Institute for Research in Cognitive Science

The Computational Analysis of the Syntax and Interpretation of "Free" Word Order in Turkish
Ph.D. Dissertation

Beryl Hoffman

University of Pennsylvania
3401 Walnut Street, Suite 400C
Philadelphia, PA 19104-6228
June 1995

Site of the NSF Science and Technology Center for Research in Cognitive Science

IRCS Report 95-17

THE COMPUTATIONAL ANALYSIS OF THE SYNTAX AND INTERPRETATION OF "FREE" WORD ORDER IN TURKISH

Beryl Hoffman

A DISSERTATION
in
Computer and Information Science

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

1995

Mark Steedman, Supervisor of Dissertation

Mark Steedman, Graduate Group Chair

ACKNOWLEDGEMENTS

I would like to thank my advisor Mark Steedman for his continual support, interest, and advice. His insights greatly influenced and improved my research. I owe much to Ellen Prince who shaped my approach to linguistics starting from my first year at Penn. Her insightful comments stimulated my research and my interest in language. I greatly appreciate the constructive criticism, comments, and suggestions provided by my other committee members: Aravind Joshi, Mark Johnson, and Mitch Marcus. I would also like to thank Annette Herskovits for introducing me to computational linguistics and for her friendship over the years.

My work was greatly influenced by Owen Rambow and Ümit Turan; I have enjoyed their friendship as well as stimulating discussions about the intersecting areas of our research. Ümit Turan and my family provided native speaker judgements other than my own and some of the naturally occurring data used in this dissertation. I would also like to thank Libby Levison, Scott Prevost, my friends and colleagues in Mark's group and Ellen's seminar for patiently listening to my preliminary research results and providing feedback, support, and friendship.

I owe much to the University, the Institute of Research in Cognitive Science (IRCS), the department, and the faculty for providing financial support during my years at graduate school and for providing a stimulating environment for interdisciplinary research.

I also thank my ex-roommates, Suneeta Ramaswami, Sabina Sawhney, Sunil Shende, and Bindi; my e-mail buddies, Yvonne Allison, Cindy Lee, Miriam Butt, Rebecca Winer; and all of my friends in Philly for providing the much needed support, friendship, and fun during grad school.

And finally, I thank my parents, Ayfer and Eugene Hoffman, and my sister, Gül, for making me bilingual and for patiently putting up with my bad moods during grad school. Their emotional support, interest, and encouragement have kept me going.

ABSTRACT

THE COMPUTATIONAL ANALYSIS OF THE SYNTAX AND INTERPRETATION OF "FREE" WORD ORDER IN TURKISH

Author: Beryl Hoffman

Supervisor: Mark Steedman

In this dissertation, I examine a language with "free" word order, specifically Turkish, in order to develop a formalism that can capture the syntax and the context-dependent interpretation of "free" word order within a computational framework. In "free" word order languages, word order is used to convey distinctions in meaning that are not captured by traditional truth-conditional semantics. The word order indicates the "information structure", e.g. what is the "topic" and the "focus" of the sentence. The context-appropriate use of "free" word order is of considerable importance in developing practical applications in natural language interpretation, generation, and machine translation.

I develop a formalism called Multiset-CCG, an extension of Combinatory Categorial Grammars, CCGs, (Ades/Steedman 1982, Steedman 1985), and demonstrate its advantages in an implementation of a data-base query system that interprets Turkish questions and generates answers with contextually appropriate word orders. Multiset-CCG is a context-sensitive and polynomially parsable grammar that captures the formal and descriptive properties of "free" word order and restrictions on word order in simple and complex sentences (with discontinuous constituents and long distance dependencies). Multiset-CCG captures the context-dependent meaning of word order in Turkish by compositionally deriving the predicate-argument structure and the information structure of a sentence in parallel. The advantages of using such a formalism are that it is computationally attractive and that it provides a compositional and flexible surface structure that allows syntactic constituents to correspond to information structure constituents. A formalism that integrates information structure and syntax such as Multiset-CCG is essential to the computational tasks of interpreting and generating sentences with contextually appropriate word orders in "free" word order languages.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Research Goals

In this dissertation, I provide an analysis of the syntax and interpretation of "free" word order in Turkish within a computational framework. Many languages (e.g. Czech, Finnish, German, Hindi, Hungarian, Japanese, Korean, Polish, Russian, Turkish, Urdu, Warlpiri) have relatively free word order when compared with English. Word order in Turkish is so free that in fact the simple transitive sentence "Chris saw Pat" can be translated to Turkish in six different word orders (i.e. all the permutations of the three word sentence). Although all six permutations have the same traditional propositional interpretation, *see(Chris,Pat)*, they are not used in the same contexts.

