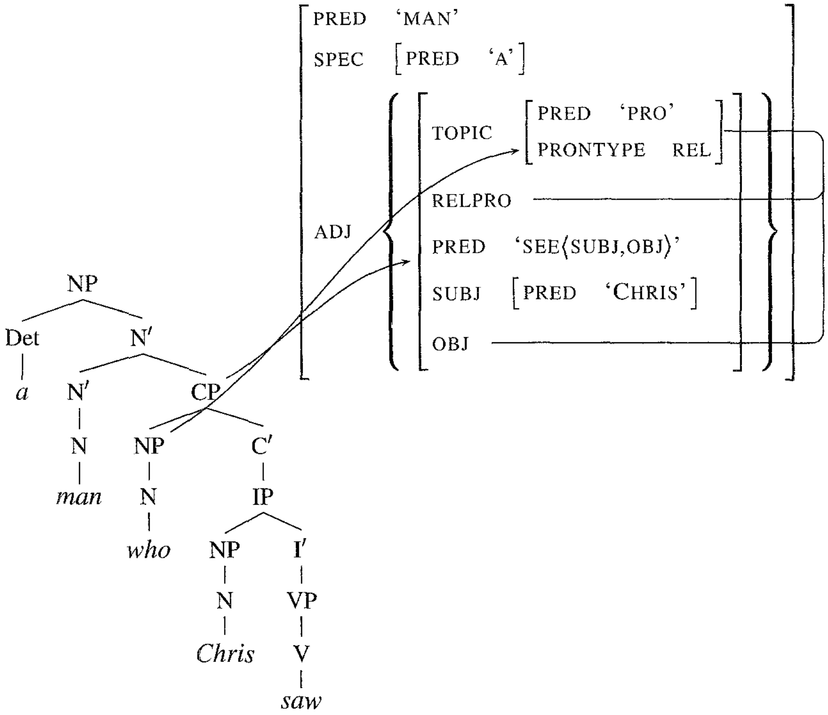
Relative clauses in English and many other languages also involve long-distance dependencies. Unlike the situation with topicalization, two long-distance dependencies are involved in a relative clause construction.

The first dependency holds between the fronted phrase and the within-clause grammatical function it fills. Bresnan and Mchombo (1987) propose that the fronted phrase in a relative clause bears the syntacticized TOPIC function. By the Extended Coherence Condition, given in (1) of this chapter, the TOPIC must be linked to a grammatical function within the clause.

The second dependency involves the relative pronoun and its position, possibly embedded, within the fronted phrase. We follow Kaplan and Bresnan (1982) in representing this syntactic dependency at f-structure; the f-structure of the relative pronoun appears as the value of the feature RELPRO within the relative clause. Similar representations have also been adopted by Butt et al. (1999) and Falk (2001).

The c-structure and f-structure for the phrase *a man who Chris saw* are given in (26):

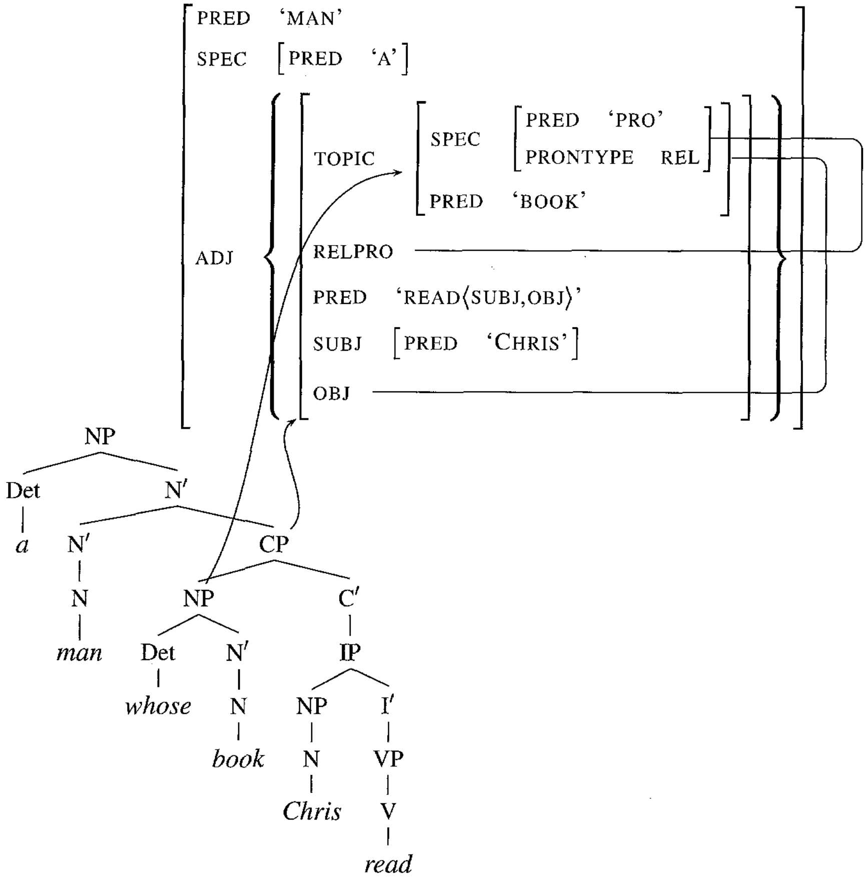
(26) *a man who Chris saw*



In (26), the relative pronoun appears in initial position in the relative clause, and its f-structure is both the TOPIC and the RELPRO of the relative clause.

Example (27) shows that the relative pronoun can also appear as a subconstituent of the initial phrase. Here the relative pronoun *whose* is a subconstituent of the fronted phrase *whose book*:

(27) *a man whose book Chris read*



In (27), the value of the TOPIC attribute is the f-structure of the fronted phrase *whose book*, and the value of the RELPRO attribute is the f-structure of the relative pronoun *whose*. We examine syntactic constraints on both of these dependencies in the following.

We propose the phrase structure rules in (28–29) for the analysis of these examples:

(28) $$\text{N}' \longrightarrow \begin{pmatrix} \text{N}' \\ \uparrow = \downarrow \end{pmatrix} \quad \begin{matrix} \text{CP}^* \\ \downarrow \in (\uparrow \text{ ADJ}) \end{matrix}$$

(29)
$$\text{CP} \longrightarrow \begin{pmatrix} \text{RelP} \\ (\uparrow \text{ TOPIC}) = \downarrow \\ (\uparrow \text{ TOPIC}) = (\uparrow \text{ RTOPICPATH}) \\ (\uparrow \text{ RELPRO}) = (\uparrow \text{ TOPIC RELPATH}) \\ (\uparrow \text{ RELPRO PRONTYPE}) =_c \text{REL} \end{pmatrix} \begin{pmatrix} \text{C}' \\ \uparrow = \downarrow \end{pmatrix}$$

The constituent structure metacategory RelP in (29) represents the phrase structure categories that can appear in initial position in a CP relative clause. The phrases in (30) exemplify the possible instantiations of RelP in English:[3]

(30) a. NP: *a man who I selected*

b. PP: *a man to whom I gave a book*

c. AP: *the kind of person proud of whom I could never be*

d. AdvP: *the city where I live*

Therefore, we define RelP for English in the rule in (29) as the following disjunction of categories:

(31) RelP ≡ {NP | PP | AP | AdvP}

The first two annotations on the RelP daughter in rule (29) are similar to the annotations on the TOPIC rule in (14) of this chapter. The constraint (↑ TOPIC) = ↓ requires the f-structure corresponding to the RelP node to fill the TOPIC role in the f-structure. The constraint (↑ TOPIC) = (↑ RTOPICPATH) ensures that the TOPIC f-structure also fills a grammatical function within the clause, constrained by the long-distance path RTOPICPATH; we define RTOPICPATH below.

The third and fourth annotations require the f-structure for the relative pronoun to appear as the value of the RELPRO attribute in the relative clause f-structure. The constraint in the third line, (↑ RELPRO) = (↑ TOPIC RELPATH), requires the value of the RELPRO attribute to appear at the end of the path RELPATH within the TOPIC f-structure. Below, we provide a definition of RELPATH that properly constrains the role of the relative pronoun within the fronted TOPIC phrase. Finally, the constraint (↑ RELPRO PRONTYPE) $=_c$ REL is a *constraining equation* (Chapter 5, Section 2.8) requiring the value of the RELPRO attribute to have a PRONTYPE feature with value REL: the value of the RELPRO attribute must be a relative pronoun.

We first discuss the definition of RTOPICPATH, the path relating the fronted constituent in a relative clause to its within-clause grammatical function. Constraints on RTOPICPATH are very similar to constraints on TOPICPATH, defined in (16) of this chapter:

(32) a. *Chris, we like.*

b. *a man who we like*

[3] Example (30c) is due to Webelhuth (1992).

(33) a. *Chris, we think that David saw.*

b. *a man who you think that David saw*

(34) a. **Chris, we whispered that David saw.*

b. **a man who you whispered that David saw*

(35) a. **Chris, [that David saw ___] surprised me.*

b. **a man who [that David saw ___] surprised me*

(36) a. *This hammer, we smashed the vase with.*

b. *the hammer which you smashed the vase with*

(37) a. **Chris, we think that David laughed when we selected.*

b. **a man who we think that David laughed when we selected*

We therefore propose the same constraints on the English RTopicPath as in (16) of this chapter, which constrains the long-distance path in topicalization constructions. The expressions in (16) and (38) are exactly the same:

(38) English RTopicPath:

$$\left\{\textsc{xcomp} \;\Big|\; \underset{(\rightarrow \textsc{ldd}) \neq -}{\textsc{comp}} \;\Big|\; \underset{(\rightarrow \textsc{tense})}{\textsc{obj}}\right\}^{*} \; \left\{\left(\textsc{adj} \; \underset{\neg(\rightarrow \textsc{tense})}{\in}\right) \; (\textsc{gf}) \;\Big|\; \textsc{gf}\right\}$$

Examination of other languages reveals different constraints on RTopicPath. As noted earlier, Kroeger (1993, Chapter 7) shows that RTopicPath in Tagalog is $\textsc{subj}^{+}$, paths consisting only of subj. Saiki (1985) discusses the definition of RTopicPath in Japanese, exploring constraints on RTopicPath in the causative and passive constructions.

Finally, we must define RelPath so as to appropriately constrain the grammatical function of the relative pronoun within the fronted topic f-structure. As originally noted by Ross (1967) and explored in detail by Bresnan (1976), Webelhuth (1992), Falk (2001), and many others, the relative pronoun may be embedded inside the fronted phrase. Ross (1967) provides this example of a deeply embedded relative pronoun:

(39) *[Reports [[the height of the lettering on the cover of which] the government prescribes ___]] should be abolished.*

Ross (1967) originally used the term *pied piping* in the transformational analysis of these constructions: in moving to the front of the sentence, the relative pronoun lures some additional material along with it, like the Pied Piper of Hamelin lured rats and children along with him as he left Hamelin.

Research on pied piping has revealed a range of constraints on the long-distance path RelPath to the relative pronoun in the fronted topic phrase:

(40) a. *the man [who] I met*

b. *the man [whose book] I read*

c. *the man [whose brother's book] I read*

d. *the report [the cover of which] I designed*

e. *the man [faster than whom] I can run*

f. *the kind of person [proud of whom] I could never be*

g. *the report [the height of the lettering on the cover of which] the government prescribes*

h. * *the man [a friend of whose brother] I met*

i. *the room [in which] I teach*

j. * *the man [the woman next to whom] I met*

In all of these examples, the phrase structure category of the fronted phrase is one of the categories defined by RelP, and no constraints on RTopicPath are violated. Example (40a) shows that the relative pronoun can itself appear in fronted position; in such a case, RelPath is the empty path. Examples (40b–c) indicate that the relative pronoun can appear as a possessor phrase, filling the spec role in the topic f-structure, or as the possessor of a possessor. It can also appear as the object of an oblique argument, as in (40d–f), or embedded inside an oblique argument, as in (40g), though it may not fill the spec role inside an oblique phrase (40h). It can appear as the object of a fronted adjunct phrase (40i), though it may not appear as an adjunct inside the fronted phrase (40f).

Given these facts, we propose the following definition of RelPath in English:

(41) English RelPath:

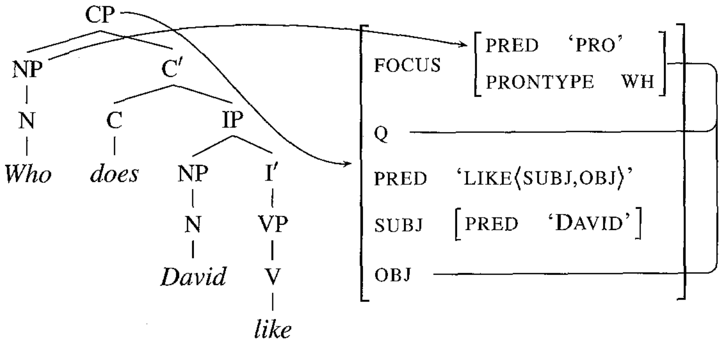
$$\{\text{SPEC}^* \mid [(\text{OBL}_\theta)\text{OBJ}]^*\}$$

In other languages the definition of RelPath differs. Webelhuth (1992) provides a thorough discussion of pied piping in Germanic, showing that constraints on pied piping in English relative clauses are different from the constraints that hold in German, Dutch, Swedish, and other Germanic languages.

1.3. Wh-Questions

In Chapter 4, Section 2.2.2, we noted that the question word in an English wh-question appears in initial position in the sentence, in the specifier position of CP:

(42) *Who does David like?*



To analyze constructions like (42), the following simplified rule was proposed in Chapter 6, Section 1.1:

(43) $$\text{CP} \longrightarrow \begin{pmatrix} \text{XP} \\ (\uparrow \textsc{focus}) = \downarrow \\ (\uparrow \textsc{focus}) = (\uparrow \textsc{comp}^{*} \ \textsc{gf}) \end{pmatrix} \begin{pmatrix} \text{C}' \\ \uparrow = \downarrow \end{pmatrix}$$

This rule ensures that the phrase in the specifier position of CP bears the FOCUS function and also fills a grammatical function within the utterance. We now refine this rule to take into account constraints on the phrase structure category of the fronted phrase and to give a more complete characterization of the path to its within-clause function. We also introduce the Q attribute, whose value is the f-structure of the possibly embedded interrogative pronoun within the fronted FOCUS phrase; see Kaplan and Bresnan (1982) for more discussion of this attribute.

We use the constituent structure metacategory QuesP and the functional abbreviations QFOCUSPATH and WHPATH in the following reformulation of rule (43):

(44) $$\text{CP} \longrightarrow \begin{pmatrix} \text{QuesP} \\ (\uparrow \textsc{focus}) = \downarrow \\ (\uparrow \textsc{focus}) = (\uparrow \text{Q}\textsc{FocusPath}) \\ (\uparrow \textsc{q}) = (\uparrow \textsc{focus} \ \text{W}\textsc{hPath}) \\ (\uparrow \textsc{q prontype}) =_{c} \textsc{wh} \end{pmatrix} \begin{pmatrix} \text{C}' \\ \uparrow = \downarrow \end{pmatrix}$$

The first issue is the correct definition of QuesP: which phrasal categories can appear as the FOCUS constituent in the specifier of CP? All of the examples in (45) are wellformed:

(45) a. NP: *Who do you like?*

b. PP: *To whom did you give a book?*

c. AdvP: *When did you yawn?*

d. AP: *How tall is Chris?*

Thus, we define QuesP in (44) above as the following disjunction of categories:

(46) QuesP ≡ {NP | PP | AdvP | AP}

The annotations on the QuesP node in rule (44) are similar to those on the relative clause rule in (29) of this chapter. The first two annotations require the f-structure corresponding to the QuesP node to fill the FOCUS role and also to bear some grammatical function defined by the long-distance path QFOCUSPATH; the correct definition of QFOCUSPATH will be our first topic of discussion in the following. The third annotation requires the value of the Q attribute to appear at the end of the long-distance path WHPATH within the FOCUS f-structure; we discuss constraints on WHPATH below. The fourth annotation requires the PRONTYPE attribute of the Q f-structure to bear the value WH, ensuring that an interrogative pronoun plays the Q role.

Our first task is to define QFOCUSPATH, the long-distance path involved in question formation. Constraints on QFOCUSPATH appear to be largely similar to those defined for TOPICPATH in (16) of this chapter (though see Postal 1998 for a discussion of differences between the two types of paths):

(47) a. *Chris, we like.*

b. *Who do you like?*

(48) a. *Chris, we think that David saw.*

b. *Who do you think that David saw?*

(49) a. **Chris, we whispered that David saw.*

b. **Who did you whisper that David saw?*

(50) a. **Chris, [that David saw ___] surprised me.*

b. **Who did [that David saw ___] surprise you?*

(51) a. *This hammer, we smashed the vase with.*

b. *What did you smash the vase with?*

(52) a. **Chris, we think that David laughed when we selected.*

b. **Who did you think that David laughed when we selected?*

Therefore, we provisionally provide the same definition for QFOCUSPATH as we gave for TOPICPATH in (16) of this chapter. Future research may reveal various additional refinements:

(53) English QFOCUSPATH:

$$\left\{ \text{XCOMP} \;\middle|\; \underset{(\rightarrow \text{LDD}) \neq -}{\text{COMP}} \;\middle|\; \underset{(\rightarrow \text{TENSE})}{\text{OBJ}} \right\}^{*} \left\{ \underset{\neg(\rightarrow \text{TENSE})}{(\text{ADJ} \quad \in \quad)} \;\; (\text{GF}) \;\middle|\; \text{GF} \right\}$$

In (41), we provided a constraint on RELPATH, the path to the relative pronoun within the fronted TOPIC phrase in a relative clause. Similarly, we must define WHPATH, the path to the interrogative pronoun in the fronted FOCUS phrase in a wh-question. Like the relative pronoun, the interrogative pronoun may be embedded inside the fronted phrase, appearing as a possessor or the possessor of a possessor and bearing the SPEC role, or as the OBJ of the fronted argument:

(54) a. *[Whose book] did you read?*
b. *[Whose brother's book] did you read?*
c. *[In which room] do you teach?*

However, WHPATH is more constrained than RELPATH, as the examples in (55–57) show:

(55) a. **[The cover of which report] did you design?*
b. (cf. *Which report did you design the cover of?*)
c. *the report [the cover of which] I designed*

(56) a. **[The height of the lettering on the cover of which report] does the government prescribe?*
b. *the report [the height of the lettering on the cover of which] the government prescribes*

(57) a. **[Faster than whom] can you run?*
b. *the man [faster than whom] I can run*

(58) a. **[Proud of whom] are you?*
b. *the kind of person [proud of whom] I could never be*

Therefore, we propose the following definition of WHPATH in English:

(59) English WHPATH:

$\{\text{SPEC}^* \mid \text{OBJ}\}$

Webelhuth (1992) provides more discussion of constraints on pied piping in Germanic languages, showing that pied piping constraints in English wh-questions are the same as in other Germanic languages.