In "free" word order languages, word order is used to communicate distinctions in meaning that are not captured by traditional truth-conditional semantics. The word order serves to structure the information conveyed to the hearer, e.g. by indicating what is the *topic* and the *focus* of the sentence (as will be defined in this dissertation) in the *information structure* of the sentence. The information structure is an important aspect of meaning in all languages. In fixed word order languages such as English, the information structure is primarily expressed through intonation and stress, while in "free" word order languages such as Turkish, it is primarily expressed through word order variation. Humans show syntactic as well as pragmatic competence in language, i.e. they use a syntactic or prosodic form appropriate to the context. Thus, this level of context-dependent meaning must be incorporated with theories of syntax in order to model the competence of speakers in all languages.

The goal of this dissertation is to examine a language with "free" word order, specifically Turkish, in order to develop an integrated grammar for syntactic and pragmatic competence within a

computational framework. The formalism I develop is called Multiset-CCG, an extension of Combinatory Categorial Grammars, CCGs, (Ades and Steedman, 1982; Steedman, 1985; Steedman, 1991). The advantages of using a combinatory categorial formalism are that it is computationally attractive, and it provides a compositional and flexible surface structure which allows syntactic constituents to easily correspond with information structure units. Multiset-CCG captures the context-appropriate use of word order in Turkish by compositionally deriving the predicate-argument structure and the information structure of a sentence in parallel.

As computational linguists, we must strive to capture the context-dependent use of language in order to model human linguistic ability as well as to develop practical computer applications in natural language processing. I investigate the formal properties (e.g. the weak generative capacity and parsability) of Multiset-CCG, and demonstrate the advantages of using Multiset CCG in an implementation of a data-base query system, outlined in Figure 1.1. This system uses Multiset-CCG to interpret Turkish questions and generate answers with contextually appropriate word orders. The implementation has helped me to test and further develop Multiset-CCG within a well-defined domain, as well as to display the advantages of using such a formalism in a computational application.

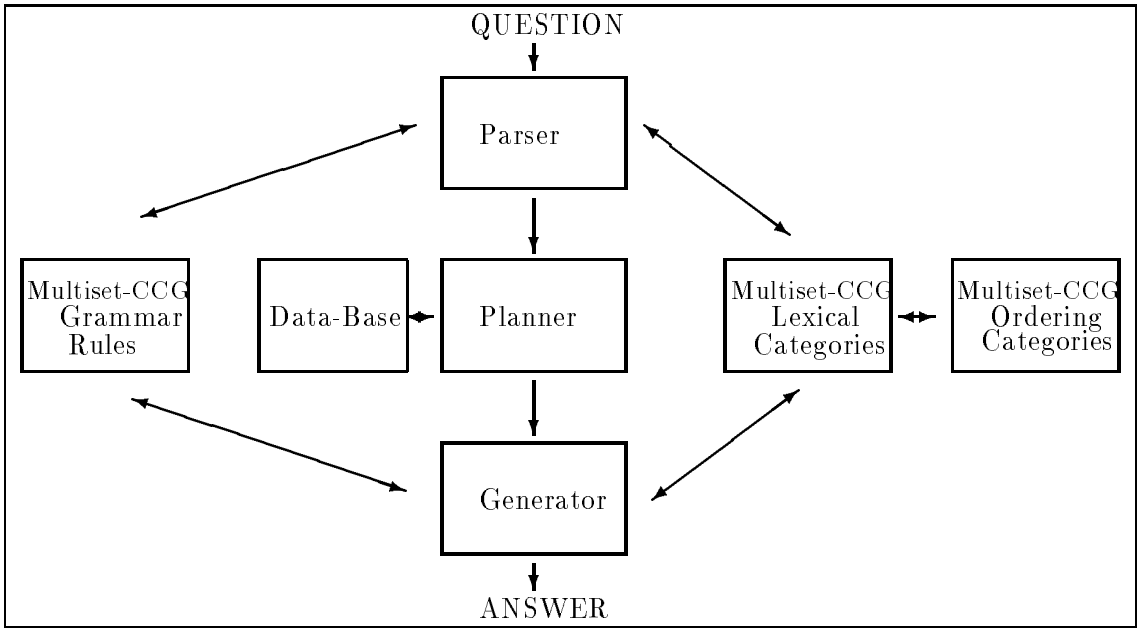


Figure 1.1: The Personal Assistant Generation System

1.2 Outline of Dissertation

In remaining sections of this introductory chapter, I present background information about Turkish morphology in Section 3 and Combinatory Categorial Grammars (CCGs) in Section 4.