2. MORPHOLOGICAL SIGNALING

Some languages signal long-distance dependency constructions by means of special morphological or phonological forms, as noted by Clements and Ford

(1979) for Kikuyu, McCloskey (1979) for Irish, Chung (1982) for Chamorro, and Georgopoulos (1985) for Palauan. These constructions were first analyzed in LFG by Zaenen (1983) in work on Kikuyu, based on the earlier LFG analysis of long-distance dependencies in terms of c-structural relations (Kaplan and Bresnan 1982). Here we review the data that Zaenen (1983) originally treated, presenting a new analysis that focuses on f-structure rather than c-structure relations. The analysis in this section is based on unpublished joint work by Ron Kaplan, John Maxwell, Annie Zaenen, and the author.

As discussed by Zaenen (1983) and in more detail by Clements and Ford (1979), in affirmative declarative Kikuyu sentences the verb is associated with a downstep tone.[4] This downstep tone affects the phonology of the words following the verb: the downstep shifts over the following phrasal category and turns the immediately following sequence of low tones into high tones. In example (60), a downstep is associated with the two verbs *ɛ́:$^!$rírɛ́* 'tell' and *átɛ́mírɛ́* 'cut':

(60) *Kamaú ɛ́:$^!$rírɛ́ Ka:nákɛ́ áté Kárìók$^!$í átɛ́mírɛ́ mótě*
Kamau SUBJ.tell.PAST Kanake that Kariũki SUBJ.cut.PAST tree
'Kamau told Kanake that Kariũki cut the tree.'

The downstep associated with *ɛ́:$^!$rírɛ́* 'tell' affects the words *áté* 'that' and *Kárìók$^!$í* 'Kariũki', whose citation forms are given in (61):

(61) a. *ate*
b. *Kariokĭ*

As Zaenen (1983) shows, the explanation for these difference can be ascribed to the tonal shift imposed by the verb *ɛ́:$^!$rírɛ́* 'tell'.

Interestingly, however, this tone change does not appear within the domain of a long-distance dependency:

(62) *nóo Kámaú ɛ́:$^!$rírɛ́ Ka:nákɛ́ áte otɛmírɛ́ mote*
who Kamau SUBJ.tell.PAST Kanake that SUBJ.cut.PAST tree
'Who did Kamau tell Kanake that ___ cut the tree?'

The question word *nóo* 'who' bears the FOCUS function and also fills the role of the SUBJ of the subordinate COMP. The downstep that would be expected to appear on both the matrix and subordinate clause verb does not appear, and the effect that this downstep would have had on the following words is not present.[5] As Clements and Ford (1979) and Zaenen (1983) show, the absence of the downstep

[4] In the following examples, high tones are marked with an acute accent, downstep tones are marked with !, and low tones are unmarked.

[5] The high tone on the initial syllable of*ate* 'that' is spread from the final high tone of the preceding word *Kárìók$^!$í* 'Kariũki' by an independent rule.

tone marks the *domain of extraction* of the fronted interrogative pronoun. Here, the domain of extraction is the f-structure for the entire sentence.

We propose that every f-structure in the domain of extraction in Kikuyu is marked with the attribute-value pair ⟨LDD, +⟩:

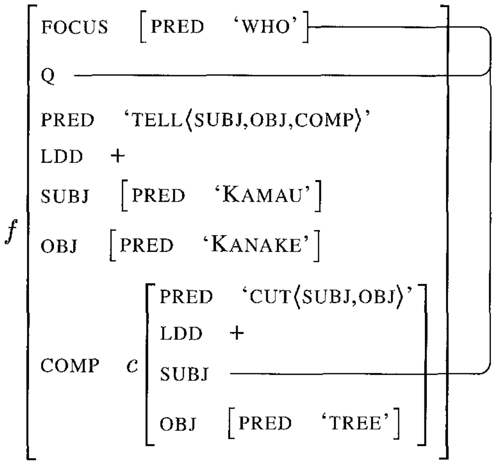
(63) Each attribute on the path relating the FOCUS to its within-clause grammatical function must be an attribute of an f-structure that also contains the pair ⟨LDD, +⟩.

In (64) this constraint is satisfied, since the f-structures labeled *f* and *c* both contain the attribute-value pair ⟨LDD, +⟩:

(64)

nóo	*Kámaú*	*ɛ́:ꜜrírɛ́*	*Ka:nákɛ́*	*áte*	*otɛmírɛ́*	*mote*
who	Kamau	SUBJ.tell.PAST	Kanake	that	SUBJ.cut.PAST	tree

'Who did Kamau tell Kanake that ___ cut the tree?'



Within the domain marked by the attribute-value pair ⟨LDD, +⟩, spreading of the downstep tone does not occur.

To constrain the path defining a long-distance dependency in Kikuyu, ensuring that the path passes only through f-structures that are appropriately marked with the LDD feature, we use *off-path constraints*, defined and discussed in Chapter 6, Section 1.4 and in Section 1.1.2 of this chapter. We assume a simplified definition of QFOCUSPATH for Kikuyu, representing it as COMP* GF:[6]

(65) $(\uparrow \text{FOCUS}) = (\uparrow \underset{(\leftarrow\text{LDD}) = +}{\text{COMP}^*} \underset{(\leftarrow\text{LDD}) =+}{\text{GF}})$

[6] It is likely that grammatical functions other than COMP are allowed in the extraction domain in Kikuyu. In a full treatment of Kikuyu long-distance dependencies, the equation in (65) should be enriched appropriately to properly characterize the full range of permitted relations between the displaced question word and its grammatical function within the clause.

In (65), the expression $\underset{(\leftarrow\text{LDD}) = +}{\text{COMP}^*}$ refers to zero or more occurrences of the constrained attribute $\underset{(\leftarrow\text{LDD}) = +}{\text{COMP}}$. The metavariable $\leftarrow$ in this off-path constraint refers to the f-structure that contains the attribute on which the constraint is written. Thus, the off-path constraint $(\leftarrow\text{LDD}) = +$ ensures that each f-structure that has a COMP attribute on the path also has the attribute LDD with value $+$. The equation in (65) associates the constraint $(\leftarrow\text{LDD}) = +$ with every attribute on the path, and marks the entire Kikuyu domain of extraction with the LDD feature, as desired.

3. LONG-DISTANCE DEPENDENCIES AT C-STRUCTURE

The theory of the constituent structure properties of long-distance dependencies in LFG has undergone a major revision since the inception of the theory. Kaplan and Bresnan (1982) originally proposed to treat long-distance dependencies in terms of a relation between two positions in the constituent structure tree, one corresponding to the displaced constituent and the other to a *gap* or *trace* in the within-clause position associated with the syntactic role of the displaced constituent. On this view, the f-structure role of a displaced constituent in a long-distance dependency is determined by the c-structure position of its corresponding gap. This theory was further developed in work by Zaenen (1980, 1983).

Subsequently, however, work by Kaplan et al. (1987) and Kaplan and Zaenen (1989) showed that constraints on long-distance dependencies are in fact primarily functional in nature and are best characterized in f-structure, not c-structure, terms. Kaplan et al. (1987) and Kaplan and Zaenen (1989) first proposed functional uncertainty as a way of stating constraints on long-distance dependencies, as we have done in the foregoing discussion. Under this approach, the role of constituent structure in constraining long-distance dependencies is considerably diminished, and it is reasonable to reevaluate the role of constituent structure and to reexamine evidence for gaps or traces. Indeed, Kaplan and Zaenen (1989) present a set of arguments against the existence of traces (see also Sag and Fodor 1994), showing that many arguments that have been made in support of traces are flawed.

However, these works do not address one argument for the presence of traces: the phenomenon of *weak crossover*, originally discussed by Wasow (1979) and illustrated in (66):

(66) * *Who$_i$ does his$_i$ mother like ___?*

The name *crossover* comes from early transformational analyses of examples like (66), which assumed that a violation ensues when a wh-phrase like *who* "crosses over" a coindexed pronoun in moving to the front of the sentence: in (66), *who* crosses over the coindexed pronoun *his*. Bresnan (1995, 1998, 2001b) proposes a theory of prominence for anaphoric binding that accounts for examples like (66) by assuming the presence of a trace in the object position of *like*.

As Bresnan demonstrates, languages vary in imposing different types of prominence requirements on the binding relation between a wh-phrase and the pronouns it binds. Three prominence dimensions are relevant: the functional hierarchy (on which, for example, SUBJ outranks OBJ); the thematic hierarchy (on which AGENT outranks THEME); and linear order (f-preceding elements outrank the elements they f-precede). Bresnan (2001b) proposes that in English, a wh-phrase must outrank any pronouns it binds on *either* the functional hierarchy *or* the linear precedence hierarchy, so that satisfaction of either type of prominence requirement is sufficient. In example (66), the wh-phrase *who* fills the OBJ role, and the pronoun *his* is contained in the SUBJ. Thus, the pronoun outranks the wh-phrase, and the functional prominence condition is not met.

In examining whether the linear prominence condition holds, we must determine whether the wh-phrase *who* f-precedes the pronoun *his*. Crucially for her analysis, Bresnan assumes that wh-question formation in English involves the presence of traces, so that the f-structure for *who* corresponds to two c-structure positions: the fronted position in which *who* appears and the gap position following *like*. Bresnan (2001b) provides the definition of f-precedence given in (107) of Chapter 6, Section 4.4, repeated in (67):

(67) F-precedence, alternate definition (Bresnan 2001b):

f f-precedes g if the rightmost node in $\phi^{-1}(f)$ precedes the rightmost node in $\phi^{-1}(g)$.

In (66), the rightmost node corresponding to the f-structure for *who* is the position of the trace, which does not precede the pronoun; therefore, the wh-phrase does not f-precede the pronoun according to the definition given in (67), and the linear prominence requirement is not satisfied. Since neither the functional prominence condition nor the linear prominence condition is met, example (66) is correctly predicted to be ungrammatical.

Bresnan (1995, 1998, 2001b) and Berman (2000) examine weak crossover violations in German, proposing that German imposes the same prominence conditions on binding by wh-phrases as English, and ascribing differences between the two languages to variations in the presence of traces in the two languages. The German equivalent of example (66) is fully grammatical:

(68)	*Wen*	*mag*	*seine*	*Mutter?*
	who-ACC	likes	his	mother-NOM

'Who$_i$ does his$_i$ mother like?'

Bresnan (2001b) and Berman (2000) argue that no trace is present in German examples like (68), since the grammatical function of the fronted argument is determined locally, by reference to its casemarking. Since no trace appears in this clause, the f-structure of the wh-phrase f-precedes the f-structure of the pronoun; the linear prominence condition is thereby satisfied, accounting for the acceptability of the example.

In contrast, they argue that traces must be assumed in extractions from subordinate clauses in German, since case information is insufficient to identify the grammatical function of the extracted element in the embedded clause:

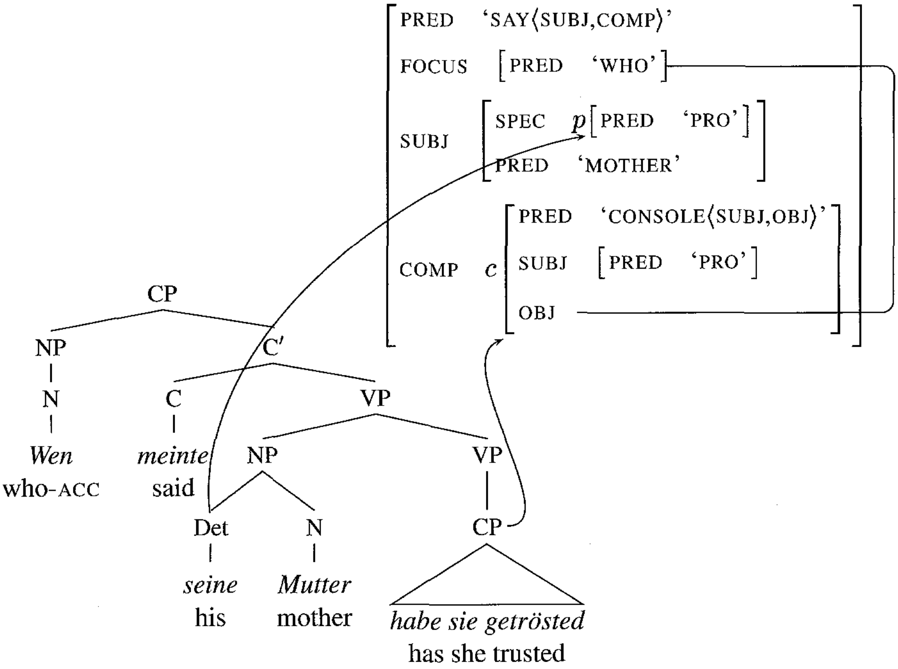
(69) * *Wen meinte seine Mutter, habe sie getrösted?*
who-ACC said his mother has she consoled
'Who$_i$ did his$_i$ mother say that she consoled?'

Here, the linear prominence condition is not satisfied. The f-structure for the wh-phrase corresponds to the position of the trace in the subordinate clause as well as the position of *wen* 'who'. The subordinate clause trace position is rightmost, and is therefore the one that is relevant in evaluating f-precedence requirements. This position does not precede the matrix clause pronoun, and the linear prominence condition is not met. Since the functional prominence condition is also not met in this example, it is correctly predicted to be ungrammatical. Thus, according to Bresnan's proposal, the availability and distribution of empty c-structure categories provide a new view of weak crossover phenomena in German, Hindi, Malayalam, and Palauan.

However, there are indications that an account of weak crossover may be available even if traces are not assumed. In unpublished joint work by Ron Kaplan, Tracy Holloway King, and the author, a different definition of the linear prominence condition is explored, based on unpublished work by Sag (1998). In this approach, the f-precedence relation that is considered in evaluating the linear prominence condition holds between the pronoun and an f-structure that *f-commands*[7] the pronoun and contains the wh-phrase. In (70), the f-structure that f-commands the pronoun and contains the f-structure for the wh-phrase is the COMP f-structure, labeled *c*:

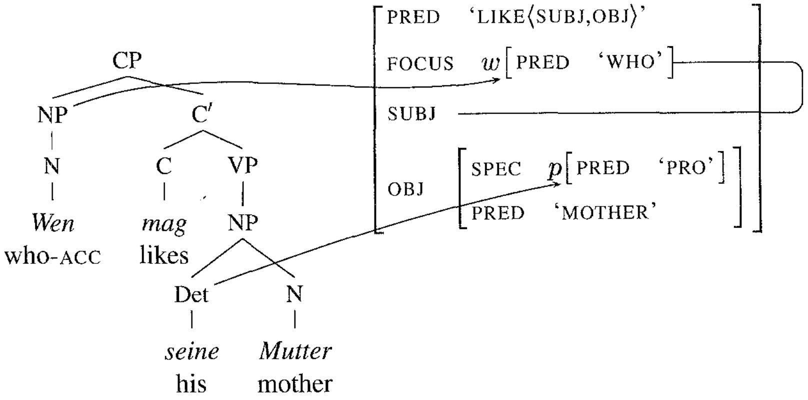
[7] F-command is defined in Chapter 6, Section 3.1. Intuitively, an f-structure f-commands its "sister" f-structures, those that are arguments of the same f-structure, and f-structures contained in its sisters.

(70) * *Wen meinte seine Mutter, [habe sie getrösted]?*
who-ACC said his mother has she consoled
'Who$_i$ did his$_i$ mother say that she consoled?'



According to either definition of f-precedence given in Chapter 6, Section 4.4, the f-structure labeled *c* does not f-precede the pronoun f-structure *p*, and the example is ruled out. In contrast, in example (71) the f-structure that f-commands the pronoun and contains the operator is the f-structure for *wen* 'who', labeled *w*:

(71) *Wen* *mag* *seine* *Mutter?*
who-ACC likes his mother-NOM
'Who$_i$ does his$_i$ mother like?'



The f-structure labeled w f-precedes the pronoun f-structure p, and the linear prominence condition is satisfied. The ungrammaticality of the corresponding English example in (66) is accounted for by the syntactic rank condition, as described above; thus, this alternative account of weak crossover violations differs from accounts presented by Bresnan (2001b) and Berman (2000) in assuming that in English, though not in German, the functional prominence condition must always be met.

On this approach, weak crossover violations are accounted for without assuming traces. Future work will reveal more about how binding by wh-phrases and other operators is constrained, and whether incontrovertible evidence exists for traces, gaps, or empty phrase structure categories. In the absence of such evidence, a simpler and more parsimonious theory of long-distance dependencies results if traces are not allowed.

4. RELATIVE CLAUSES AND SEMANTIC COMPOSITION

4.1. Semantics of Relative Clauses

Like the adjectival modifiers studied in Chapter 10, a relative clause is a noun modifier, producing a modified meaning of type $\langle e \multimap t \rangle$ when combined with a noun meaning of the same type. Recall that a simple noun like *man* has a meaning like the following, which we can think of as picking out the set of individuals that are men:

(72) *man*

$\lambda X.\mathit{man}(X)$

The meaning of a noun modified by a relative clause, like *man who Chris saw*, is of the same type. It represents the set of individuals that are men (the meaning contribution of *man*), that are people (the meaning contribution of *who*), and that were seen by Chris (the meaning contribution of *Chris saw*):

(73) *man who Chris saw*

$\lambda X.\mathit{person}(X) \wedge \mathit{see}(\mathit{Chris}, X) \wedge \mathit{man}(X)$

The meaning contribution of the relative pronoun *who* is redundant here, since the fact that an individual is a man entails that he is a person. In some cases, however, the information contributed by the relative pronoun is not redundant: consider an example like *pitcher who/which Chris saw*.

Relative clauses with possessive relative pronouns are interpreted similarly.[8] The phrase *man whose book Chris read* is interpreted as in (74) and refers to individuals who are men and who possess the book that Chris read:

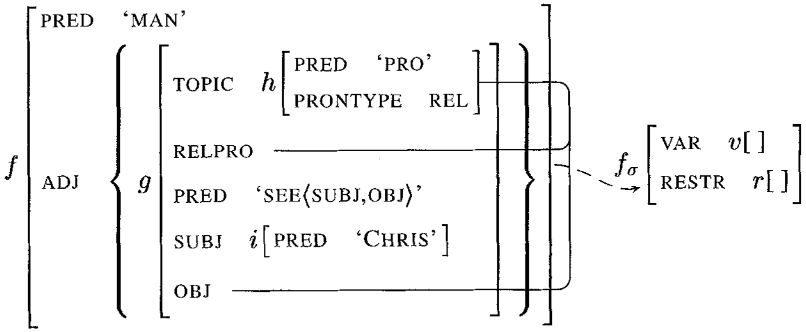
(74) *man whose book Chris read*

$\lambda X.\mathit{the}(Y, \mathit{poss}(X, Y) \wedge \mathit{book}(Y), \mathit{read}(\mathit{Chris}, Y)) \wedge \mathit{man}(X)$

4.2. Relative Clauses and Meaning Assembly

We assume the f-structure, semantic structure, and meaning constructor in (75) for the phrase *man who Chris saw*:

(75) *man who Chris saw*



$\lambda X.\mathit{person}(X) \wedge \mathit{see}(\mathit{Chris}, X) \wedge \mathit{man}(X) : v \multimap r$

[8] We treat the possessive determiner as definite, following Partee and Borschev (1998).