In Chapter 2, I present the Turkish data on "free" word order that motivates this thesis and outline what is required of a competence grammar for "free" word order languages. I explain why I have chosen to develop a categorial formalism based on CCGs and why CCG has to be extended to handle the formal and descriptive properties of "free" word order languages. The argument is based on the unbounded nature of long distance scrambling, i.e. the extraction and permutation of an unbounded number of elements over unbounded distances.

In Chapter 3, I present a syntactic competence grammar for a fragment of Turkish that captures the basic syntactic and semantic relationships between predicates and their arguments or adjuncts while allowing "free" word order. This is represented as the boxes for the Multiset-CCG Grammar Rules and Lexical Categories in Figure 1.1. The syntactic Multiset-CCG uniformly handles free word order among arguments and adjuncts in simple and complex clauses, as well as word order variation across clause boundaries, i.e. long distance scrambling. In addition, it can capture the appropriate syntactic restrictions on word order such as island constraints and head-finality.

In Chapter 4, I investigate the formal aspects of Multiset-CCG and show that it is a computationally attractive formalism. I investigate the weak generative capacity of different versions of Multiset-CCG and show that they can generate some, but not all, context-sensitive languages. I present a polynomial-time parsing algorithm for Multiset-CCG. In addition, I compare Multiset-CCG to other computational formalisms that have been proposed to capture "free" word order syntax.

In Chapter 5, I investigate the discourse functions of word order in Turkish. I provide a detailed analysis of a corpus of naturally occurring discourses to contrast the characteristics of different sentence positions in Turkish. I develop the information structure representation which will be used in Multiset-CCG. In addition, I argue that information structure is distinct from the predicate-argument structure of a sentence in that adjuncts and elements from embedded clauses can play a role in the information structure of the matrix sentence without being an argument in the predicate-argument structure of the matrix sentence.

In Chapter 6, I integrate a grammar component that captures the information structure of Turkish sentences with Multiset-CCG. This is represented as the box for Multiset-CCG Ordering Categories in Figure 1.1. Multiset-CCG provides a flexible surface structure which directly reflects the syntactic as well as the informational/pragmatic constituency. Thus, the word order of a sentence is influenced by its information structure. This is demonstrated in the implementation,

Figure 1.1, also described in Chapter 6, which uses Multiset-CCG to generate Turkish sentences with word orders appropriate to the context of data-base queries.

In summary, this dissertation provides:

- **Linguistic Analysis** of the syntactic and pragmatic features of Turkish word order.
- **Multiset CCG**, a formalism that integrates syntactic and pragmatic information.
- **Formal Analysis** of the problem and the developed formalism.
- **Computational Application** of the developed formalism.

1.3 Background Assumptions

1.3.1 Turkish Morphology

Turkish, a member of the Altaic family of languages, is an agglutinative language. Verb stems can combine with morphemes for tense, voice, mood, number, person, passive, causative, reciprocal, question, etc. Subject-verb agreement is marked by person and number morphemes on the verb. The third person and singular morphemes are phonologically unmarked. The plural is indicated by the morpheme "ler" or "lar" that is affixed to noun phrases and optionally to verbs:

(1) Kedi-ler uyu-yor-(lar).
Cat-Pl sleep-Prog-(Pl).
"The cats are sleeping."

Turkish nouns can occur with six different cases: nominative, accusative, genitive, dative, ablative, and locative cases. Since Turkish morphology exhibits vowel harmony, each case is associated with a morpheme and its vowel-harmony variants, i.e. allomorphs. The nominative case morpheme is phonologically null. The dative, ablative, and locative cases have 2 allomorphs which depend on whether the closest vowel in the noun stem is a front or back vowel. The four allomorphs of the accusative and genitive cases differ according whether the closest vowel is rounded or not and is front or back. The Turkish case allomorphs are presented in the chart below:

Stem Vowel	Nom	Acc	Gen	Dat	Abl	Loc
a/ı	∅	-(y)ı	-(n)ın	-(y)a	-dan	-da
e/i	∅	-(y)i	-(n)in	-(y)e	-den	-de
o/u	∅	-(y)u	-(n)un	-(y)a	-dan	-da
ö/ü	∅	-(y)ü	-(n)ün	-(y)e	-den	-de

Table 1.1: Case Allomorphs in Turkish

Turkish subordinate verbs are similar to English gerunds. They have both nominal and verbal properties: they are case-marked like NPs and have genitive case-marked subjects, but they also subcategorize for arguments that are case-marked just like a verb's arguments. There are four main subordinate clause morphemes in Turkish: the gerundive morphemes -dik and -ecek, and the infinitival morphemes -mek, and -me, and their vowel-harmony variants. (Kural, 1993) claims that -k is a complementizer morpheme within -dik, -ecek, and -mek, while other linguists have claimed that there are no overt complementizers in Turkish.[1] Example (2)a contains a -dik subordinate clause which usually occurs in factive contexts (Kornfilt, 1984). It carries a past or present tense

[1] Kural analyzes -dik as the past tense morpheme -di followed by a complementizer morpheme, and -ecek as a future tense morpheme followed by a complementizer.

reading; -ecek in (2)b is the future tense form of the same morpheme. The subordinate verbs are also marked with a subject-agreement marker that is overt for even third person singular subjects (this type of agreement morpheme also occurs in the possessive constructions on the possessed NP) and they are assigned case like NP arguments.

(2) a. Ben [Ayşe'nin ev-e gel-diğ-i-ni] bil-iyor-um.
I [Ayşe-Gen house-Dat come-Ger-3S-Acc] know-PresProg-1S.
"I know that Ayşe came/is coming home."

b. Ben [Ayşe'nin ev-e gel-eceğ-i-ni] bil-iyor-um.
I [Ayşe-Gen house-Dat come-Ger-3S-Acc] know-Pres-1S.
"I know that Ayşe will come home."

Infinitive clauses marked by -mek are seen in (3); they can occur with or without case-marking in the object position. In (4), we see a clause marked by the morpheme -me which has been called an action nominal or gerund by many Turkish linguists, but I agree with (Kural, 1993) and others that it is an infinitival morpheme that has no tense markings, but unlike -mek, it can take agreement markings. The clauses marked by the -mek morpheme always have a PRO subject controlled by the matrix clause's subject, while those marked by -me and agreement morphemes can have a genitive case-marked subject.

(3) a. Ahmet [PRO yarın sinema-ya git-me-yi] çok ist-iyor.
Ahmet [PRO tomorrow movie-Dat go-Inf-Acc] very want-Prog.
"Ahmet wants to go to the movie tomorrow very much."

b. Ahmet [PRO yarın sinema-ya git-mek] ist-iyor.
Ahmet [PRO tomorrow movie-Dat go-Inf] want-Prog.
"Ahmet wants to go to the movie tomorrow."

(4) Ben [Fatma'nın ev-e git-me-si-ni] ist-iyor-um.
I [Fatma-Gen house-Dat go-Inf-3S-Acc] want-Prog-1Sg.
"I want Fatma to go home."

A small number of matrix verbs subcategorize for direct complement clauses that have no nominalizing morphemes or case. This type of clause is seen in (5) with and without agreement markings.

(5) a. Ali [ben ev-e git-ti-m] san-ıyor.
Ali [I house-Dat go-Past-1sg] think-Prog-(3SG).
"Ali thinks that I went home."

b. Ali beni [ev-e git-ti] san-ıyor.
Ali I-Acc [house-dat go-Past] think-Prog.
"Ali thinks of me that I went home."

The subject-agreement morphology on Turkish verbs allows the information concerning the person and number of a zero subject to be recovered. Turkish sentences within a discourse often have null subjects, especially for first and second person subjects. Objects can also be dropped in Turkish even though Turkish verbs are not marked for object-agreement or with object clitics. Null objects are quite commonly used instead of overt pronouns in discourse contexts where an antecedent to the null object can be easily found (Turan, 1995). For example,

(6) Fatma kitab-ı-nı ara-dı. $\emptyset_s$ $\emptyset_o$ bula-ma-dı.
Fatma book-Poss3s-Acc search-Past. $\emptyset_s$ $\emptyset_o$ find-Neg-Past.
"Fatma searched for (her) book. (She) could not find (it)."

However, since we cannot tell the word order when arguments are dropped, I use linguistic examples which contain overt arguments. The Turkish examples in this dissertation either reflect my judgements as a native speaker of Turkish or are taken from naturally occurring data from the CHILDES corpus (MacWhinney and Snow, 1985) or from colloquial speech and text that I have collected. The Turkish word order facts will be presented in the next chapter.