As described in Chapter 10, noun modifiers such as the relative clause *who Chris saw* combine with the noun meanings they modify to produce a new, modified meaning of the same type. Thus, the implicational meaning constructor in (75) is similar to the meaning constructor for a proper noun like *man*, but with a modified meaning reflecting the meaning of the relative clause:

(76) *man*

$$f\left[\text{PRED} \quad \text{`MAN'}\right] \dashrightarrow f_\sigma \begin{bmatrix} \text{VAR} & v[\,] \\ \text{RESTR} & r[\,] \end{bmatrix}$$

$$\lambda X.\mathit{man}(X) : v \multimap r$$

In the analysis of example (75), we assume the relative clause rule given in (29) of this chapter, augmented with the meaning constructor labeled **[rel]**:

$$\text{(77)}\quad \text{CP} \longrightarrow \left(\begin{array}{c} \text{RelP} \\ (\uparrow \text{TOPIC}) = \downarrow \\ (\uparrow \text{TOPIC}) = (\uparrow \text{RTOPICPATH}) \\ (\uparrow \text{TOPIC RELPATH}) = (\uparrow \text{RELPRO}) \\ (\uparrow \text{RELPRO PRONTYPE}) =_c \text{REL} \end{array}\right) \left(\begin{array}{c} \text{C}' \\ \uparrow = \downarrow \\ \textbf{[rel]} \end{array}\right)$$

As discussed in Chapter 9, Section 6, there are cases in which meaning contributions are associated with phrase structure configurations rather than lexical items. Relative clause formation in English is one such case: the meaning constructor **[rel]**, which combines the relative clause meaning with the meaning of the modified noun, is associated with the relative clause CP rule. The definition of **[rel]** is given in (78):

$$\begin{aligned} \text{(78)}\quad \textbf{[rel]}\quad & \lambda P.\lambda Q.\lambda X.P(X) \wedge Q(X) : \\ & [(\uparrow \text{RELPRO})_\sigma \multimap \uparrow_\sigma] \multimap \\ & \quad [[((\text{ADJ} \in \uparrow)_\sigma \text{ VAR}) \multimap ((\text{ADJ} \in \uparrow)_\sigma \text{ RESTR})] \multimap \\ & \quad\;\; [((\text{ADJ} \in \uparrow)_\sigma \text{ VAR}) \multimap ((\text{ADJ} \in \uparrow)_\sigma \text{ RESTR})]] \end{aligned}$$

Instantiating the $\uparrow$ metavariables in (78) according to the f-structure labels in (81), we have the meaning constructor premise in (79):

$$\text{(79)}\quad \textbf{[rel]}\quad \lambda P.\lambda Q.\lambda X.P(X) \wedge Q(X) : [h_\sigma \multimap g_\sigma] \multimap [[v \multimap r] \multimap [v \multimap r]]$$

This meaning constructor consumes the meaning resource of the relative clause $h_\sigma \multimap g_\sigma$ and the meaning resource of the modified noun $v \multimap r$ to produce a new noun resource, also associated with $v \multimap r$ but reflecting a modified meaning.

We must also provide a meaning for the relative pronoun *who*. We propose that *who* augments the relative clause meaning by providing the additional meaning that the entity involved is a person. The lexical entry for *who* is given in (80):

(80) *who* $(\uparrow \textsc{pred}) = \textsc{`pro'}$
$(\uparrow \textsc{prontype}) = \textsc{rel}$
$\lambda S.\lambda X.\mathit{person}(X) \wedge S(X) :$
$[\uparrow_\sigma \multimap (\textsc{relpro}\uparrow)_\sigma] \multimap [\uparrow_\sigma \multimap (\textsc{relpro}\uparrow)_\sigma]$

Instantiating the $\uparrow$ metavariables in the meaning constructor in (81), we have the meaning constructor labeled **[who]** in (80). We also assume the standard meaning constructor for the name *Chris* as given in Chapter 9. Since the example we discuss does not involve anaphoric binding, we provide simple noncontextual meaning constructors rather than the context-meaning pair constructors discussed in Chapter 11. Thus, the meaning constructor premises in (81) are relevant in the analysis of *man who Chris saw*:

(81) Meaning constructor premises for *man who Chris saw*:

[man]	$\lambda X.\mathit{man}(X)$	:	$v \multimap r$
[who]	$\lambda S.\lambda X.\mathit{person}(X) \wedge S(X)$	:	$[h_\sigma \multimap g_\sigma] \multimap [h_\sigma \multimap g_\sigma]$
[see]	$\lambda X.\lambda Y.\mathit{see}(X,Y)$	:	$i_\sigma \multimap [h_\sigma \multimap g_\sigma]$
[Chris]	Chris	:	i_σ
[rel]	$\lambda P.\lambda Q.\lambda X.P(X) \wedge Q(X)$	:	$[h_\sigma \multimap g_\sigma] \multimap [[v \multimap r] \multimap [v \multimap r]]$

We begin by combining the premises labeled **[Chris]** and **[see]** to obtain the meaning constructor labeled **[Chris-see]** in (82):

(82) **[Chris-see]** $\lambda Y.\mathit{see}(\mathit{Chris},Y) : h_\sigma \multimap g_\sigma$

This provides the meaning resource required by the premise **[who]**. Combining **[Chris-see]** and **[who]**, we have the meaning constructor labeled **[who-Chris-see]** in (83):

(83) **[who-Chris-see]** $\lambda X.\mathit{person}(X) \wedge \mathit{see}(\mathit{Chris},X) : h_\sigma \multimap g_\sigma$

Next, we combine the premises **[who-Chris-see]** and **[rel]**, producing the premise **[who-Chris-see-rel]** in (84):

(84) **[who-Chris-see-rel]** $\lambda Q.\lambda X.\mathit{person}(X) \wedge \mathit{see}(\mathit{Chris},X) \wedge Q(X) :$
$[v \multimap r] \multimap [v \multimap r]$

As discussed in Chapter 10, this meaning constructor has the characteristic form of a modifier: it consumes a resource and produces a new resource that is the same as the modified meaning constructor on the right-hand side but is associated with a modified meaning. Combining **[who-Chris-see-rel]** with **[man]**, we have, as desired:

(85) **[man], [who-Chris-see-rel]** $\vdash$

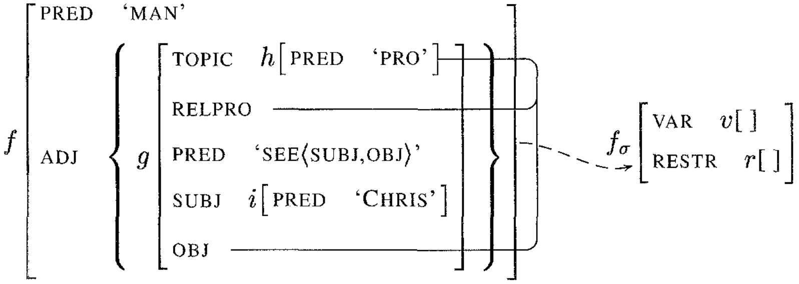
$$\lambda X.\mathit{person}(X) \wedge \mathit{see}(\mathit{Chris}, X) \wedge \mathit{man}(X) : v \multimap r$$

We now expand our treatment of English relative clauses to encompass relative clauses with no relative pronoun, as in an example like *the man Chris saw*. An advantage of our analysis is that no additions or changes need be made to the lexical entries or meaning constructors provided so far. All that is necessary is to augment the c-structure rule given in (77) to provide the proper syntactic constraints when a relative pronoun is not present, using the ϵ notation introduced in Chapter 6, Section 4.5.

(86)
$$\text{CP} \longrightarrow \left\{ \begin{array}{c|c} \text{RelP} & \epsilon \\ (\uparrow \text{TOPIC}) = \downarrow & (\uparrow \text{TOPIC PRED}) = \text{`PRO'} \\ (\uparrow \text{TOPIC}) = (\uparrow \text{RTOPICPATH}) & (\uparrow \text{TOPIC}) = (\uparrow \text{RTOPICPATH}) \\ (\uparrow \text{TOPIC RELPATH}) = (\uparrow \text{RELPRO}) & (\uparrow \text{TOPIC}) = (\uparrow \text{RELPRO}) \\ (\uparrow \text{RELPRO PRONTYPE}) =_c \text{REL} & \end{array} \right\} \left(\begin{array}{c} \text{C}' \\ \uparrow = \downarrow \\ \textbf{[rel]} \end{array} \right)$$

According to this rule, when no RelP phrase is present, the equations under ϵ must be satisfied: the rule provides a TOPIC attribute whose value for the attribute PRED is 'PRO' and equates the value of the TOPIC with the value of the RELPRO attribute. With these assumptions, the phrase *man Chris saw* has the f-structure, semantic structure, and meaning constructor given in (87):

(87) *man Chris saw*



$$\lambda X.\mathit{see}(\mathit{Chris}, X) \wedge \mathit{man}(X) : v \multimap r$$

The meaning derivation proceeds straightforwardly from the premises in (88), which are contributed by the lexical items and the CP phrase structure rule figuring in the analysis of this phrase:

(88) Meaning constructor premises for *man Chris saw*:

[man]	$\lambda X.\mathit{man}(X)$	$: v \multimap r$
[see]	$\lambda X.\lambda Y.\mathit{see}(X,Y)$	$: i_\sigma \multimap [h_\sigma \multimap g_\sigma]$
[Chris]	Chris	$: i_\sigma$
[rel]	$\lambda P.\lambda Q.\lambda X.P(X) \wedge Q(X)$	$: [h_\sigma \multimap g_\sigma] \multimap [[v \multimap r] \multimap [v \multimap r]]$

As above, we combine the premises labeled **[Chris]** and **[see]**, obtaining the premise labeled **[Chris-see]** in (89):

(89) **[Chris-see]** $\lambda Y.\mathit{see}(\mathit{Chris},Y) : h_\sigma \multimap g_\sigma$

We combine **[Chris-see]** with **[rel]** to obtain **[Chris-see-rel]**, given in (90):

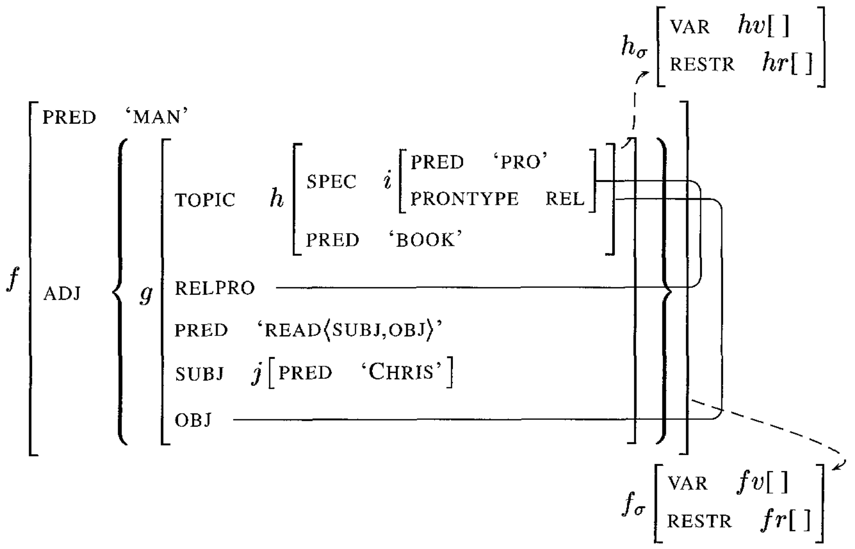
(90) **[Chris-see-rel]** $\lambda Q.\lambda X.\mathit{see}(\mathit{Chris},X) \wedge Q(X) : [v \multimap r] \multimap [v \multimap r]$

Combining **[Chris-see-rel]** with **[man]**, we obtain the desired result:

(91) **[man], [Chris-see-rel]** $\vdash \lambda X.\mathit{see}(\mathit{Chris},X) \wedge \mathit{man}(X) : v \multimap r$

Our final task is to examine relative clauses with possessive relative pronouns. The f-structure, semantic structures, and meaning constructor for *man whose book Chris read* are given in (92):

(92) *man whose book Chris read*



$\lambda X.\mathit{the}(Y, \mathit{poss}(X,Y) \wedge \mathit{book}(Y), \mathit{read}(\mathit{Chris},Y)) \wedge \mathit{man}(X) : fv \multimap fr$

Using the predicate name *poss* to represent the generalized "possession" relation holding between the possessor and the possessed entity, we assume the lexical entry in (93) for the possessive relative pronoun *whose*:

(93) *whose* $(\uparrow \text{PRED}) = \text{`PRO'}$
$(\uparrow \text{PRONTYPE}) = \text{REL}$

$$\lambda P.\lambda Q.\lambda X.\mathit{the}(Y, \mathit{poss}(X,Y) \wedge P(Y), Q(Y)) :$$
$$[((\text{SPEC}\uparrow)_\sigma\ \text{VAR}) \multimap ((\text{SPEC}\uparrow)_\sigma\ \text{RESTR})] \multimap$$
$$[[(\text{SPEC}\uparrow)_\sigma \multimap (\text{TOPIC SPEC}\uparrow)_\sigma] \multimap [\uparrow_\sigma \multimap (\text{TOPIC SPEC}\uparrow)_\sigma]]$$

In (93) we instantiate this meaning constructor according to the f-structure labels in (92), obtaining the meaning constructor labeled **[whose]** in (94). We also provide the standard meaning constructor for the common noun *book*, labeled **[book]**:

(94) Meaning constructor premises for *whose book*:

[book] $\lambda X.\mathit{book}(X) : hv \multimap hr$

[whose] $\lambda P.\lambda Q.\lambda X.\mathit{the}(Y, \mathit{poss}(X,Y) \wedge P(Y), Q(Y)) :$
$[hv \multimap hr] \multimap [[h_\sigma \multimap g_\sigma] \multimap [i_\sigma \multimap g_\sigma]]$

The meaning constructor of *whose* is similar to the meaning constructor for *every*, discussed in Chapter 9: *every* requires a meaning for its restriction and a meaning for its scope to produce a meaning for the sentence in which it appears. The possessive determiner *whose* also requires a meaning for its restriction; this requirement is represented on the right-hand side of the meaning constructor labeled **[whose]** by a requirement for a resource $hv \multimap hr$, corresponding to the meaning P, exactly as for a determiner like *every*. This requirement is satisfied by the meaning constructor **[book]**. Combining **[whose]** and **[book]**, we have the meaning constructor **[wh-book]** in (95):

(95) **[wh-book]** $\lambda Q.\lambda X.\mathit{the}(Y, \mathit{poss}(X,Y) \wedge \mathit{book}(Y), Q(Y)) :$
$[h_\sigma \multimap g_\sigma] \multimap [i_\sigma \multimap g_\sigma]$

Besides a meaning for its restriction, a determiner like *every* requires a meaning resource for its scope. When an appropriate scope meaning resource is found, it is consumed and a new meaning resource for the scope is provided, incorporating the semantics of the quantifier. Analogously, the possessive relative determiner *whose* requires a meaning resource $h_\sigma \multimap g_\sigma$, corresponding to its scope. When such a meaning is found, a meaning resource $i_\sigma \multimap g_\sigma$ for the relative clause is provided.

In the derivation of the meaning of *man whose book Chris read*, the meaning constructors in (96) are relevant:

(96) Meaning constructors for *man whose book Chris read*:

[man] $\lambda X.\mathit{man}(X) \ : \ f v \multimap f r$

[wh-book] $\lambda Q.\lambda X.\mathit{the}(Y, \mathit{poss}(X,Y) \wedge \mathit{book}(Y), Q(Y)) \ :$
$[h_\sigma \multimap g_\sigma] \multimap [i_\sigma \multimap g_\sigma]$

[read] $\lambda X.\lambda Y.\mathit{read}(X,Y) \ : \ j_\sigma \multimap [h_\sigma \multimap g_\sigma]$

[Chris] $\mathit{Chris} \ : \ j_\sigma$

[rel] $\lambda P.\lambda Q.\lambda X.P(X) \wedge Q(X) \ :$
$[i_\sigma \multimap g_\sigma] \multimap [[f v \multimap f r] \multimap [f v \multimap f r]]$

We begin by combining **[Chris]** and **[read]** to produce **[Chris-read]**:

(97) **[Chris-read]** $\lambda Y.\mathit{see}(\mathit{Chris}, Y) : h_\sigma \multimap g_\sigma$

This meaning constructor provides the scope resource required by **[wh-book]**. Combining **[Chris-read]** and **[wh-book]**, we have the meaning constructor labeled **[wh-book-Chris-read]** in (98):

(98) **[wh-book-Chris-read]**
$\lambda X.\mathit{the}(Y, \mathit{poss}(X,Y) \wedge \mathit{book}(Y), \mathit{see}(\mathit{Chris}, Y)) : i_\sigma \multimap g_\sigma$

We can now combine **[rel]** with **[wh-book-Chris-read]**, obtaining the meaning constructor labeled **[wh-book-Chris-read-rel]**:

(99) **[wh-book-Chris-read-rel]**
$\lambda Q.\lambda X.\mathit{the}(Y, \mathit{poss}(X,Y) \wedge \mathit{book}(Y), \mathit{see}(\mathit{Chris}, Y)) \wedge Q(X) :$
$[f v \multimap f r] \multimap [f v \multimap f r]$

Finally, we combine **[wh-book-Chris-read-rel]** with **[man]**, producing the desired meaning constructor for this example:

(100) **[man], [wh-book-Chris-read-rel]** $\vdash$

$\lambda X.\mathit{the}(Y, \mathit{poss}(X,Y) \wedge \mathit{book}(Y), \mathit{read}(\mathit{Chris}, Y)) \wedge \mathit{man}(X) : f v \multimap f r$

5. WH-QUESTIONS AND SEMANTIC COMPOSITION

Most current work on the semantics of questions has its roots in the work of Hamblin (1958, 1976), who shows that there is an intimate relationship between the meaning of a question and the meanings of its possible complete answers. Hamblin (1958) outlines the following postulates for the interpretation of questions:

(101) (i) An answer to a question is a statement.

(ii) Knowing what counts as an answer is equivalent to knowing the question.

(iii) The possible answers to a question are an exhaustive set of mutually exclusive possibilities.

Subsequent work by Karttunen (1977), Groenendijk and Stokhof (1984), Ginzburg (2001), and many others has expanded and refined our view of the semantics of questions. Useful overviews are presented by Higginbotham (1995), Ginzburg (1996), and Groenendijk and Stokhof (1997).

Many complications arise in the semantic analysis of questions, and a complete treatment of question semantics in a glue setting must await future research. In the following, we discuss some of the issues that must be addressed.

The first issue is to determine an appropriate representation of question meaning. The meaning of a question is inherently intensional, and so intensional logic provides a more appropriate way of representing question meanings than the predicate logic representations that we have assumed in previous chapters. Here as always, however, our primary focus is on semantic composition, not the details of semantic interpretation. In fact, we believe that the same issues in semantic composition arise in considering an appropriately simplified question meaning as would arise in a more complete treatment, so that a simple predicate logic meaning representation is sufficient for our present discussion. Following Ginzburg (1996), we provide the meaning in (102) for the question *Who does David like?*:

(102) *Who does David like?*

$$\lambda P.[\exists X.(\mathit{person}(X) \ \wedge P = \mathit{like}(\mathit{David}, X))]$$

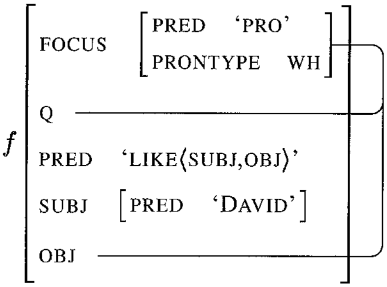
In this expression, P represents members of the set of propositions that constitute answers to the question of who David likes: in other words, this expression picks out the set of propositions of the form $\mathit{like}(\mathit{David}, X)$ for each person X whom David likes:

(103) $$\lambda P.[\exists X.(\mathit{person}(X) \ \wedge P = \mathit{like}(\mathit{David}, X))] = \\ \{\mathit{like}(\mathit{David}, \mathit{Ken}), \mathit{like}(\mathit{David}, \mathit{Mary}), \mathit{like}(\mathit{David}, \mathit{Matty}) \ldots\}$$

This simple treatment is compatible with the Hamblin postulates, since it identifies the meaning of the question with the set of propositions that constitute its complete answer.