1.3.2 Combinatory Categorial Grammars (CCGs)

Combinatory Categorial Grammar, CCG, (Ades and Steedman, 1982; Steedman, 1985; Steedman, 1987) is an extension of Categorial Grammars (Ajdukiewicz, 1935; Bar-Hillel, 1953). Categorial grammars and their various extensions are concerned with capturing the function-argument relations in language and preserving a parallel and compositional syntax and semantics. They are lexicalist formalisms in which most of the grammatical information is placed in the lexicon. They avoid movement or deletion rules and often have a more flexible surface structure than phrase-structure grammars.

There are many different formalisms that extend categorial grammars in order to handle discontinuous constituents. The different extensions differ mostly in the way they deal with long distance dependencies without using movement rules or traces. (Lambek, 1958) introduces an algebraic calculus in which function composition and type-raising are theorems that can be used to deal with this phenomenon. (Steedman, 1985) generalizes the function composition rules in order to capture long distance dependencies while preserving the adjacency restriction in forming constituents. The composition rules are restricted in his grammar in order to avoid overgeneration. The Lambek tradition (van Benthem, 1988; Moortgat, 1988; Hepple, 1990) avoids restrictions on rules in order to preserve a simple and free underlying calculus. (Moortgat, 1988; Hepple, 1990) extend Lambek calculus with special rules and categories (e.g. modality operators) in order to handle extraction and island phenomena.

(Bach, 1988) introduces wrap-rules that give up adjacency requirements in forming constituents in order to handle long distance dependencies. Bach also follows the Montague tradition in developing a categorial semantics (Montague, 1974; Dowty, Wall, and Peters, 1981; Dowty, 1982; Partee and Rooth, 1983; Bach, 1988). In the Bach approach, derivations directly reflect binding and control relationships (Bach, 1988; Szabolcsi, 1987; Dowty, 1988; Jacobson, 1990). Steedman preserves the principle of adjacency in derivations and defines binding and control intrinsically using predicate-argument structure. A more detailed introduction to categorial grammars and the evolution of related formalisms is found in (Wood, 1993). The rest of this section will provide an introduction to CCGs.

CCG was developed in (Ades and Steedman, 1982; Steedman, 1985; Steedman, 1987) to account for coordination and long distance dependencies in extraction without the use of movement rules and traces. Grammatical entities in CCGs, as in all categorial grammars, are of two types, *functions* and *basic categories*, capturing the inherent function-argument relations in language. For example, a lexical item such as "Mary" is associated with the basic category NP which a shorthand for a set of syntactic and semantic features. An intransitive verb such as "sleeps" is associated with the category $S\backslash NP$ which represents a function looking for an argument of

category NP on its left, indicated by the backwards slash, and results in the category S once it finds this NP. Functions may have more than one argument; for example, a transitive verb such as "likes" is associated with the category $(S\backslash NP)/NP$; this represents a function that is looking for an argument of type NP, its object, on its right and results in a function which is looking for another argument of type NP, its subject, on its left and results in the category S.

A small set of combinatory rules (related to Curry's combinatory logic (Curry and Feys, 1958)) serves to combine these categories while preserving a transparent relation between syntax and semantics. The rules only apply to linguistically-realized adjacent entities in a bottom-up fashion. They combine both the syntactic and the semantic types associated with two constituents to derive a resulting constituent in a compositional fashion which maintains syntactic and semantic parallelism.

The *application* rules below allow functors to combine with their arguments. In Steedman's notation, the syntactic category of each constituent in the rule is separated from its semantic interpretation by a colon, and Fy represents the application of the function's interpretation, F, to the argument's interpretation, y.

(7) a. **Forward Application (>):**

$X/Y : F \quad Y : y \Rightarrow X : Fy$

b. **Backward Application (<):**

$Y : y \quad X\backslash Y : F \Rightarrow X : Fy$

These rules can be used to provide a derivation for the following sentence:

(8)
$$\begin{array}{llll} \text{Pat} & \text{likes} & & \text{Chris.} \\ \text{NP} & \text{S}\backslash\text{NP}/\text{NP} & & \text{NP} \\ & & \dfrac{\qquad\qquad}{\text{S}\backslash\text{NP}}> & \\ & \dfrac{\qquad\qquad}{\text{S}}< & & \end{array}$$

In addition, CCGs include *composition* rules which combine two function categories together, syntactically and semantically. If these two functors have the semantic interpretation F and G, the result of their composition has the interpretation $\lambda x F(Gx)$; the symbol B represents Curry's combinator which composes two functions.

(9) a. **Forward Composition (>B):**

$X/Y \quad Y/Z \Rightarrow X/Z$

b. **Backward Composition (<B):**

$Y\backslash Z \quad X\backslash Y \Rightarrow X\backslash Z$

c. **Forward Crossing Composition (>Bx):**

$X/Y \quad Y\backslash Z \Rightarrow X\backslash Z$

d. **Backward Crossing Composition (<Bx):**

$$Y/Z \quad X\backslash Y \Rightarrow X/Z$$

X, Y, and Z in these rules are variables which can match any category; grammars of different languages may exclude certain rules or place certain restrictions upon what categories may instantiate the variables in the rules.

The composition rules can be generalized to categories with more than one argument. The symbol B^n refers to the composition of Curry's combinator B with itself n times.

(10) a. **Generalized Forward Composition ($> B^n$):**

$$X/Y \quad Y|_1 Z_1 \ldots|_n Z_n \Rightarrow X|_1 Z_1 \ldots|_n Z_n$$

b. **Generalized Backward Composition ($< B^n$):**

$$Y|_1 Z_1 \ldots|_n Z_n \quad X\backslash Y \Rightarrow X|_1 Z_1 \ldots|_n Z_n$$

In the English CCG, n may be bounded by the maximum valency in the lexicon (Steedman, 1989). With such a restriction, CCGs are weakly equivalent to other mildly context-sensitive grammars, TAGs, HGs, and LIGs (Weir and Joshi, 1988), but if n is unbounded the resulting grammar is more powerful; this will be discussed further in Chapter 4.

The application and composition rules provide general schemas that can combine the rich lexical functions and basic element categories. In addition, a type-raising rule can be used in the lexicon to convert basic elements into functions. For example, an NP category can be type-raised to the category $S/(S\backslash NP)$ representing a function looking for an intransitive verb on its right (the intransitive verb is defined as a function which is looking for the NP on its left). Type-raising has been used in the Montague tradition to provide the correct interpretation for quantified NPs. In CCGs, type-raising is used to capture syntactic coordination and extraction facts. The most general type-raising schemas (Steedman, 1989) can be written as

(11) $$X \rightarrow T/(T\backslash X)$$
$$X \rightarrow T\backslash(T/X)$$

These are called order-preserving rules by (Dowty, 1988); non-order preserving type-raising, such as $T/(T/X)$, is not needed for the English grammar. Although the schemata capture the general nature of type-raising, they may overgenerate ungrammatical sentences if they freely apply to any category X during the derivation of the sentence. To avoid this, Steedman places type-raising in the lexicon, converting a selected set of basic categories into functions, while Dowty assigns determiners categories which type-raise the common nouns to be generalized quantifiers in English.[2] As mentioned by (Steedman, 1985) among others, in languages with case-marking, the case-markers may function to raise nouns into type-raised categories with a grammatical relation indicated by the case.

[2] After type-raising, we may end up with many categories possible for each lexical item. As suggested by (Partee and Rooth, 1983), a strategy of trying simpler types first may help the processing load.

Type-raising, in combination with the composition rules, in CCGs can produce constituents that are non-traditional. For example, in the sentence "Mary saw John", a traditional grammar would produce the bracketing in (12)a whereas in CCGs both (12)a and (12)b is possible. A type-raised subject can combine with the verb before the VP constituent has been completed.

(12) a. [Mary [saw John]$_{vp}$]$_{s}$

b. [[Mary saw]$_{s/np}$ John]$_{s}$

Permitting nontraditional constituents in the grammar provides a handle on coordination and extraction as well as providing an incremental (left to right) parsing and interpretation strategy. For example, right node raising in English can be handled in CCGs by allowing a type-raised subject to combine with the verb by the forward composition rule. The general coordination schema below conjoins "like" constituents:

(13) a. **Coordination (&):** X conj X $\Rightarrow$ X

b.

[John	cooked]	and	[Mary	ate]	the beans.
S/(S\NP)	S\NP/NP	conj	S/(S\NP)	S\NP/NP	NP
	——> B S/NP			——> B S/NP	
	——————— (&) S/NP				
	——————————————— > S				

The same method of using the type-raising and composition rules provides an account of *leftward object extraction* in English:

(14)

the beans	which	[Mary	ate]
NP	(NP\NP)/(S/NP)	S/(S\NP)	S\NP/NP
		——> B S/NP	
	——————————— > NP\NP		
——————— < NP			

In the CCG for Dutch (Steedman, 1985; Steedman, 1993), Steedman introduces a general type-raised category which contains a variable ranging over all verbal functions (i.e. $v/(v\backslash NP)$ where v is a variable). The use of the variable provides a convenient generalization over all possible verbal functions.[3] This allows a sequence of NPs, which may belong to different clauses, to compose together producing a left-branching NP constituent that can then be coordinated or used in a relativization context, like the ones below in Dutch:

(15) a. dat [Jan Piet] en [Cecilia Henk] de kinderen zag laten zwemmen
that [Jan Piet] and [Cecilia Henk] the children saw make swim
"that Jan saw Piet and Cecilia saw Henk make the children swim."