Next, we examine the issue of semantic composition and identification of the meaning constructors that are involved in the derivation of a question meaning. The sentence *Who does David like?* has the f-structure shown in (104):

(104) *Who does David like?*



A major difference between the meaning of a declarative sentence like *David likes Chris* and the meaning of the question *Who does David like?* is the type of the expressions representing their meaning. The expression in (102) is of type $\langle t \rightarrow t \rangle$: it represents a set of possible answer meanings such as $\mathit{like}(\mathit{David}, \mathit{Chris})$, each of which has the type t of a declarative sentence. Up to now, we have associated only the basic types e and t with semantic structures, not more complex types like $\langle t \rightarrow t \rangle$. We can continue to associate only basic types with semantic structures if the meaning of a question is associated not with a single semantic structure but with an implicational meaning constructor, one whose right-hand side has the form $A \multimap B$. However, we have argued in Chapter 9 that a semantically complete and coherent glue derivation results in a meaning constructor that is *not* implicational, so that some refinement to the definition of semantic completeness and coherence would need to be provided.

An alternative approach is to permit higher types such as $\langle t \rightarrow t \rangle$ to be associated with semantic structures. For example, the meaning representation in (102) is of type $\langle t \rightarrow t \rangle$, and it would be associated with the semantic structure f_σ in example (104). In that case, we must provide a theory that accounts for the discrepancy between the question type $\langle t \rightarrow t \rangle$ and the basic type t that is associated with a declarative sentence whose head is the verb *likes*. It is unclear what additional issues might arise from allowing semantic structures to be associated with higher types, though no obvious obstacles present themselves.

Finally, a desirable characteristic of any analysis of wh-questions is that it should extend unproblematically to *multiple wh-questions*. Questions can contain more than one interrogative pronoun:

(105) *Who likes who?*

$$\lambda P.[\exists X.\exists Y.(\mathit{person}(X) \wedge \mathit{person}(Y) \wedge P = \mathit{like}(X, Y))]$$

The meaning constructors that are relevant in the analysis of the question *Who does David like?* should be reusable straightforwardly and without augmentation to produce the meaning representation in (105).

In sum, the best analysis of question semantics and semantic composition in a glue setting is obtained by assuming that the meaning constructor for an interrogative pronoun like *who* combines seamlessly with the independently motivated semantic contributions of the other meaning-bearing items in the sentence to produce the desired meaning constructor, just as the meaning constructors for the relative pronoun and the relative clause combine with the other meaning constructors in the relative clause to produce an appropriate relative clause meaning constructor. It may be that some basic assumptions about meaning representations, semantic types, or other aspects of the glue approach will require some degree of modification to give an adequate account of the meanings of questions and other nondeclaratives; future research will reveal more about how these issues should be resolved.

6. FURTHER READING AND RELATED ISSUES

There has been much work exploring the syntactic properties of long-distance dependency constructions in LFG. In current work, as is evident from the discussion in this chapter, there is no general agreement on whether some or all long-distance dependencies involve the presence of *traces*, phonologically empty constituent structure categories that appear in the within-clause position corresponding to the displaced phrase in a long-distance dependency. Future work may help to resolve this question.

A very interesting discussion of the syntax of long-distance dependencies is provided by Falk (2001, Chapter 6), including an analysis of so-called "that-trace" effects in examples like:

(106) *Who do you think (*that) put the book on the shelf?*

Falk proposes an analysis that differs from the one proposed in this chapter in assuming the existence of traces for some (but not all) long-distance dependencies.

The syntactic discussion in this chapter has concentrated primarily on syntactic constraints on long-distance dependencies in topicalization, relativization, and question formation, and we have not discussed relative clause and question constructions that do not involve long-distance dependencies. Culy (1990) discusses *internally-headed relative clauses*, which he defines as follows:

(107) A (restrictive) internally headed relative clause is a nominalized sentence which modifies a nominal, overt or not, internal to the sentence. (Culy 1990, page 27)

Culy gives the following Donno Sɔ example of an internally-headed relative clause:

(108) *[kandɔw nyan gɛ tɛgɔ] nɛ yu gaw to*
just fire granary burn.PAST.3SG.DEF in millet a lot exist.3SG
'There was a lot of millet in the granary that the fire just burnt.'
(i.e., the granary was burnt)

In this example, the clausal constituent *kandɔw nyan gɛ tɛgɔ* 'the fire just burned the granary' is the object of the postposition *nɛ* 'in' and behaves like the English phrase *the granary that the fire just burned.* Culy gives LFG as well as Head-Driven Phrase Structure Grammar syntactic analyses of internally headed relative clauses.

In many languages, wh-questions do not involve a long-distance dependency, and the interrogative pronoun is not displaced or fronted. In his analysis of Mandarin Chinese wh-questions, Huang (1993) shows that although these constructions do not involve long-distance dependencies, the same sorts of scope phenomena found in English wh-questions can also be found in Chinese. Huang uses inside-out functional uncertainty (Chapter 6, Section 1.2) in his analysis.

Other constructions have also been shown to involve long-distance dependencies in English and other languages. In particular, the English *tough* construction involves an arbitrarily deeply embedded position in the infinitival complement clause:

(109) *This book is tough to convince anyone to try to read* ___.

LFG-based analyses of the English *tough* construction have been proposed by Kaplan and Bresnan (1982), Barron (1999), and Dalrymple and King (2000). Huang (1997) discusses a similar construction in Chinese, and Saiki (1991) and Yamamoto (1996) discuss a similar Japanese construction.

15

RELATED RESEARCH THREADS AND NEW DIRECTIONS

This book has presented the syntactic theory of Lexical Functional Grammar and the relation of LFG's syntactic structures to other linguistic structures. We concentrated in particular detail on semantics and the syntax-semantics interface, providing both syntactic and semantic analyses of a variety of linguistic constructions. Much work remains to be done on the constructions we examined as well as on topics that have unfortunately gone unexamined in this book. In this final chapter, we turn to a brief discussion of some LFG work that has not been touched on in earlier chapters.

1. ALGORITHMIC ISSUES: PARSING AND GENERATION

In an LFG computational linguistic setting, *parsing* traditionally means providing all possible c-structure/f-structure pairs for a given string of words, and *generation* means finding the strings of words that correspond to a given f-structure. As described in Chapter 7 and Chapter 9, more recent work in LFG explores a number of interrelated facets of linguistic structure, termed *projections*, including morphosyntactic structure, semantic structure, and more. Under these assumptions, the result of parsing a string is the set of structures for a given input string and the correspondence functions that relate them; for instance, the result of parsing might be a c-structure, an f-structure, a morphosyntactic structure, and a semantic structure. In generation, the input need not be a syntactic representation like the f-structure; semantic input can also be analyzed, and in this case generation involves determining the syntactic structures corresponding to the semantic input as well as the string that expresses that meaning.

1.1. Parsing

Much important work has been done on the theory of parsing with LFG grammars as well as on efficient parsing algorithms, some of which is summarized in Dalrymple et al. (1995c). More recently, significant breakthroughs have been made on two fronts.

Maxwell and Kaplan (1991) examine the problem of processing disjunctive specifications of constraints, which are computationally very difficult to process: in the worst case, processing disjunctive constraints is *exponentially difficult*, meaning that the time needed to process the constraints can increase exponentially as the number of constraints increases.[1] However, this worst-case scenario assumes that every disjunctive constraint can interact significantly with every other constraint. In linguistic processing, such interactions are found only very rarely; in fact, interactions of disjunctive constraints are almost always locally confined. For example, an ambiguity in the syntactic properties of the SUBJ of a sentence rarely correlates with ambiguities in the OBJ or other arguments. This insight is the basis of Maxwell and Kaplan's algorithm, which works by turning a set of disjunctively specified constraints into a set of *contexted*, conjunctively specified constraints, where the context of a constraint indicates where the constraint is relevant. Solving these contexted constraints turns out to be very efficient for linguistically motivated sets of constraints, where only local interactions among disjunctions tend to occur.

The second breakthrough was made by Maxwell and Kaplan (1993, 1996), who explore the issue of c-structure processing and its relation to solving f-structural constraints. It has long been known that constituent structure parsing — determining the phrase structure trees for a given sentence — is very fast in comparison to solving the equations that determine the f-structure for the sentence. For this reason, an important task in designing algorithms for linguistic processing

[1] An exponential increase can be thought of in terms of some constant number raised to the$n$th power, where n is the number of constraints. For example, if solving one constraint takes$2^1 = 2$ seconds, solving two constraints could take $2^2 = 4$ seconds, and solving ten constraints could take $2^{10} = 1024$ seconds or about 17 minutes.

of different kinds of structures like the c-structure and the f-structure is to optimize the interactions between these computationally very different tasks. Previous research often assumed that the most efficient approach would be to interleave the construction of the phrase structure tree with the solution of f-structure constraints. Maxwell and Kaplan (1993) explore and compare a number of different methods for combining phrase structure processing with constraint solving; they show that in certain situations, interleaving the two processes can actually give very bad results. In subsequent work, Maxwell and Kaplan (1996) showed that if phrase structure parsing and f-structural constraint solving are combined in the right way, parsing can be very fast; in fact, if the grammar that results from combining phrase structure and functional constraints happens to be *context-free equivalent*,[2] the algorithm for computing the c-structure and f-structure operates in *cubic time*, the same as for pure phrase structure parsing.

1.2. Generation

Work on generation in LFG generally assumes that the generation task is to determine the strings of a language that correspond to a specified f-structure, given a particular grammar. Based on these assumptions, several interesting theoretical results have been attained.

Wedekind (1995, 1999) addresses the issue of the decidability of generation from f-structures: the problem of determining whether there is any sentence that corresponds to a given f-structure according to a given grammar. Wedekind (1995) demonstrates that the problem is decidable for *fully specified* f-structures: if we assume that the f-structure we are given is complete and no additional features can be added, we can always determine whether or not there is a sentence that corresponds to that f-structure. Wedekind (1999) shows that the corresponding problem for *underspecified* f-structures is not decidable: it is not always possible to determine whether there is a sentence that corresponds to a given f-structure if we are allowed to add additional attributes and values to the f-structure.

In further work on the formal properties of generation from f-structures, Kaplan and Wedekind (2000) show that if we are given an LFG grammar and an *acyclic*[3] f-structure, the set of strings that corresponds to that f-structure according to the grammar is a *context-free language*. Kaplan and Wedekind also provide a method for constructing the context-free grammar for that set of strings by a process of specialization of the full grammar that we are given. This result leads to a new way of thinking about generation, opens the way to new, more efficient generation

[2] The formal properties of context-free languages and their grammars, context-free grammars, are described in Partee et al. (1993, Chapter 16).

[3] An *acyclic* f-structure is one in which no f-structure contains a path leading back to itself. Kaplan and Bresnan (1982) suggest that acyclic structures are the only f-structures that are motivated for linguistic analysis.

algorithms, and clarifies a number of formal and mathematical issues relating to LFG parsing and generation.

Wedekind and Kaplan (1996) explore issues in *ambiguity-preserving generation*, where a set of f-structures rather than a single f-structure is considered, and the sentences of interest are those that correspond to all of the f-structures under consideration; Shemtov (1997) also explores issues in ambiguity management and ambiguity preservation in generation from sets of f-structures. The potential practical advantages of ambiguity-preserving generation are clear: consider, for example, a scenario involving translation from English to German. We first parse the input English sentence, producing several f-structures if the English sentence is ambiguous. For instance, the English sentence *Hans saw the man with the telescope* is ambiguous: it means either that the man had the telescope or that Hans used the telescope to see the man. The best translation for this sentence would be a German sentence that is ambiguous in exactly the same way as the English sentence, if such a German sentence exists; in the case at hand, we would like to produce the German sentence *Hans sah den Mann mit dem Fernrohr*, which has exactly the same two meanings as the English input. To do this, we map the English f-structures for the input sentence to the set of corresponding German f-structures; our goal is then to generate the German sentence *Hans sah den Mann mit dem Fernrohr*, which corresponds to each of these f-structures. Though this approach is appealing, Wedekind and Kaplan (1996) show that determining whether there is a single sentence that corresponds to each member of a set of f-structures is in general *undecidable* for an arbitrary (possibly linguistically unreasonable) LFG grammar: there are LFG grammars and sets of f-structures for which it is impossible to determine whether there is any sentence that corresponds to those f-structures. This result is important in understanding the limits of ambiguity-preserving generation.

2. PSYCHOLOGICAL REALITY: PROCESSING AND ACQUISITION

One of the original aims of LFG was to produce a *psychologically realistic* linguistic theory, one that would not only account for observed patterns of linguistic behavior but would also provide insight into the mental representation of language. Bresnan and Kaplan (1982) enumerate a list of constraints on the psycholinguistic problem of linguistic parsing, how a speaker determines the structure of a string of words:

Creativity: The theory must account for the fact that the speaker can understand and produce entirely novel sentences.

Finite Capacity: The theory must be capable of producing an unbounded number of possible sentences from a finite number of words and rules.

Reliability: The theory must provide a nonarbitrary, reliably computable, and consistent method for deciding on the structure of a sentence.

Order-Free Composition: The theory must explain how a speaker can produce coherent analyses for arbitrary or incomplete fragments of sentences.

Universality: The theory must incorporate a universal, effective means for determining the grammar of the language that the speaker encounters.

The theory of LFG meets these desiderata. Work on psycholinguistic processing in LFG was pioneered by Ford et al. (1982), and Pinker (1982, 1989) studied issues of language acquisition in an LFG setting.

In more recent work, a new view of language processing and acquisition has emerged in the framework of *Data-Oriented Parsing* or *DOP* (Bod 1998). DOP views language acquisition as the analysis of a pool of linguistic structures that are presented to the language learner. The learner breaks up these structures into their component pieces, and new utterances are assembled from these pieces. The likelihood of assigning a particular analysis to a new sentence depends on the frequency of occurrence of its component parts in the original pool of structures.

LFG-DOP (Bod and Kaplan 1998; Cormons 1999) specializes the general DOP theory to LFG assumptions about linguistic structures and the relations between them. LFG-DOP assumes that the body of linguistic evidence that a language learner is presented with consists of wellformed c-structure/f-structure pairs, and that language acquisition consists in determining the relevant component parts of these structures and then combining these parts to produce new c-structure/f-structure pairs for novel sentences. The field of LFG-DOP is a dynamic and rapidly expanding one: Bod (2000a) provides an evaluation of the framework, and Bod (2000b) proposes a parser for LFG-DOP linguistic analysis. The theory is also applied in work on translation by Way (1999, 2001).

3. LFG AND OPTIMALITY THEORY

Much recent research in phonology, morphology, syntax, and semantics has been conducted in the framework of *Optimality Theory* or *OT*. OT-based analyses assume that the grammar of a language consists of a *generator* component that proposes candidate linguistic structures for an input and an *evaluation* component that selects the optimal structure from these candidates. The evaluation component consists of a ranked set of universally valid constraints that the optimal analysis must meet.

In OT-LFG, the input is taken to be an underspecified f-structure, and the generator component is a "universal" LFG grammar that generates all wellformed c-structure/f-structure pairs that are compatible with the input. The evaluation component determines the optimal candidate in a particular language from this set. Bresnan (2001c) gives a useful overview of the theory, and Bresnan (2000) provides an OT-LFG analysis of competition between different linguistic structures. A number of other papers incorporating OT-LFG analyses have already been mentioned in earlier chapters; besides these, work by Vincent (2001) on diachronic syntax from an OT-LFG perspective is of particular interest.

Important work on the formal properties of OT-LFG has also been done. Johnson (1998) discusses formal and algorithmic issues in OT-LFG parsing. Responding to issues raised by Johnson, Kuhn (2001a) proves that parsing in OT-LFG is decidable under certain reasonable assumptions and resolves a number of other difficult formal issues with the theory. A clear understanding of the formal properties of generation is crucial in analyzing the formal properties of OT-LFG; the results outlined in Section 1.2 of this chapter are particularly important.

APPENDIX: PROOF RULES FOR LINEAR LOGIC

The following table of proof rules for the glue fragment of linear logic is adapted from Crouch and van Genabith (2000).

$$\multimap \qquad \frac{\begin{array}{c}[A]^i\\ \vdots\\ B\end{array}}{A \multimap B} \qquad\qquad \frac{A \qquad A \multimap B}{B}$$

$$\otimes \qquad \frac{A \quad B}{A \otimes B} \qquad\qquad \frac{A \otimes B \qquad \begin{array}{c}[A]^i[B]^j\\ \vdots\\ C\end{array}}{C}$$

$$\forall \qquad \frac{A}{\forall X.A[X/a]} \qquad\qquad \frac{\forall X.A}{A[a/X]}$$

provided a does not occur in any assumptions that A depends on.

In these rules, $[\,]^i$ indicates the discharge of a hypothesis labeled i.

The rules for the *of course* connective "!" used in our analysis of coordination in Chapter 13 are:

Weakening

$$\frac{!A \qquad B}{B}$$

Dereliction

$$\frac{!A}{A}$$

Contraction

$$\frac{!A \qquad \begin{array}{c}[!A]^i[!A]^j \\ \vdots \\ C\end{array}}{C}$$

BIBLIOGRAPHY

The web addresses in this bibliography were valid as of February 2001. Since the web changes rapidly, however, some of these addresses may not remain permanently valid.

Abney, S. P. (1987). *The English Noun Phrase in Its Sentential Aspect.* Doctoral dissertation, MIT.

Ackerman, F. (1987). *Miscreant Morphemes: Phrasal Predicates in Ugric.* Doctoral dissertation, University of California, Berkeley.

Ackerman, F. (1990). Locative alternation vs. locative inversion. In *Proceedings of the West Coast Conference on Formal Linguistics* (A. Halpern, ed.), volume 9, 1–13. CSLI Publications, Stanford, CA.

Ackerman, F. (1992). Complex predicates and morpholexical relatedness: Locative alternation in Hungarian. In *Lexical Matters* (I. Sag and A. Szabolcsi, eds.), 55–83. CSLI Publications, Stanford, CA. CSLI Lecture Notes, number 24.

Ackerman, F. and LeSourd, P. (1997). Toward a lexical representation of phrasal predicates. In *Complex Predicates* (A. Alsina, J. Bresnan, and P. Sells, eds.), 67–106. CSLI Publications, Stanford, CA.

Ackerman, F. and Moore, J. (1999). Telic entity as a proto-property of lexical predicates. In *On-line Proceedings of the LFG99 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/4/lfg99.html`.

Ackerman, F. and Moore, J. (2001). *Proto-Properties and Argument Selection: A Correspondence Theory of Grammatical Encoding.* CSLI Publications, Stanford University. To appear.