[3]In the Dutch grammar, this polymorphic variable is typed; the variable is restricted to unify only with categories in the transitive closure over functions into S and S_{te-inf} but not into S'.

b. ..de leraar die Hendrik Cecilia zag helpen.
..the teacher who Hendrik Cecilia saw help.
"..the teacher who Hendrik saw Cecilia help."

The analysis of coordination and relativization in English and Dutch crucially depends on the production of nontraditional constituents. For example, in English, we form the nontraditional constituent consisting of the subject and the verb in order to account for the extraction of an object in relative clause formation and the coordination in right node raising constructions. The freer surface structure of CCG allows an analysis of these constructions without resorting to movement rules and traces. A nontransformational theory such as CCG is computationally efficient. (Weir and Joshi, 1988) prove that CCG is weakly equivalent to other mildly context-sensitive grammars, and (Vijay-Shanker and Weir, 1990; Vijay-Shanker and Weir, 1993) show that CCG is polynomially parsable.

In this dissertation, I will not provide an analysis for coordination in Turkish. I briefly discuss type-raising in the Turkish grammar in Chapter 3, page 63, and I discuss the potential use and power of type-raising with variables and unbounded composition (B^n) to handle scrambling, in the next chapter, page 32. However, I will show that the flexibility of surface structure in CCGs is crucial for capturing word order freeness in Turkish and in integrating the information structure of a sentence with its surface structure.

My work is influenced by (Steedman, 1991) in which a theory of prosody, closely related to a theory of information structure, is added to CCGs. Intonational phrase boundaries often do not correspond to traditional phrase structure boundaries. However, by using the CCG type-raising and composition rules, the CCG formalism can produce nontraditional syntactic constituents that match the intonational phrasing. Steedman argues that the surface structure of a sentence is identical to the intonational structure, and thus the competence grammar must have a freer notion of syntactic constituency. The composition rules in CCG allow many different derivations of a sentence, however this ambiguity is not spurious, but in fact, necessary to capture prosodic and pragmatic phrasing. I will discuss (Steedman, 1991)'s method of integrating information structure with surface structure in CCGs further in Chapter 2, page 25.

Chapter 2

Motivation for the Dissertation

In this chapter, I present the motivation for the formalism I develop in the dissertation. In the first section, I present the Turkish data with respect to word order and outline what is required of a competence grammar for "free" word order languages on page 18. Then, in Section 2, I summarize some of the previous approaches to "free" word order languages and in Section 3, explain why I have chosen to develop a categorial formalism based on Combinatory Categorial Grammars (CCG) (Ades and Steedman, 1982; Steedman, 1985). In Section 4, I explain why the CCG formalism must be altered to handle "free" word order languages. CCGs must be extended in order to be formally and descriptively adequate to handle the unbounded nature of long distance scrambling, i.e. the permutation of elements from an unbounded number of clauses over unbounded distances. In Section 5, I summarize the motivation for the Multiset-CCG formalism developed in this dissertation.

2.1 Capturing the Data

2.1.1 "Free" Word Order in Turkish

The most common word order used in simple transitive sentences in Turkish is SOV (Subject-Object-Verb), however all six permutations of a transitive sentence are grammatical, as seen in (1), since case-marking, rather than word order, serves to differentiate the arguments in Turkish.[1] This word order variation within a clause has been called *local scrambling*. The relative frequencies of these different word orders is seen next to each example; these frequencies were determined by (Slobin and Bever, 1982) from 500 utterances of spontaneous speech. As can be seen, 52% of

[1] As described on page 5, the accusative, dative, genitive, ablative, and locative cases are associated with specific morphemes (and their vowel-harmony variants) which attach to the noun; nominative case and subject-verb agreement for third person singular are unmarked.

the transitive sentences were not in the canonical SOV word order; thus, "free" word order is a phenomenon that we must capture in order to model natural Turkish discourses.