Ackerman, F. and Nikolaeva, I. (1997). Similar forms and different functions: The person/number paradigm in Western Armenian and North Ostyak. In *On-line*

Proceedings of the LFG97 Conference (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/2/lfg97.html`.

Ackerman, F. and Webelhuth, G. (1996). The construct PREDICATE: Empirical arguments and theoretical status. In *On-line Proceedings of the LFG96 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/1/lfg1.html`.

Ackerman, F. and Webelhuth, G. (1998). *A Theory of Predicates.* CSLI Publications, Stanford, CA.

Alsina, A. (1993). *Predicate Composition: A Theory of Syntactic Function Alternations.* Doctoral dissertation, Stanford University.

Alsina, A. (1995). The fall of function-argument biuniqueness. In *Proceedings of the Conference of the European Summer School in Logic, Language and Information* (G. Morrill and R. Oehrle, eds.), 1–16. Barcelona.

Alsina, A. (1996). *The Role of Argument Structure in Grammar: Evidence from Romance.* CSLI Publications, Stanford University.

Alsina, A. (1999). Where's the mirror principle? *The Linguistic Review* **16** (1), 1–42.

Andrews, A., III (1982). The representation of case in modern Icelandic. In *The Mental Representation of Grammatical Relations* (J. Bresnan, ed.), 427–503. The MIT Press, Cambridge, MA.

Andrews, A., III (1983a). *Constituent Coordination in LFG.* MS, Australian National University.

Andrews, A., III (1983b). A note on the constituent structure of modifiers. *Linguistic Inquiry* **14** (4), 695–697.

Andrews, A., III (1985). The major functions of the noun phrase. In *Language Typology and Syntactic Description* (T. Shopen, ed.). Cambridge University Press, Cambridge, England.

Andrews, A., III (1990a). Case structures and control in Modern Icelandic. In *Syntax and Semantics: Modern Icelandic Syntax* (J. Maling and A. Zaenen, eds.), volume 24, 187–234. Academic Press, San Diego.

Andrews, A., III (1990b). Unification and morphological blocking. *Natural Language and Linguistic Theory* **8** (4), 507–557.

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

Arka, I. W. (1993). *Morpholexical Aspects of the -kan Causative in Indonesian.* Master's thesis, University of Sydney.

Arka, I. W. (1998). *From Morphosyntax to Pragmatics in Balinese.* Doctoral dissertation, University of Sydney.

Arka, I. W. and Manning, C. D. (1998). On the three subjects in Indonesian: Evidence from binding. In *On-line Proceedings of the LFG98 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/3/lfg98.html`.

Arka, I. W. and Simpson, J. (1998). Control and complex arguments in Balinese. In *On-line Proceedings of the LFG98 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/3/lfg98.html`.

Arka, I. W. and Wechsler, S. (1996). Argument structure and linear order in Balinese binding. In *On-line Proceedings of the LFG96 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/1/lfg1.html`.

Asher, N. and Wada, H. (1988). A computational account of syntactic, semantic and discourse principles for anaphora resolution. *Journal of Semantics* **6** (3/4), 309–344.

Asudeh, A. (2000a). Functional identity and resource sensitivity in control. In *On-line Proceedings of the LFG2000 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/5/lfg00.html`.

Asudeh, A. (2000b). Functional identity and resource sensitivity in control. Handout for talk at LFG2000, University of California at Berkeley.

Asudeh, A. (2001a). Linking, optionality, and ambiguity in Marathi: An optimality theory analysis. In *Formal and Empirical Issues in Optimality Theoretic Syntax* (P. Sells, ed.). CSLI Publications, Stanford, CA. To appear.

Asudeh, A. (2001b). A resource-sensitive semantics for equi and raising. In *Proceedings of Semantics Fest 2000* (D. Beaver and S. Kaufmann, eds.). CSLI Publications, Stanford, CA. To appear.

Austin, P. and Bresnan, J. (1996). Non-configurationality in Australian Aboriginal languages. *Natural Language and Linguistic Theory* **14** (2), 215–268.

Baader, F., Bürckert, H., Nebel, B., Nutt, W., and Smolka, G. (1991). *On the expressivity of feature logics with negation, functional uncertainty and sort equations.* Technical Report RR-91-01, DFKI, Saarbrücken.

Bach, E. (1980). In defense of passive. *Linguistics and Philosophy* **3**, 297–341.

Bach, E. (1989). *Informal Lectures on Formal Semantics.* SUNY Series in Linguistics. State University of New York Press, Albany, NY.

Bader, T. (1995). *The Analysis of Passive and Raising Constructions in Head-Driven Phrase Structure Grammar and Arc Pair Grammar.* Doctoral dissertation, University of Bern, Switzerland.

Baker, M. (1983). Objects, themes and lexical rules in Italian. In *Papers in Lexical-Functional Grammar* (L. S. Levin, M. Rappaport, and A. Zaenen, eds.), 1–45. Indiana University Linguistics Club, Bloomington.

Bakhofen, R. (1993). On the decidability of functional uncertainty. In *Proceedings of the 31st Annual Meeting of the ACL,* Columbus, OH, 201–208. Association for Computational Linguistics.

Baltin, M. (1983). A landing site theory of movement rules. *Linguistic Inquiry* **13** (1), 1–38.

Barron, J. (1999). *Perception, Volition, and Reduced Clausal Complementation.* Doctoral dissertation, University of Manchester.

Barwise, J. (1979). On branching quantifiers in English. *Journal of Philosophical Logic* **8**, 47–80.

Barwise, J. (1987). Noun phrases, generalized quantifiers and anaphora. In *Generalized Quantifiers: Linguistic and Logical Approaches* (P. Gärdenfors, ed.), 1–30. D. Reidel, Dordrecht.

Barwise, J. and Cooper, R. (1981). Generalized quantifiers and natural language. *Linguistics and Philosophy* **4** (2), 159–219.

Barwise, J. and Perry, J. (1983). *Situations and Attitudes.* The MIT Press, Cambridge, MA.

Bech, G. (1957). *Studien über das deutsche verbum infinitivum.* Munksgaard, Copenhagen.

Bell, S. (1983). Advancements and ascensions in Cebuano. In *Studies in Relational Grammar 1* (D. M. Perlmutter, ed.), 143–218. University of Chicago Press, Chicago.

Berman, J. (1999). Does German satisfy the Subject Condition? In *On-line Proceedings of the LFG99 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/4/lfg99.html`.

Berman, J. (2000). *Topics in the Clausal Syntax of German.* Doctoral dissertation, University of Stuttgart, Stuttgart, Germany.

Bloomfield, L. (1933). *Language.* George Allen and Unwin, London. Reprinted in 1961, Holt, Reinhart, and Winston.

Bod, R. (1998). *Beyond Grammar.* CSLI Publications, Stanford University.

Bod, R. (2000a). An empirical evaluation of LFG-DOP. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING2000),* Saarbrücken.

Bod, R. (2000b). An improved parser for Data-Oriented Lexical-Functional analysis. In *Proceedings of the 38th Annual Meeting of the ACL,* Hong Kong.

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

Bodomo, A. B. (1996). Complex verbal predicates: The case of serial verbs in Dagaare and Akan. In *On-line Proceedings of the LFG96 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/1/lfg1.html`.

Bodomo, A. B. (1997). *Paths and Pathfinders: Exploring the Syntax and Semantics of Complex Verbal Predicates in Dagaare and Other Languages.* Doctoral dissertation, Norwegian University of Science and Technology.

Börjars, K. (1998). Clitics, affixes, and parallel correspondences. In *On-line Proceedings of the LFG98 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/3/lfg98.html`.

Börjars, K., Chisarik, E., and Payne, J. (1999). On the justification for Functional Categories in LFG. In *On-line Proceedings of the LFG99 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/4/lfg99.html`.

Börjars, K., Vincent, N., and Chapman, C. (1997). Paradigms, pronominal inflection and periphrasis. In *Yearbook of Morphology* (G. de Booij and J. van Marle, eds.), 155–180. Kluwer Academic Publishers.

Bos, J., Mastenbroek, E., McGlashan, S., Millies, S., and Pinkal, M. (1994). A compositional DRS-based formalism for NLP-applications. In *International Workshop on Computational Semantics.* Tilburg. Also published as *Verbmobil Report 59,* Universität des Saarlandes, Saarbrücken, Germany.

Brame, M. (1982). The head-selector theory of lexical specifications and the nonexistence of coarse categories. *Linguistic Analysis* **10** (4), 321–325.

Bresnan, J. (1976). On the form and functioning of transformations. *Linguistic Inquiry* **7** (1), 3–40.

Bresnan, J. (1978). A realistic transformational grammar. In *Linguistic Theory and Psychological Reality* (M. Halle, J. Bresnan, and G. A. Miller, eds.). The MIT Press, Cambridge, MA.

Bresnan, J. (1980). Polyadicity. In *Lexical Grammar* (T. Hoekstra, H. van der Hulst, and M. Moortgat, eds.), 97–121. Foris Publications, Dordrecht. Republished in revised form in Bresnan (1982b).

Bresnan, J. (1982a). Control and complementation. In *The Mental Representation of Grammatical Relations* (J. Bresnan, ed.), 282–390. The MIT Press, Cambridge, MA.

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

Bresnan, J. (1982c). The passive in lexical theory. In *The Mental Representation of Grammatical Relations* (J. Bresnan, ed.), 3–86. The MIT Press, Cambridge, MA.

Bresnan, J. (1984). *Bound Anaphora on Functional Structures.* Presented at the Tenth Annual Meeting of the Berkeley Linguistics Society, February 17, 1984.

Bresnan, J. (1990). Monotonicity and the theory of relation-changes in LFG. *Language Research* **26** (4), 637–652.

Bresnan, J. (1994). Locative inversion and the architecture of Universal Grammar. *Language* **70** (1), 72–131.

Bresnan, J. (1995). Linear order, syntactic rank, and empty categories: On weak crossover. In *Formal Issues in Lexical-Functional Grammar* (M. Dalrymple, R. M. Kaplan, J. T. Maxwell, and A. Zaenen, eds.), 241–274. CSLI Publications, Stanford University.

Bresnan, J. (1997). Mixed categories as head sharing constructions. In *On-line Proceedings of the LFG97 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/2/lfg97.html`.

Bresnan, J. (1998). Morphology competes with syntax: Explaining typological variation in weak crossover effects. In *Is the Best Good Enough? Proceedings from the Workshop on Optimality in Syntax* (P. Barbosa, D. Fox, P. Hagstrom, M. McGinnis, and D. Pesetsky, eds.). The MIT Press. Invited paper presented at the MIT Workshop on Optimality in Syntax, May 1995.

Bresnan, J. (2000). Explaining morphosyntactic competition. In *Handbook of Contemporary Syntactic Theory* (M. Baltin and C. Collins, eds.). Blackwell Publishers, Oxford.

Bresnan, J. (2001a). The emergence of the unmarked pronoun. In *Optimality-Theoretic Syntax* (G. Legendre, S. Vikner, and J. Grimshaw, eds.). The MIT Press. To appear.

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

Bresnan, J. (2001c). Optimal syntax. In *Optimality Theory: Phonology, Syntax and Acquisition* (J. Dekkers, F. van der Leeuw, and J. van de Weijer, eds.). Oxford University Press, Oxford.

Bresnan, J., Halvorsen, P.-K., and Maling, J. (1985a). *Logophoricity and Bound Anaphors.* MS, Stanford University.

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

Bresnan, J. and Kaplan, R. M. (1982). Introduction: Grammars as mental representations of language. In *The Mental Representation of Grammatical Relations*, xvii–lii. The MIT Press, Cambridge, MA.

Bresnan, J., Kaplan, R. M., and Peterson, P. (1985b). *Coordination and the Flow of Information through Phrase Structure.* MS, Xerox PARC.

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

Bresnan, J. and Mchombo, S. A. (1995). The lexical integrity principle: Evidence from Bantu. *Natural Language and Linguistic Theory* **13** (2), 181–254.

Bresnan, J. and Moshi, L. (1990). Object asymmetries in comparative Bantu syntax. *Linguistic Inquiry* **21** (2), 147–186.

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

Broadwell, G. A. (1998). Directionals as complex predicates in Choctaw. In *On-line Proceedings of the LFG98 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/3/lfg98.html`.

Brun, C. (1996a). La coordination dans le cadre d'une grammaire LFG du français. In *Proceedings of the First Student Conference in Computational Linguistics in Montreal*, 192–199. Montréal.

Brun, C. (1996b). *Using Priority Union for Non-constituent Coordination in LFG.* Presented at the LFG Workshop, Grenoble, France, August 1996.

Butt, M. (1996). *The Structure of Complex Predicates in Urdu.* Doctoral dissertation, Stanford University. *Dissertations in Linguistics* series, CSLI Publications, Stanford University. Revised and corrected version of 1993 Stanford University dissertation.

Butt, M. (1998). Constraining argument merger through aspect. In *Syntax and Semantics: Complex Predicates in Nonderivational Syntax* (E. Hinrichs, A. Kathol, and T. Nakazawa, eds.), volume 30, 73–113. Academic Press, New York.

Butt, M., Dalrymple, M., and Frank, A. (1997). The nature of argument structure. In *On-line Proceedings of the LFG97 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/2/lfg97.html`.

Butt, M. and King, T. H. (1996). Structural topic and focus without movement. In *On-line Proceedings of the First LFG Conference, Rank Xerox, Grenoble, August 26–28, 1996* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/1/lfg1.html`.

Butt, M. and King, T. H. (1998). Interfacing phonology with LFG. In *On-line Proceedings of the LFG98 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/3/lfg98.html`.

Butt, M. and King, T. H. (1999a). *Licensing Semantic Case.* MS, University of Konstanz and Xerox PARC.

Butt, M. and King, T. H. (1999b). *The Status of Case.* MS, University of Konstanz and Xerox PARC. Submitted as a contribution to *Clause Structure in South Asian Languages*, to be edited by Veneeta Dayal and Anoop Mahajan, in preparation.

Butt, M. and King, T. H. (2000). Null elements in discourse structure. In *Papers from the NULLS Seminar* (K. V. Subbarao, ed.). Motilal Banarasidas, Delhi.

Butt, M., King, T. H., Niño, M., and Segond, F. (1999). *A Grammar Writer's Cookbook.* CSLI Publications, Stanford University.

Butt, M., Niño, M.-E., and Segond, F. (1996). Multilingual processing of auxiliaries in LFG. In *Natural Language Processing and Speech Technology: Results of the 3rd KONVENS Conference* (D. Gibbon, ed.), 111–122. Mouton De Gruyter. Universität Bielefeld, 7 - 9 October 1996.

Chierchia, G. (1984). Anaphoric properties of infinitives and gerunds. In *Proceedings of the Third West Coast Conference on Formal Linguistics* (M. Cobler, S. MacKaye, and M. Wescoat, eds.), 28–39. Stanford Linguistics Association, Stanford University.

Chierchia, G. (1985). Formal semantics and the grammar of predication. *Linguistic Inquiry* **16** (3), 417–443.

Choi, H. (1999). *Optimizing Structure in Context: Scrambling and Information Structure.* Doctoral dissertation, Stanford University. *Dissertations in Linguistics* series, CSLI Publications, Stanford University. Revised and corrected version of 1996 Stanford University dissertation.

Chomsky, N. (1955). *The Logical Structure of Linguistic Theory.* Plenum Press, New York. Published in 1975.

Chomsky, N. (1957). *Syntactic Structures.* Mouton, The Hague.

Chomsky, N. (1965). *Aspects of the Theory of Syntax.* The MIT Press, Cambridge, MA.

Chomsky, N. (1970). Remarks on nominalization. In *Readings in English Transformational Grammar* (R. Jacobs and P. Rosenbaum, eds.). Ginn.

Chomsky, N. (1981). *Lectures on Government and Binding.* Foris Publications, Dordrecht.

Chomsky, N. (1986). *Barriers.* The MIT Press, Cambridge, MA.

Chomsky, N. (1993). A minimalist program for linguistic theory. In *The View from Building 20: Essays in Linguistics in Honor of Sylvain Bromberger*, 1–52. The MIT Press, Cambridge, MA.

Chung, S. (1982). Unbounded dependencies in Chamorro grammar. *Linguistic Inquiry* **13** (1), 39–78.

Cinque, G. (1990). *Types of Ā-Dependencies.* Linguistic Inquiry Monographs. The MIT Press, Cambridge, MA.

Clements, G. N. and Ford, K. C. (1979). Kikuyu tone shift and its synchronic consequences. *Linguistic Inquiry* **10** (2), 179–210.

Colban, E. (1987). Prepositional phrases in situation schemata. In *Situations, Language and Logic* (J. E. Fenstad, P.-K. Halvorsen, T. Langholm, and J. van Benthem, eds.). D. Reidel, Dordrecht.

Comrie, B. (1975). Causatives and universal grammar. *Transactions of the Philological Society 1974* , 1–32.

Corbett, G. G. (1983). Resolution rules: Agreement in person, number, and gender. In *Order, Concord, and Constituency* (G. Gazdar, E. Klein, and G. K. Pullum, eds.), 175–206. Foris Publications, Dordrecht.

Corbett, G. G. (1991). *Gender.* Cambridge University Press, Cambridge, England.

Cormons, B. (1999). *Analyse et désambiguïsation: Une approche à base de corpus (Data-Oriented Parsing) pour les représentations lexicales fonctionnelles.* Doctoral dissertation, Université de Rennes, France.

Covington, M. A. (1984). *Syntactic Theory in the High Middle Ages.* Cambridge University Press, Cambridge, England.

Crouch, R. (1999). Ellipsis and glue languages. In *Fragments: Studies in Ellipsis and Gapping* (S. Lappin and E. Benmamoun, eds.). Oxford University Press, Oxford.

Crouch, R., Frank, A., and van Genabith, J. (2001). Glue, underspecification and translation. In *Computing Meaning* (H. Bunt and R. Muskens, eds.), volume 2. Kluwer Academic Publishers. In press.

Crouch, R. and van Genabith, J. (1999). Context change, underspecification and the structure of glue language derivations. In *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach* (M. Dalrymple, ed.). The MIT Press, Cambridge, MA.

Crouch, R. and van Genabith, J. (2000). *Linear Logic for Linguists: ESSLLI-2000 course notes.* ESSLLI, Birmingham, UK. URL `www.cs.bham.ac.uk/~esslli/notes/crouch`. MS in preparation with CSLI Press.

Culy, C. D. (1990). *The Syntax and Semantics of Internally Headed Relative Clauses.* Doctoral dissertation, Stanford University, Department of Linguistics.

Culy, C. D. (1991). Command and Fula ɗum pronominals. In *Proceedings of the Special Session on African Linguistics, Berkeley Linguistics Society.* Berkeley Linguistics Society, University of California, Berkeley.

Culy, C. D. (1994). *Extraposition in English and Grammatical Theory.* MS, University of Iowa.

Culy, C. D. (1996). Agreement and Fula pronouns. *Studies in African Linguistics* **25** (1), 1–27.

Culy, C. D. (2000). An incorporated topic marker in Takelma. In *On-line Proceedings of the LFG2000 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/5/lfg00.html`.

Culy, C. D., Kodio, K., and Togo, P. (1995). Dogon pronominal systems: Their nature and evolution. *Studies in African Linguistics* **23** (3), 315–344.

Dahlstrom, A. (1986a). Nominal arguments and pronominal inflection in Fox. Written version of paper presented at the 61st Annual Meeting of the LSA, New York, 1986.

Dahlstrom, A. (1986b). *Plains Cree morphosyntax.* Doctoral dissertation, University of California, Berkeley.

Dahlstrom, A. (1987). Discontinuous constituents in Fox. In *Native American Languages and Grammatical Typology* (R. E. Moore and P. D. Kroeber, eds.). Indiana University Linguistics Club, Bloomington.

Dahlstrom, A. (1997). *Discontinuity in morphology and syntax: Separable prefixes in Fox (Mesquakie).* Presented at LFG97, San Diego, June 1997.

Dalrymple, M. (1993). *The Syntax of Anaphoric Binding.* CSLI Lecture Notes, number 36. CSLI Publications, Stanford University.

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

Dalrymple, M., Gupta, V., Lamping, J., and Saraswat, V. (1999a). Relating resource-based semantics to categorial semantics. In *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach* (M. Dalrymple, ed.). The MIT Press, Cambridge, MA.

Dalrymple, M., Hinrichs, A., Lamping, J., and Saraswat, V. (1993). The resource logic of complex predicate interpretation. In *Proceedings of the 1993 Republic of China Computational Linguistics Conference (ROCLING)* (K.-J. Chen and C.-R. Huang, eds.). Computational Linguistics Society of Republic of China, Hsitou National Park, Taiwan. Also published as Xerox Technical Report ISTL-NLTT-1993-08-03.

Dalrymple, M., Kanazawa, M., Kim, Y., Mchombo, S. A., and Peters, S. (1997a). Reciprocal expressions and the concept of reciprocity. *Linguistics and Philosophy* **21** (2), 159–210.

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

Dalrymple, M., Kaplan, R. M., Maxwell, J. T., III, and Zaenen, A. (1995a). Formal architecture. In *Formal Issues in Lexical-Functional Grammar* (M. Dalrymple, R. M. Kaplan, J. T. Maxwell, III, and A. Zaenen, eds.), 1–5. CSLI Publications, Stanford University.

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

Dalrymple, M., Kaplan, R. M., Maxwell, J. T., III, and Zaenen, A. (1995c). Mathematical and computational issues. In *Formal Issues in Lexical-Functional Grammar* (M. Dalrymple, R. M. Kaplan, J. T. Maxwell, III, and A. Zaenen, eds.), 331–338. CSLI Publications, Stanford University.

Dalrymple, M., Kaplan, R. M., Maxwell, J. T., III, and Zaenen, A. (1995d). Nonlocal dependencies. In *Formal Issues in Lexical-Functional Grammar* (M. Dalrymple, R. M. Kaplan, J. T. Maxwell, III, and A. Zaenen, eds.), 131–135. CSLI Publications, Stanford University.

Dalrymple, M. and King, T. H. (2000). Missing-object constructions: Lexical and constructional variation. In *On-line Proceedings of the LFG2000 Conference* (M. Butt and T. H. King, eds.).

Dalrymple, M., Lamping, J., Pereira, F., and Saraswat, V. (1999b). Overview and introduction. In *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach* (M. Dalrymple, ed.). The MIT Press, Cambridge, MA.

Dalrymple, M., Lamping, J., Pereira, F. C. N., and Saraswat, V. (1995e). Linear logic for meaning assembly. In *Proceedings of Computational Logic for Natural Language Processing* (S. Manandhar, G. P. Lopes, and W. Nutt, eds.). Edinburgh.

Dalrymple, M., Lamping, J., Pereira, F. C. N., and Saraswat, V. (1997b). Quantifiers, anaphora, and intensionality. *Journal of Logic, Language, and Information* **6** (3), 219–273. Reprinted in Dalrymple (1999).

Dalrymple, M. and Lødrup, H. (2000). The grammatical functions of complement clauses. In *On-line Proceedings of the LFG2000 Conference* (M. Butt and T. H. King, eds.).

Dalrymple, M., Shieber, S. M., and Pereira, F. C. N. (1991). Ellipsis and higher-order unification. *Linguistics and Philosophy* **14** (4), 399–452.

Dalrymple, M. and Zaenen, A. (1991). Modeling anaphoric superiority. In *Proceedings of the International Conference on Current Issues in Computational Linguistics*, 235–247. Penang, Malaysia.

Davison, A. (1998). Structural case, lexical case and complex predicates. *South Asian Language Review VIII* .

Dowty, D. R. (1982). Grammatical relations and Montague Grammar. In *The Nature of Syntactic Representation* (P. Jacobson and G. K. Pullum, eds.), 79–130. D. Reidel, Dordrecht.

Dowty, D. R. (1991). Thematic proto roles and argument selection. *Language* **67** (3), 547–619.

Dryer, M. S. (1987). Primary objects, secondary objects, and antidative. *Language* **62** (4), 808–845.

Dyvik, H. (1999). The universality of f-structure: Discovery or stipulation? The case of modals. In *On-line Proceedings of the LFG99 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/4/lfg99.html`.

Emonds, J. E. (1976). *A Transformational Approach to English Syntax.* Academic Press, New York.

Engdahl, E. (1999). *The Choice Between* bli-*Passive and* s-*Passive in Danish, Norwegian and Swedish.* URL `www.ling.gu.se/~engdahl/passive2.ps`. MS, Goteborg University.

Evans, G. (1980). Pronouns. *Linguistic Inquiry* **11** (2), 337–362.

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

Falk, Y. N. (1984). The English auxiliary system: A Lexical-Functional analysis. *Language* **60** (3), 483–509.

Falk, Y. N. (2000). Pivots and the theory of grammatical functions. In *On-line Proceedings of the LFG2000 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/5/lfg00.html`.

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

Faltz, L. M. (1977). *Reflexivization: A Study in Universal Syntax.* Doctoral dissertation, University of California, Berkeley. Reprinted by Garland Press, New York, 1985.

Fassi-Fehri, A. (1981). *Complémentation et Anaphore en Arabe Moderne: Une Approche Lexicale Fonctionelle.* Doctoral dissertation, Université de Paris III, Paris, France.

Fassi-Fehri, A. (1988). Agreement in Arabic, binding, and coherence. In *Agreement in Natural Language: Approaches, Theories, Descriptions* (M. Barlow and C. A. Ferguson, eds.). CSLI Publications, Stanford University.

Fenstad, J. E., Halvorsen, P.-K., Langholm, T., and van Benthem, J. (1987). *Situations, Language and Logic.* D. Reidel, Dordrecht.

Fillmore, C. J. (1968). The case for case. In *Universals in Linguistic Theory* (E. Bach and R. T. Harms, eds.), 1–210. Holt, Rinehart, and Winston, New York.

Ford, M., Bresnan, J., and Kaplan, R. M. (1982). A competence-based theory of syntactic closure. In *The Mental Representation of Grammatical Relations* (J. Bresnan, ed.), 727–796. The MIT Press, Cambridge, MA.

Frank, A. (1996). A note on complex predicate formation: Evidence from auxiliary selection, reflexivization, and past participle agreement in French and Italian. In *Online Proceedings of the LFG96 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/1/lfg1.html`.

Frank, A. and Zaenen, A. (2002). Tense in LFG: Syntax and morphology. In *How we say WHEN it happens: Contributions to the theory of temporal reference in natural language* (H. Kamp and U. Reyle, eds.). Niemeyer, Tübingen. To appear.

Franks, S. and King, T. H. (2000). *A Handbook of Slavic Clitics.* Oxford University Press.

Frantz, D. G. (1981). *Grammatical relations in Universal Grammar.* Indiana University Linguistics Club, Bloomington.

Frey, W. and Reyle, U. (1983). A PROLOG implementation of Lexical-Functional Grammar. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, 727–729. Karlsruhe.

Fry, J. (1999a). Proof nets and negative polarity licensing. In *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach* (M. Dalrymple, ed.), 91–116. The MIT Press, Cambridge, MA.

Fry, J. (1999b). *Resource-logical Event Semantics for LFG.* Presented at LFG99, University of Manchester.

Gabbay, D. M. (1996). *Labelled Deductive Systems.* Clarendon Press, Oxford.

Gamut, L. T. F. (1991a). *Logic, Language, and Meaning*, volume 1: Introduction to Logic. The University of Chicago Press, Chicago.

Gamut, L. T. F. (1991b). *Logic, Language, and Meaning*, volume 2: Intensional Logic and Logical Grammar. The University of Chicago Press, Chicago.

Gawron, J. M. and Peters, S. (1990). *Anaphora and quantification in Situation Semantics.* CSLI Lecture Notes, number 19. CSLI Publications, Stanford University.

Gazdar, G., Klein, E., Pullum, G. K., and Sag, I. A. (1985). *Generalized Phrase Structure Grammar.* Harvard University Press, Cambridge, MA.

Gazdar, G. and Pullum, G. K. (1981). Subcategorization, constituent order, and the notion 'head'. In *The Scope of Lexical Rules* (M. Moortgat, H. van der Hulst, and T. Hoekstra, eds.), 107–123. Foris Publications, Dordrecht.

Georgopoulos, C. (1985). Variables in Palauan syntax. *Natural Language and Linguistic Theory* **3** (1), 59–94.

Ginzburg, J. (1996). Interrogatives: Questions, facts, and dialogue. In *Handbook of Contemporary Semantic Theory* (S. Lappin, ed.). Blackwell Publishers, Oxford.

Ginzburg, J. (2001). *Questions, Queries, and Facts: A Semantics and Pragmatics for Interrogatives.* CSLI Publications. To appear, *Dissertations in Linguistics* series, CSLI Publications, Stanford University.

Girard, J. (1987). Linear logic. *Theoretical Computer Science* **50**, 1–102.

Givón, T., ed. (1979). *Syntax and Semantics: Discourse and Syntax*, volume 12. Academic Press, New York.

Givón, T. (1997). Grammatical relations: An introduction. In *Grammatical Relations: A Functional Perspective*. John Benjamins, Amsterdam.

Goodall, G. (1987). *Parallel Structures in Syntax*. Cambridge University Press, Cambridge, England.

Greenberg, J. H. (1966). Some universals of grammar with particular reference to the order of meaningful elements. In *Universals of Language* (J. H. Greenberg, ed.), 73–113. The MIT Press.

Grimes, B. F. (1999). *ETHNOLOGUE: Languages of the World*. SIL International. URL `www.sil.org/ethnologue/`.

Grimshaw, J. (1982). Subcategorization and grammatical relations. In *Subjects and Other Subjects* (A. Zaenen, ed.), 35–56. Indiana University Linguistics Club, Bloomington.

Grimshaw, J. (1990). *Argument Structure*. The MIT Press, Cambridge, MA.

Groenendijk, J. and Stokhof, M. (1984). *Studies on the Semantics of Questions and the Pragmatics of Answers*. Doctoral dissertation, University of Amsterdam.

Groenendijk, J. and Stokhof, M. (1991). Dynamic predicate logic. *Linguistics and Philosophy* **14** (1), 39–100.

Groenendijk, J. and Stokhof, M. (1997). Questions. In *Handbook of Logic and Language*, 1055–1124. The MIT Press/Elsevier, Cambridge/Amsterdam.

Grosz, B., Joshi, A., and Weinstein, S. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics* **2** (21), 203–225.

Gruber, J. S. (1965). *Studies in Lexical Relations*. Doctoral dissertation, MIT.

Hagége, C. (1974). Les pronoms logophoriques. *Bulletin de la Société de Linguistique de Paris* **69**, 287–310.

Hale, K. (1983). Warlpiri and the grammar of non-configurational languages. *Natural Language and Linguistic Theory* **1** (1), 5–47.

Hall, B. (1965). *Subject and Object in English*. Doctoral dissertation, MIT.

Halpern, A. (1995). *On the Placement and Morphology of Clitics*. Doctoral dissertation, Stanford University. *Dissertations in Linguistics* series, CSLI Publications, Stanford University. Revised and corrected version of 1992 Stanford University dissertation.

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

Halvorsen, P.-K. and Kaplan, R. M. (1988). Projections and semantic description in Lexical-Functional Grammar. In *Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS-88)*, 1116–1122. Tokyo. Reprinted in Dalrymple et al. (1995b, pp. 279–292).

Hamblin, C. L. (1958). Questions. *Australasian Journal of Philosophy* **36** (3), 159–168.

Hamblin, C. L. (1976). Questions in Montague English. In *Montague Grammar* (B. Partee, ed.), 247–259. Academic Press, New York. Originally published in *Foundations of Language* 10, 1973.

Heim, I. (1982). *The Semantics of Definite and Indefinite Noun Phrases.* Doctoral dissertation, University of Massachusetts at Amherst.

Hellan, L. (1988). *Anaphora in Norwegian and the Theory of Grammar.* Foris Publications, Dordrecht.

Henadeerage, K. (1998). Anaphoric binding in Colloquial Sinhala. In *On-line Proceedings of the LFG98 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/3/lfg98.html`.

Heny, F. (1973). Sentence and predicate modifiers in English. In *Syntax and Semantics* (J. P. Kimball, ed.), volume 2, 217–245. Academic Press, New York.

Her, O. (1999). Interaction of thematic structure and syntactic structures: On Mandarin dative alternations. In *Chinese Languages and Linguistics V: Interaction of Form and Function*, 373–412. Academia Sinica, Taipei.

Her, O. and Huang, H. (1998). *Locative Inversion: English, Chinese, and Universal Grammar.* URL `www-lfg.stanford.edu/lfg/archive/archive.html`. LFG Archive.

Higginbotham, J. (1983). Logical form, binding, and nominals. *Linguistic Inquiry* **14** (3), 395–420.

Higginbotham, J. (1992). Reference and control. In *Control and Grammar* (R. K. Larson, S. Iatridou, U. Lahiri, and J. Higginbotham, eds.). Kluwer Academic Publishers, Dordrecht.

Higginbotham, J. (1995). The semantics of questions. In *Handbook of Contemporary Semantic Theory* (S. Lappin, ed.). Blackwell Publishers, Oxford.

Hoeksema, J. (1988). The semantics of non-boolean 'and'. *Journal of Semantics* **6**, 19–40.

Hong, K. (1991). *Argument Selection and Case Marking in Korean.* Doctoral dissertation, Stanford University.

Hornstein, N. and Weinberg, A. (1995). The empty category principle. In *Government and Binding Theory and the Minimalist Program.* Blackwell Publishers, Oxford.

Huang, C.-R. (1990). A unification-based LFG analysis of lexical discontinuity. *Linguistics* **28** (2), 263–307.

Huang, C.-R. (1993). Reverse long-distance dependency and functional uncertainty: The interpretation of Mandarin questions. In *Language, Information, and Computing* (C. Lee and B.-M. Kang, eds.), 111–120. Thaehaksa, Seoul.

Huang, C.-R. (1997). Morphological transparency and autonomous morphology: A comparative study of tough constructions and nominalization. In *Chinese Languages and Linguistics III: Morphology and Lexicon*, 369–399. Academia Sinica.

Huang, C.-T. J. (1982). *Logical Relations in Chinese and the Theory of Grammar.* Doctoral dissertation, MIT.

Huck, G. and Ojeda, A., eds. (1987). *Syntax and Semantics: Discontinuous Constituency*, volume 20. Academic Press, San Diego.

Hudson, R. A. (1976). Conjunction reduction, gapping, and right-node raising. *Language* **52** (3), 535–562.

Hudson, R. A. (1984). *Word Grammar.* Blackwell, Oxford.

Iida, M. (1987). Case-assignment by nominals in Japanese. In *Working Papers in Grammatical Theory and Discourse Structure, Vol. I: Interactions of Morphology, Syntax, and Discourse* (M. Iida, S. Wechsler, and D. Zec, eds.), CSLI Lecture Notes, number 11, 93–138. CSLI Publications, Stanford University.

Ishikawa, A. (1985). *Complex Predicates and Lexical Operations in Japanese.* Doctoral dissertation, Stanford University.

Izvorski, R. (1993). On WH-movement and focus movement in Bulgarian. In *ConSole II Proceedings* (R. Eckhardt and V. van Geenhoven, eds.). Sole Publications, Amsterdam.

Jackendoff, R. S. (1977). *$\bar{X}$ Syntax: A Study of Phrase Structure.* The MIT Press, Cambridge, MA.

Jackendoff, R. S. (1983). *Semantics and Cognition.* The MIT Press, Cambridge, MA.

Jackendoff, R. S. (1990). *Semantic Structures.* The MIT Press, Cambridge, MA.

Jacobson, P. (1990). Raising as function composition. *Linguistics and Philosophy* **13**, 423–475.

Jacobson, P. (1992). Raising without movement. In *Control and Grammar* (R. K. Larson, S. Iatridou, U. Lahiri, and J. Higginbotham, eds.), 149–194. Kluwer Academic Publishers, Dordrecht.

Jaeggli, O. (1980). *On Some Phonologically Null Elements in Syntax.* Doctoral dissertation, MIT.

Janssen, T. M. V. (1997). Compositionality. In *Handbook of Logic and Language,* 419–473. The MIT Press/Elsevier, Cambridge/Amsterdam.

Jelinek, E. (1984). Empty categories, case, and configurationality. *Natural Language and Linguistic Theory* **2** (1), 39–76.

Johnson, D. E. and Postal, P. (1980). *Arc Pair Grammar.* Princeton University Press, Princeton.

Johnson, M. (1986). The LFG treatment of discontinuity and the double infinitive construction in Dutch. In *Proceedings of the Fifth West Coast Conference on Formal Linguistics* (M. Dalrymple, J. Goldberg, K. Hanson, M. Inman, C. Piñon, and S. Wechsler, eds.), 102–118. Stanford Linguistics Association, Stanford University.

Johnson, M. (1988). *Attribute-Value Logic and the Theory of Grammar.* CSLI Lecture Notes, number 16. CSLI Publications, Stanford University.

Johnson, M. (1998). *Optimality-Theoretic Lexical Functional Grammar.* URL `www.cog.brown.edu/~mj/papers/cuny98.ps.gz`. Presented to the 11th Annual CUNY Conference on Human Sentence Processing, Special Session on the Lexical Basis of Syntactic Processing: Formal and Computational Issues, Rutgers University.

Johnson, M. (1999). Type-driven semantic interpretation and feature dependencies in R-LFG. In *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach* (M. Dalrymple, ed.). The MIT Press, Cambridge, MA.

Joshi, S. (1993). *Selection of Grammatical and Logical Functions in Marathi.* Doctoral dissertation, Stanford University.

Junghanns, U. and Zybatow, G. (1997). Syntax and information structure of Russian clauses. In *Formal Approaches to Slavic Linguistics: The Cornell Meeting, 1995* (W. Browne, E. Dornisch, N. Kondrashova, and D. Zec, eds.), 289–319. Michigan Slavic Association, Ann Arbor.

Kameyama, M. (1985). *Zero Anaphora: The Case of Japanese.* Doctoral dissertation, Stanford University.

Kameyama, M. (1989). *Functional Precedence Conditions on Overt and Zero Pronominals.* Technical report, Microelectronics and Computer Technology Corporation.

Kamp, H. (1975). Two theories about adjectives. In *Formal Semantics of Natural Language* (E. L. Keenan, ed.). Cambridge University Press, Cambridge, England.

Kamp, H. (1981). A theory of truth and semantic representation. In *Formal Methods in the Study of Language* (J. Groenendijk, T. Janssen, and M. Stokhof, eds.), 277–321. Mathematical Centre, Amsterdam.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic: An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory.* Kluwer Academic Publishers, Dordrecht.

Kanerva, J. M. (1987). Morphological integrity and syntax: The evidence from Finnish possessive suffixes. *Language* **63** (3), 498–521.

Kaplan, R. M. (1975a). On process models for sentence analysis. In *Explorations in cognition* (D. A. Norman and D. E. Rumelhart, eds.), 117–135. W. H. Freeman, San Francisco.

Kaplan, R. M. (1975b). *Transient Processing Load in Relative Clauses.* Doctoral dissertation, Harvard University.

Kaplan, R. M. (1976). *Models of Comprehension Based on Augmented Transition Networks.* Presented at the ATT-MIT Convocation on Communications, March 1976.

Kaplan, R. M. (1987). Three seductions of computational psycholinguistics. In *Linguistic Theory and Computer Applications* (P. Whitelock, M. M. Wood, H. L. Somers, R. Johnson, and P. Bennett, eds.), 149–188. Academic Press, London. Also published as CCL/UMIST Report No. 86.2: Alvey/ICL Workshop on Linguistic Theory and Computer Applications: Transcripts of Presentations and Discussions. Center for Computational Linguistics, University of Manchester. Reprinted in Dalrymple et al. (1995b, pp. 337–367).

Kaplan, R. M. (1988). *Correspondences and Their Inverses.* Presented at the Titisee Workshop on Unification Formalisms: Syntax, Semantics and Implementation, Titisee, Germany.

Kaplan, R. M. (1989). The formal architecture of Lexical-Functional Grammar. In *Proceedings of ROCLING II* (C.-R. Huang and K.-J. Chen, eds.), 3–18. Reprinted in Dalrymple et al. (1995b, pp. 7–27).

Kaplan, R. M. and Bresnan, J. (1982). Lexical-Functional Grammar: A formal system for grammatical representation. In *The Mental Representation of Grammatical Relations* (J. Bresnan, ed.), 173–281. The MIT Press, Cambridge, MA. Reprinted in Dalrymple et al. (1995b, pp. 29–130).

Kaplan, R. M. and Kay, M. (1994). Regular models of phonological rule systems. *Computational Linguistics* **20** (3), 331–378.

Kaplan, R. M. and Maxwell, J. T., III (1988). Constituent coordination in Lexical-Functional Grammar. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING88),* Budapest, volume 1, 303–305. Reprinted in Dalrymple et al. (1995b, pp. 199–210).

Kaplan, R. M. and Maxwell, J. T., III (1996). *LFG Grammar Writer's Workbench.* Technical report, Xerox Palo Alto Research Center. URL `ftp://ftp.parc.xerox.com/pub/lfg/lfgmanual.ps`.

Kaplan, R. M., Maxwell, J. T., III, and Zaenen, A. (1987). Functional uncertainty. *CSLI Monthly Newsletter* Stanford University.

Kaplan, R. M. and Wedekind, J. (1993). Restriction and correspondence-based translation. In *Proceedings of the 6th Meeting of the EACL.* European Chapter of the Association for Computational Linguistics, University of Utrecht.

Kaplan, R. M. and Wedekind, J. (2000). LFG generation produces context-free languages. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING2000),* Saarbrücken.

Kaplan, R. M. and Zaenen, A. (1989). Long-distance dependencies, constituent structure, and functional uncertainty. In *Alternative Conceptions of Phrase Structure* (M. Baltin and A. Kroch, eds.), 17–42. Chicago University Press. Reprinted in Dalrymple et al. (1995b, pp. 137–165).

Karttunen, L. (1976). Discourse referents. In *Syntax and Semantics: Notes from the Linguistic Underground* (J. D. McCawley, ed.), volume 7, 363–386. Academic Press, New York.

Karttunen, L. (1977). The syntax and semantics of questions. *Linguistics and Philosophy* **1**, 3–44.

Kasper, R. (1995). *Semantics of Recursive Modification.* URL `ftp://ling.ohio-state.edu/pub/HPSG/Workshop.Tue.95/Kasper/`. Handout for HPSG Workshop Tübingen, June 1995.

Kay, M. (1979). Functional Grammar. In *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistic Society* (C. Chiarello, J. Kingston, E. E. Sweetser, J. Collins, H. Kawasaki, J. Manley-Buser, D. W. Marschak, C. O'Connor, D. Shaul, M. Tobey, H. Thompson, and K. Turner, eds.), 142–158. Berkeley Linguistics Society, University of California, Berkeley.

Kay, P. (1998). An informal sketch of a formal architecture for Construction Grammar. In *Proceedings of the Conference on Formal Grammar, HPSG and Categorial Grammar.* Saarbrücken. URL `www.icsi.berkeley.edu/~kay/cg.arch.ps`.

Kay, P. and Fillmore, C. J. (1997). *Grammatical Constructions and Linguistic Generalizations: the* What's X doing Y? *Construction.* URL `www.icsi.berkeley.edu/~kay/bcg/wxdy.ps`. MS, University of California-Berkeley.

Keenan, E. L. and Comrie, B. (1977). Noun phrase accessibility and universal grammar. *Linguistic Inquiry* **8** (1), 63–99.

Keenan, E. L. and Westerståhl, D. (1997). Generalized quantifiers in linguistics and logic. In *Handbook of Logic and Language,* 837–893. The MIT Press/Elsevier, Cambridge/Amsterdam.

Kehler, A. (1995). *Interpreting Cohesive Forms in the Context of Discourse Influence.* Doctoral dissertation, Harvard University.

Kehler, A. (2000). Discourse. In *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, by Daniel Jurafsky and James H. Martin (contributing writers: Andrew Kehler, Keith Vander Linden, and Nigel Ward), Prentice-Hall, New Jersey.

Kehler, A., Dalrymple, M., Lamping, J., and Saraswat, V. (1995). The semantics of resource-sharing in Lexical-Functional Grammar. In *Proceedings of the 7th Meeting of the EACL.* European Chapter of the Association for Computational Linguistics, University College Dublin. Reprinted in Dalrymple (1999, pp. 191–208).

Keller, B. (1993). *Feature Logics, Infinitary Descriptions, and Grammar.* Number 44 in CSLI Lecture Notes. CSLI Publications, Stanford University.

Kennedy, C. (1997). *Projecting the Adjective: The Syntax and Semantics of Gradability and Comparison.* Doctoral dissertation, University of California, Santa Cruz.

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

King, T. H. (1997). Focus domains and discourse function structure. In *On-line Proceedings of the LFG97 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/2/lfg97.html`.

Kiparsky, P. and Kiparsky, C. (1970). Fact. In *Progress in Linguistics* (M. Bierwisch and K. E. Heidolph, eds.), 143–173. Mouton, The Hague.

Klavans, J. (1982). Configuration in non-configurational languages. In *Proceedings of the West Coast Conference on Formal Linguistics*, volume 1, 292–306.

Krifka, M. (1990). Boolean and non-boolean 'and'. In *Papers from the Second Symposium on Logic and Language* (L. Kálmán and L. Pólos, eds.). Akadémiai Kiadó, Budapest.

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

Krylova, O. and Khavronina, S. (1988). *Word Order in Russian.* Russky Yazyk Publishers.

Kuhn, J. (2001a). Generation and parsing in Optimality Theoretic syntax: Issues in the formalization of OT-LFG. In *Formal and Empirical Issues in Optimality Theoretic Syntax* (P. Sells, ed.). CSLI Publications, Stanford, CA. To appear.

Kuhn, J. (2001b). Resource sensitivity in the syntax-semantics interface and the German split NP construction. In *Constraint-Based Approaches to Germanic Syntax* (T. Kiss and D. Meurers, eds.). CSLI Publications, Stanford University. To appear.

Laczkó, T. (1995). *The Syntax of Hungarian Noun Phrases – A Lexical-Functional Approach.* Peter Lang GmbH, Frankfurt am Main.

Lambek, J. (1958). The mathematics of sentence structure. *American Mathematical Monthly* **65**, 154–170.

Lamping, J. and Gupta, V. (1998). Efficient linear logic meaning assembly. In *Proceedings of COLING/ACL98: Joint Meeting of the 36th Annual Meeting of the ACL and the*

17th International Conference on Computational Linguistics, Montréal. Association for Computational Linguistics.

Landman, F. (1989). Groups. *Linguistics and Philosophy* **12**, 559–605, 723–744.

Lapata, M. (1998). Anaphoric binding in Modern Greek. In *On-line Proceedings of the LFG98 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/3/lfg98.html`.

Lapointe, S. (1980). *A Theory of Grammatical Agreement.* Doctoral dissertation, University of Massachusetts at Amherst.

Lasersohn, P. (1995). *Plurality, Conjunction and Events.* Kluwer Academic Publishers, Dordrecht.

Levin, B. (1993). *English Verb Classes and Alternations.* University of Chicago Press, Chicago.

Levin, L. S. (1982). Sluicing. In *The Mental Representation of Grammatical Relations* (J. Bresnan, ed.), 590–654. The MIT Press, Cambridge, MA.

Levin, L. S. (1986). *Operations on Lexical Forms: Unaccusative Rules in Germanic Languages.* Doctoral dissertation, MIT.

Levin, L. S. (1987). *Toward a Linking Theory of Relation-Changing Rules in LFG.* MS, University of Pittsburgh.

Li, C. N., ed. (1976). *Subject and Topic.* Academic Press, New York.

Link, G. (1983). The logical analysis of plurals and mass terms: A lattice-theoretical approach. In *Meaning, Use, and the Interpretation of Language* (R. Bauerle, C. Schwarze, and A. von Stechow, eds.). W. de Gruyter, Berlin.

Lipson, A. (1981). *A Russian Course.* Slavica, Columbus, OH.

Lødrup, H. (1999). Linking and optimality in the Norwegian presentational focus construction. *Nordic Journal of Linguistics* **22** (2), 205–229.

Maling, J. (1983). Transitive adjectives: A case of categorial reanalysis. In *Linguistic Categories: Auxiliaries and Related Puzzles, volume 1* (F. Heny, ed.). D. Reidel, Dordrecht.

Maling, J. and Zaenen, A., eds. (1990). *Syntax and Semantics: Modern Icelandic Syntax,* volume 24. Academic Press, San Diego.

Manning, C. D. (1996a). Argument structure as a locus for binding theory. In *On-line Proceedings of the LFG96 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/1/lfg1.html`.

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

Markantonatou, S. (1992). *The Syntax of the Modern Greek Noun Phrases with a Deverbal Nominal Head.* Doctoral dissertation, University of Essex.

Markantonatou, S. (1995). Modern Greek deverbal nominals: An LMT approach. *Journal of Linguistics* **31**, 267–299.

Markantonatou, S. and Sadler, L. (1995). *Linking Indirect Arguments.* Technical Report 49, University of Essex.

Matsumoto, Y. (1996). *Complex Predicates in Japanese: A Syntactic and Semantic Study of the Notion 'Word'.* Doctoral dissertation, Stanford University. *Studies in Japanese Linguistics* series, CSLI Publications/Kuroiso Publishers, Stanford/Tokyo. Revised and corrected version of 1992 Stanford University dissertation, *On the wordhood of complex predicates in Japanese.*

Matsumoto, Y. (1998). A reexamination of the cross-linguistic parameterization of causative predicates: Japanese perspectives. In *On-line Proceedings of the LFG98 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/3/lfg98.html`.

Maxwell, J. T., III and Kaplan, R. M. (1991). A method for disjunctive constraint satisfaction. In *Current Issues in Parsing Technology* (M. Tomita, ed.), 173–190. Kluwer Academic Publishers. Reprinted in Dalrymple et al. (1995b, pp. 381–401).

Maxwell, J. T., III and Kaplan, R. M. (1993). The interface between phrasal and functional constraints. *Computational Linguistics* **19** (4), 571–590.

Maxwell, J. T., III and Kaplan, R. M. (1996). An efficient parser for LFG. In *On-line Proceedings of the First LFG Conference, Rank Xerox, Grenoble, August 26–28, 1996* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/1/lfg1.html`.

Maxwell, J. T., III and Manning, C. D. (1996). A theory of non-constituent coordination based on finite-state rules. In *On-line Proceedings of the First LFG Conference, Rank Xerox, Grenoble, August 26–28, 1996* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/1/lfg1.html`. Also presented at the Symposium on Coordination, 1996 European Summer School on Logic, Language, and Information, Prague.

May, R. (1977). *The Grammar of Quantification.* Doctoral dissertation, MIT.

McCawley, J. D. (1968). Concerning the base component of a transformational grammar. *Foundations of Language* **4**, 243–269. Reprinted in McCawley (1973).

McCawley, J. D. (1973). *Grammar and Meaning: Papers on Syntactic and Semantic Topics.* Taishukan, Tokyo.

McCawley, J. D. (1982). Parentheticals and discontinuous constituent structure. *Linguistic Inquiry* **13** (1), 91–106.

McCawley, J. D. (1988). *The Syntactic Phenomena of English.* The University of Chicago Press, Chicago.

McCloskey, J. (1979). *Transformational Syntax and Model Theoretic Semantics: A Case Study in Modern Irish.* D. Reidel, Dordrecht.

McConnell-Ginet, S. (1979). On the deep (and surface) adjective 'good'. In *Grammatical Studies: Semantics and Syntax* (L. R. Waugh and F. van Coetsem, eds.). E.J. Brill, Leiden.

McGregor, R. S. (1972). *Outline of Hindi Grammar.* Oxford University Press, Oxford, England and Delhi, India.

Mel'čuk, I. A. (1988). *Dependency Syntax.* State University of New York Press, Albany.

Milward, D. (1994). Non-constituent coordination: Theory and practice. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING94)*, Kyoto.

Mohanan, K. P. (1982). Grammatical relations and clause structure in Malayalam. In *The Mental Representation of Grammatical Relations* (J. Bresnan, ed.), 504–589. The MIT Press, Cambridge, MA.

Mohanan, K. P. (1983). Functional and anaphoric control. *Linguistic Inquiry* **14** (4), 641–674.

Mohanan, T. (1994). *Arguments in Hindi.* Doctoral dissertation, Stanford University. *Dissertations in Linguistics* series, CSLI Publications, Stanford University. Reprinted version of 1990 Stanford University dissertation.

Mohanan, T. (1995). Wordhood and lexicality: Noun incorporation in Hindi. *Natural Language and Linguistic Theory* **13** (1), 75–134.

Moltmann, F. (1992). *Coordination and Comparatives.* Doctoral dissertation, MIT.

Montague, R. (1970). Universal grammar. *Theoria* **36**, 373–398.

Montague, R. (1974a). English as a formal language. In *Formal Philosophy* (R. Thomason, ed.). Yale University Press, New Haven.

Montague, R. (1974b). The proper treatment of quantification in ordinary English. In *Formal Philosophy* (R. Thomason, ed.). Yale University Press, New Haven.

Moortgat, M. (1988). *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus.* Doctoral dissertation, University of Amsterdam, Amsterdam.

Moortgat, M. (1996). Generalized quantifiers and discontinuous type constructors. In *Discontinuous Constituency* (H. Bunt and A. van Horck, eds.), 181–207. Mouton de Gruyter, Berlin.

Moravcsik, E. A. (1978). Agreement. In *Universals of Human Language: Syntax* (J. H. Greenberg, C. A. Ferguson, and E. A. Moravcsik, eds.), volume 4, 331–374. Stanford University Press, Stanford, CA.

Morimoto, Y. (1999). An optimality account of argument reversal. In *On-line Proceedings of the LFG99 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/4/lfg99.html`.

Morimoto, Y. (2000). *Discourse Configurationality in Bantu Morphosyntax.* Doctoral dissertation, Stanford University.

Morrill, G. V. (1994). *Type Logical Grammar: Categorial Logic of Signs.* Kluwer Academic Publishers, Dordrecht.

Muskens, R. (1995). Order-independence and underspecification. In *Ellipsis, Underspecification, Events and More in Dynamic Semantics* (J. Groenendijk, ed.). The DYANA-2 Project Administrator, ILLC, University of Amsterdam. Dyana-2 Deliverable R2.2.C.

Neidle, C. (1982). Case agreement in Russian. In *The Mental Representation of Grammatical Relations* (J. Bresnan, ed.), 391–426. The MIT Press, Cambridge, MA.

Newman, P. S. (1996). Representing 2P clitic placement in LFG. In *On-line Proceedings of the First LFG Conference, Rank Xerox, Grenoble, August 26–28, 1996* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/1/lfg1.html`. Presented at the LFG Workshop, Grenoble, France, August 1996.

Nichols, J. (1986). Head-marking and dependent-marking grammar. *Language* **62** (1), 56–119.

Niño, M.-E. (1995). *Split Inflection and the Architecture of Grammar: The Case of Finnish Verbal Morphology.* URL `www-linguistics.stanford.edu/Linguistics/papers/spinfl.html.` MS, Stanford University.

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

Nordlinger, R. and Bresnan, J. (1996). Nonconfigurational tense in Wambaya. In *Online Proceedings of the LFG96 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/1/lfg1.html.`

O'Connor, M. C. (1987). *Topics in Northern Pomo Grammar.* Doctoral dissertation, University of California, Berkeley. Reprinted by Garland Press, New York, 1992.

Oehrle, D. (1999). LFG as labeled deduction. In *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach* (M. Dalrymple, ed.). The MIT Press, Cambridge, MA.

Oehrle, R. T., Bach, E., and Wheeler, D., eds. (1988). *Categorial Grammars and Natural Language Semantics.* D. Reidel, Dordrecht.

Ostler, N. (1979). *Case-Linking: A Theory of Case and Verb Diathesis Applied to Classical Sanskrit.* Doctoral dissertation, MIT.

Partee, B. (1970). Negation, conjunction, and quantifiers: Syntax vs. semantics. *Foundations of Language* **6** (2), 153–165.

Partee, B. and Rooth, M. (1983). Generalized conjunction and type ambiguity. In*Meaning, Use, and Interpretation of Language* (R. Bauerle, C. Schwarze, and A. von Stechow, eds.), 361–383. W. de Gruyter, Berlin.

Partee, B., ter Meulen, A., and Wall, R. E. (1993). *Mathematical methods in linguistics.* Kluwer Academic Publishers, Dordrecht. Corrected second printing of the first edition.

Partee, B. H. and Borschev, V. (1998). Integrating lexical and formal semantics: Genitives, relational nouns, and type-shifting. In *Proceedings of the Second Tbilisi Symposium on Language, Logic, and Computation* (R. Cooper and T. Gamkrelidze, eds.), 229–241. Center on Language, Logic, Speech, Tbilisi State University, Tbilisi, URL `www-unix.oit.umass.edu/~partee/docs/tbilisi17.pdf.`

Pereira, F. C. N. (1990). Categorial semantics and scoping. *Computational Linguistics* **16** (1), 1–10.

Pereira, F. C. N. and Warren, D. H. D. (1983). Parsing as deduction. In *Proceedings of the 21st Annual Meeting of the ACL,* Cambridge, MA, 137–144. Association for Computational Linguistics.

Perlmutter, D. and Postal, P. M. (1983a). Some proposed laws of basic clause structure. In *Studies in Relational Grammar 1.* The University of Chicago Press, Chicago.

Perlmutter, D. and Postal, P. M. (1983b). Toward a universal characterization of passivization. In *Studies in Relational Grammar 1.* The University of Chicago Press, Chicago.

Peterson, P. G. (1982). *Conjunction in LFG.* MS, Department of Linguistics, MIT and University of Newcastle, New South Wales, Australia.

Pica, P. (1987). On the nature of the reflexivization cycle. In *Proceedings of the Seventeenth Annual Meeting of the North Eastern Linguistic Society* (J. McDonough and B. Plunkett, eds.), volume 17, 483–500.

Pinker, S. (1982). A theory of the acquisition of Lexical-Interpretive Grammars. In *The Mental Representation of Grammatical Relations* (J. Bresnan, ed.), 655–726. The MIT Press, Cambridge, MA.

Pinker, S. (1989). *Learnability and Cognition: The Acquisition of Argument Structure.* The MIT Press, Cambridge, MA.

Pollard, C. and Sag, I. A. (1987). *Information-Based Syntax and Semantics, Volume I.* CSLI Lecture Notes, number 13. CSLI Publications, Stanford University.

Pollard, C. and Sag, I. A. (1994). *Head-Driven Phrase Structure Grammar.* The University of Chicago Press, Chicago.

Pollock, J. (1989). Verb movement, Universal Grammar, and the structure of IP. *Linguistic Inquiry* **20** (3).

Poser, W. J. (1992). Blocking of phrasal constructions by lexical items. In *Lexical Matters*, CSLI Lecture Notes, number 24, 111–130. CSLI Publications, Stanford, CA.

Postal, P. M. (1972). On coreferential complement subject deletion. *Linguistic Inquiry* **1** (4).

Postal, P. M. (1974). *On Raising.* The MIT Press, Cambridge, MA.

Postal, P. M. (1998). *Three Investigations of Extraction.* The MIT Press, Cambridge, MA.

Prince, A. S. and Smolensky, P. (1993). *Optimality Theory: Constraint Interaction in Generative Grammar.* Technical Report 2, Rutgers University Center for Cognitive Science, New Brunswick, NJ.

Pulleyblank, D. (1986). Clitics in Yoruba. In *Syntax and Semantics: The Syntax of Pronominal Clitics* (H. Borer, ed.), volume 19, 43–64. Academic Press, New York.

Quine, W. V. O. (1953). *Reference and Modality.* Harvard University Press, Cambridge.

Quirk, R., Greenbaum, S., Leech, G., and Svartvik, J. (1985). *A Comprehensive Grammar of the English Language.* Longman, New York.

Radford, A. (1981). *Transformational Syntax: A Student's Guide to Chomsky's Extended Standard Theory.* Cambridge University Press, Cambridge, England.

Reape, M. (1994). Domain union and word order variation in German. In *German in Head-Driven Phrase Structure Grammar* (J. Nerbonne, K. Netter, and C. Pollard, eds.), 151 – 197. CSLI Publications, Stanford University.

Reinhart, T. (1976). *The Syntactic Domain of Anaphora.* Doctoral dissertation, MIT.

Reyle, U. (1988). Compositional semantics for LFG. In *Natural Language Parsing and Linguistic Theories* (U. Reyle and C. Rohrer, eds.). D. Reidel, Dordrecht.

Reyle, U. (1993). Dealing with ambiguities by underspecification: Construction, representation, and deduction. *Journal of Semantics* **10**, 123–179.

Rhodes, R. (1990). Ojibwa secondary objects. In *Grammatical Relations: A Cross-Theoretical Perspective* (K. Dziwirek, P. Farrell, and E. Mejías-Bikandi, eds.), 401–414. CSLI Publications, Stanford University.

Rizzi, L. (1990). *Relativized Minimality.* The MIT Press, Cambridge, MA.

Robins, R. H. (1967). *A Short History of Linguistics.* Indiana University Press, Bloomington.

Rögnvaldsson, E. (1984). Af Lýsingaraorðsviðurlöm. *Íslenskt mál* **6**, 58–80.

Rosen, C. (1990). Rethinking Southern Tiwa: The geometry of a triple-agreement language. *Language* **66** (4), 669–713.

Rosén, V. (1996). The LFG architecture and "verbless" syntactic constructions. In *On-line Proceedings of the First LFG Conference, Rank Xerox, Grenoble, August 26–28, 1996* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/1/lfg1.html`.

Ross, J. R. (1967). *Constraints on Variables in Syntax.* Doctoral dissertation, MIT.

Rounds, W. C. (1988). *Set Values for Unification-Based Grammar Formalisms and Logic Programming.* Technical Report CSLI-88-129, Center for the Study of Language and Information, Stanford University.

Rounds, W. C. (1997). Feature logics. In *Handbook of Logic and Language*, 475–533. The MIT Press/Elsevier, Cambridge/Amsterdam.

Rudin, C. (1985). *Aspects of Bulgarian Syntax: Complementizers and WH Constructions.* Slavica, Columbus, OH.

Sadler, L. (1997). Clitics and the structure-function mapping. In *On-line Proceedings of the LFG97 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/2/lfg97.html`.

Sadler, L. (1998). English auxiliaries as tense inflections. In *Special Issue of Essex Research Reports Produced on the Occasion of the Retirement of Keith Brown.*

Sadler, L. (1999). Non-distributive features and coordination in Welsh. In *On-line Proceedings of the LFG99 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/4/lfg99.html`.

Sadler, L. and Arnold, D. (1994). Prenominal adjectives and the phrasal/lexical distinction. *Journal of Linguistics* **30**, 187–226.

Sadler, L. and Spencer, A. (2000). Workshop on morphology and LFG. In *On-line Proceedings of the LFG2000 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/5/lfg00morphwrk.html`.

Sadock, J. M. (1991). *Autolexical Syntax: A Theory of Parallel Grammatical Representations.* The University of Chicago Press, Chicago.

Sag, I. A. (1986). *Grammatical Hierarchy and Linear Precedence.* Technical report, Center for the Study of Language and Information, Stanford University. Reprinted in Huck and Ojeda (1987, pp. 303–340).

Sag, I. A. (1998). *Without a Trace.* MS, Stanford University.

Sag, I. A. and Fodor, J. D. (1994). Extraction without traces. In *Proceedings of the Thirteenth West Coast Conference on Formal Linguistics.* CSLI Publications, Stanford University.

Sag, I. A., Gazdar, G., Wasow, T., and Weisler, S. (1985). Coordination and how to distinguish categories. *Natural Language and Linguistic Theory* **3** (2), 117–171.

Sag, I. A. and Pollard, C. (1991). An integrated theory of complement control. *Language* **67** (1), 63–113.

Saiki, M. (1985). On the coordination of gapped constituents in Japanese. In *Papers from the Twenty-First Regional Meeting of the Chicago Linguistic Society* (W. H. Eilfort, P. D. Kroeber, and K. L. Peterson, eds.), 371–387. Chicago Linguistic Society, University of Chicago.

Saiki, M. (1987). *On the Manifestations of Grammatical Functions in the Syntax of Japanese Nominals.* Doctoral dissertation, Stanford University.

Saiki, M. (1991). The tough construction in Japanese. *Studies in Humanities* **28** (2), 91–115.

Saraswat, V. A. (1999). LFG as concurrent constraint programming. In *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach* (M. Dalrymple, ed.). The MIT Press, Cambridge, MA.

Scedrov, A. (1993). A brief guide to linear logic. In *Current Trends in Theoretical Computer Science* (G. Rozenberg and A. Salomaa, eds.). World Scientific Publishing Co. Revised and expanded version of an article originally appearing in *Bulletin of the European Association for Theoretical Computer Science* 41, 1990.

Schachter, P. (1976). Topic, pronoun, and grammatical agreement. In *Subject and Topic* (C. Li, ed.). Academic Press, New York.

Schwarze, C. (1997). Lexical-functional morphology and the structure of the lexicon. In *Boundaries of Morphology and Syntax* (L. Mereu, ed.), Current Issues in Linguistic Theory. John Benjamins, Rome.

Schwarzschild, R. (1996). *Pluralities.* Kluwer Academic Publishers, Dordrecht.

Seiter, W. J. (1983). Subject–direct object raising in Niuean. In *Studies in Relational Grammar 1* (D. M. Perlmutter, ed.), 317–359. The University of Chicago Press, Chicago.

Sells, P. (1988). Thematic and grammatical hierarchies: Albanian reflexivization. In *Proceedings of the Seventh West Coast Conference on Formal Linguistics* (H. Borer, ed.), 293–303. Stanford Linguistics Association, Stanford University.

Sells, P. (1994). Subphrasal syntax in Korean. *Language Research* **30** (2), 351–386.

Sells, P. (1995). Raising and the order of clausal constituents in the Philippine languages. In *Proceedings of the 2nd Annual Meeting of the Austronesian Formal Linguistics Association.* Kluwer Academic Publishers.

Sells, P. (1998). Scandinavian clause structure and object shift. In *On-line Proceedings of the LFG98 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/3/lfg98.html`.

Sgall, P., Hajičová, E., and Panevová, J. (1986). *The Meaning of the Sentence in its Semantic and Pragmatic Aspects.* D. Reidel, Dordrecht. Edited by Jacob Mey.

Shemtov, H. (1997). *Ambiguity Management in Natural Language Generation.* Doctoral dissertation, Stanford University.

Shieber, S. M. (1986). *An Introduction to Unification-Based Approaches to Grammar.* CSLI Lecture Notes, number 4. CSLI Publications, Stanford University.

Siegel, M. E. A. (1976). *Capturing the Adjective.* Doctoral dissertation, University of Massachusetts at Amherst.

Siewierska, A. (1999). Reduced pronominals and argument prominence. In *On-line Proceedings of the LFG99 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/4/lfg99.html`.

Simpson, J. (1983). Discontinuous verbs and the interaction of morphology and syntax. In *Proceedings of the Second West Coast Conference on Formal Linguistics* (M. Barlow, D. P. Flickinger, and M. T. Wescoat, eds.). Stanford Linguistics Association, Stanford University.

Simpson, J. (1991). *Warlpiri Morpho-Syntax: A Lexicalist Approach.* Kluwer Academic Publishers, Dordrecht.

Simpson, J. and Bresnan, J. (1983). Control and obviation in Warlpiri. *Natural Language and Linguistic Theory* **1** (1), 49–64.

Speas, M. (1990). *Phrase Structure in Natural Language.* Kluwer Academic Publishers, Dordrecht.

Spencer, A. (2000). *The Word and Paradigm model of morphosyntax.* URL `privatewww.essex.ac.uk/~spena/papers/wp_morphosyntax.ps`. Written draft of a paper read at the University of Manchester Postgraduate Linguistics Conference, March 2000.

Steedman, M. J. (1996). *Surface Structure and Interpretation.* The MIT Press, Cambridge, MA.

Stirling, L. (1988). *Switch-Reference and Logophoricity in Discourse Representation Theory.* Doctoral dissertation, University of Edinburgh.

Stowell, T. (1981). *The Origins of Phrase Structure.* Doctoral dissertation, MIT.

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

Strand, K. (1992). *Indeksering av nomenfraser i et textforstående system [Indexing of nominal phrases in a text understanding system].* Master's thesis, University of Oslo.

Sung, W. (1996). *Etude des pronoms coréens.* Doctoral dissertation, Université Paris 7.

Tang, C.-C. J. (1989). Chinese reflexives. *Natural Language and Linguistic Theory* **7** (1), 93–121.

Tesnière, L. (1959). *Élements de syntaxe structurale.* Klinksieck, Paris.

Thráinsson, H. (1986). On auxiliaries, AUX and VP's in Icelandic. In *Topics in Scandinavian Syntax* (K. K. Christiansen and L. Hellan, eds.). D. Reidel, Dordrecht.

Toivonen, I. (2000). The morphosyntax of finnish possessives. *Natural Language and Linguistic Theory* **18** (3), 579–609.

Toivonen, I. (2001). *The Phrase Structure of Non-projecting Words.* Doctoral dissertation, Stanford University.

Vallduví, E. (1992). *The Informational Component.* Garland, New York.

van Genabith, J. and Crouch, R. (1999a). Dynamic and underspecified semantics for LFG. In *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach* (M. Dalrymple, ed.). The MIT Press, Cambridge, MA.

van Genabith, J. and Crouch, R. (1999b). How to glue a donkey to an f-structure: Porting a dynamic meaning representation into LFG's linear logic based glue-language semantics. In *Computing Meaning* (H. Bunt and R. Muskens, eds.), volume 1, 129–148. Kluwer Academic Publishers.

Van Valin, R. D. (1987). The role of government in the grammar of head-marking languages. *International Journal of American Linguistics* **53**, 371–397.

Vilkuna, M. (1995). Discourse configurationality in Finnish. In *Discourse Configurational Languages* (K. É. Kiss, ed.). Oxford University Press, Oxford.

Vincent, N. (2001). Competition and correspondence in syntactic change: Null arguments in Latin and Romance. In *Diachronic Syntax: Models and Mechanisms* (S. Pintzuk, G. Tsoulas, and A. Warner, eds.). Oxford University Press, Oxford, URL `www-lfg.stanford.edu/lfg/archive/archive.html`.

Vincent, N. and Börjars, K. (2000). Feature resolution and the content of features. In *Online Proceedings of the LFG2000 Conference* (M. Butt and T. H. King, eds.). URL `csli-publications.stanford.edu/LFG/5/lfg00.html`.

Visser, F. T. (1963–1973). *An Historical Syntax of the English Language.* F.J. Brill, Leiden.

Wada, H. and Asher, N. (1986). BUILDRS: An implementation of DR theory and LFG. In *Proceedings of the 11th International Conference on Computational Linguistics (COLING86),* Bonn, 540–545.

Ward, G. L. (1990). Discourse functions of VP preposing. *Language* **66** (4), 742–763.

Wasow, T. (1979). *Anaphora in Generative Grammar.* E. Story, Ghent.

Way, A. (1999). A hybrid architecture for robust MT using LFG-DOP. *Journal of Experimental and Theoretical Artificial Intelligence* **11**. Special Issue on Memory-Based Language Processing.

Way, A. (2001). *LFG-DOP: A Hybrid Architecture for Robust MT.* Doctoral dissertation, University of Essex. In preparation.

Webelhuth, G. (1992). *Principles and Parameters of Syntactic Saturation.* Oxford University Press, Oxford.

Webelhuth, G. (1995). *Government and Binding Theory and the Minimalist Program.* Oxford University Press, Oxford.

Wedekind, J. (1995). Some remarks on the decidability of the generation problem in LFG- and PATR-style unification grammars. In *Proceedings of the 7th Meeting of the EACL.* European Chapter of the Association for Computational Linguistics, Dublin.

Wedekind, J. (1999). Semantic-driven generation with LFG- and PATR-style grammars. *Computational Linguistics* **25** (2), 277–281.

Wedekind, J. and Kaplan, R. M. (1993). Type-driven semantic interpretation of f-structures. In *Proceedings of the 6th Meeting of the EACL.* European Chapter of the Association for Computational Linguistics, University of Utrecht.

Wedekind, J. and Kaplan, R. M. (1996). Ambiguity-preserving generation with LFG- and PATR-style grammars. *Computational Linguistics* **22** (4), 555–568.

Williams, E. (1978). Across-the-board rule application. *Linguistic Inquiry* **9** (1), 31–44.

Williams, E. (1983). Against small clauses. *Linguistic Inquiry* **14** (2), 287–308.

Williams, E. (1992). Adjunct control. In *Control and Grammar* (R. K. Larson, S. Iatridou, U. Lahiri, and J. Higginbotham, eds.), 297–322. Kluwer Academic Publishers, Dordrecht.

Winter, Y. (1998). *Flexible Boolean Semantics.* Doctoral dissertation, Utrecht University.

Woods, W. A. (1970). Transition network grammars for natural language analysis. *Communications of the ACM* **13** (10), 591–606.

Yadroff, M. (1992). *The Syntactic Properties of Adjunction (Scrambling in Russian).* MS, Indiana University.

Yamamoto, K. (1996). The *tough* constructions in English and Japanese: A Lexical Functional Grammar approach. In *Tough Constructions in English and Japanese* (A. Ikeya, ed.), 43–61. Kuroiso, Tokyo.

Zaenen, A. (1980). *Extraction Rules in Icelandic.* Doctoral dissertation, Harvard University. Reprinted by Garland Press, New York, 1985.

Zaenen, A., ed. (1982). *Subjects and Other Subjects: Proceedings of the Harvard Conference on the Representation of Grammatical Relations.* Indiana University Linguistics Club, Bloomington.

Zaenen, A. (1983). On syntactic binding. *Linguistic Inquiry* **14** (3), 469–504.

Zaenen, A. (1993). Unaccusativity in Dutch: An integrated approach. In *Semantics and the Lexicon* (J. Pustejovsky, ed.). Kluwer Academic Publishers, Dordrecht.

Zaenen, A. and Engdahl, E. (1994). Descriptive and theoretical syntax in the lexicon. In *Computational Approaches to the Lexicon* (B. T. S. Atkins and A. Zampolli, eds.), 181–212. Oxford University Press, Oxford.

Zaenen, A. and Kaplan, R. M. (1995). Formal devices for linguistic generalizations: West Germanic word order in LFG. In *Linguistics and Computation*, CSLI Lecture Notes, number 52. CSLI Publications, Stanford University. Reprinted in Dalrymple et al. (1995b, pp. 215–239).

Zaenen, A. and Maling, J. (1983). Passive and oblique case. In *Papers in Lexical-Functional Grammar* (L. S. Levin, M. Rappaport, and A. Zaenen, eds.), 159–191. Indiana University Linguistics Club, Bloomington.

Zaenen, A. and Maling, J. (1990). Unaccusative, passive, and quirky case. In *Syntax and Semantics: Modern Icelandic Syntax* (J. Maling and A. Zaenen, eds.), volume 24, 187–234. Academic Press, San Diego.

Zaenen, A., Maling, J., and Thráinsson, H. (1985). Case and grammatical functions: The Icelandic passive. *Natural Language and Linguistic Theory* **3** (4), 441–483. Reprinted in Maling and Zaenen (1990, pp. 95–164).

Zec, D. (1987). On obligatory control in clausal complements. In *Working Papers in Grammatical Theory and Discourse Structure* (M. Iida, S. Wechsler, and D. Zec, eds.), CSLI Lecture Notes, number 11, 139–168. CSLI Publications, Stanford University.

Zec, D. and Inkelas, S. (1990). Prosodically constrained syntax. In *The Phonology-Syntax Connection* (S. Inkelas and D. Zec, eds.). CSLI Publications, Stanford University.

Zwicky, A. M. (1985). Clitics and particles. *Language* **61** (2), 283–305.

Zwicky, A. M. (1990). Syntactic representations and phonological shapes. In *The Phonology-Syntax Connection* (S. Inkelas and D. Zec, eds.), 379–397. CSLI Publications, Stanford University.

Zybatow, G. and Mehlhorn, G. (2000). Experimental evidence for focus structure in Russian. In *Formal Approaches to Slavic Linguistics: The Philadelphia Meeting 1999* (T. H. King and I. Sekerina, eds.). Michigan Slavic Publications, Ann Arbor.

AUTHOR INDEX

LANGUAGE INDEX

In most cases, language families are listed according to the classifications given in the Ethnologue database (Grimes 1999). Information on where languages are spoken is often only approximate; see the Ethnologue database for full details.

SUBJECT INDEX