(1) a. Fatma Ahmet'i gör-dü. (SOV 48%)
Fatma Ahmet-Acc see-Past.
"Fatma saw Ahmet."

b. Ahmet'i Fatma gördü. (OSV 8%)

c. Fatma gördü Ahmet'i. (SVO 25%)

d. Ahmet'i gördü Fatma. (OVS 13%)

e. Gördü Fatma Ahmet'i. (VSO 6%)

f. Gördü Ahmet'i Fatma. (VOS <1%)

The traditional propositional interpretation assigned to all six of the sentences above is *see(Fatma,Ahmet)*. However, each word order conveys a different discourse meaning only appropriate to a specific discourse situation. Turkish speakers often place the topical information to link the sentence to the previous context at the start of the sentence, the important and/or new information immediately before the verb, and the background information that is not really needed but may help the hearer understand the sentence better, after the verb. This context-dependent aspect of meaning is called the information structure of the sentence.

As will be discussed in Chapter 5, I define the information structure of Turkish sentences by dividing each sentence into a *topic* and a *comment* component, where the topic is the main element that the sentence is about, and the comment is the main information the speaker wants to convey about the topic. Assuming the hearer's discourse model or knowledge store is organized by topics, the sentence topic can be seen as specifying an "address" in the hearer's knowledge store (Reinhart, 1981; Vallduví, 1990). In Turkish, the sentence topic is placed in the sentence-initial position (Erguvanlı, 1984). I further divide the comment into the *focus* and the *ground*. The *focus* is the most information-bearing constituent in the sentence, (Vallduví, 1990); it is the new or important information in the sentence. In Turkish, the focus is usually placed in the immediately preverbal position and receives the primary stress and highest pitch in the sentence. The post-verbal elements in Turkish have a gradually falling intonation and are always occupied by known discourse entities evoked by the prior discourse; thus, they help to ground the sentence in the current context.

We can now explain why certain word orders are appropriate or inappropriate in a certain context, in this case wh-questions. For example, a speaker may use the SOV order in (2b) to answer the wh-question in (2a) because the speaker wants to *focus* the new object, Ahmet, and so places it in the immediately preverbal position. However, given a different wh-question in (3), the OSV word order is used to indicate that the object Ahmet is the *topic*, a link to the

previous context, while the subject, Fatma, is the *focus* of the answer. Here, we translate these Turkish sentences to English using different "stylistic" constructions (e.g. topicalization, it-clefts, phonological focusing indicated by capitals, etc.) in order to preserve approximately the same meanings.

(2) a. Fatma kim-i gör-dü?
Fatma who-Acc see-Past?
"Who did Fatma see?"

b. Fatma Ahmet'i gör-dü. SOV
Fatma Ahmet-Acc see-Past.
"Fatma saw AHMET."

(3) a. Ahmet'i kim gör-dü?
Ahmet-Acc who see-Past.
"Who saw Ahmet?"

b. Ahmet'i Fatma gör-dü. OSV
Ahmet-Acc Fatma see-Past.
"As for Ahmet, it was FATMA who saw him."

Crucially, we cannot interchange the answers to the questions above. The word order in (3)b would sound strange and inappropriate in response to the question in (2)a, and the word order in (2)b would be an inappropriate response to (3)a. Each word order expresses a different information structure, a different context-dependent meaning.

Scrambling to post-verbal positions is also very common in Turkish. Some "free" word order languages, such as German, Japanese, Korean, are said to be strictly verb-final. (Kuno, 1980) shows that afterthoughts to the right of the verb are possible in colloquial Japanese, however, it has been claimed that these constructions are very different than preverbal scrambling.[2] In Turkish, there is no reason to believe that there is a syntactic difference between movement to the right or the left of the verb. First of all, post-verbal elements are very common in both spoken and written speech in Turkish. In addition, unlike Japanese, the intonation of a Turkish sentence is such that there is no pause between the verb and the post-verbal elements. The post-verbal elements are spoken without stress and with low pitch, but they are in the same intonation contour as the rest of the sentence.

Adjuncts can also occur in different sentence positions in Turkish sentences depending on the context. The different positions of the sentential adjunct "yesterday" in the following sentences result in different discourse meanings, much as in English. Thus, components of the information

[2](Whitman, 1991) argues that post-verbal placement in Japanese is a movement operation just like right dislocation in English, but (Kuno, 1973) argues for a base-generated analysis.

structure of a sentence do not have to be arguments in the predicate-argument structure of the sentence.

(4) a. Fatma Ahmet'i dün gör-dü.
Fatma Ahmet-Acc dün see-Past.
"Fatma saw Ahmet YESTERDAY."

b. Dün Fatma Ahmet'i gör-dü.
Yesterday Fatma Ahmet-Acc see-Past.
"Yesterday, Fatma saw Ahmet."

c. Fatma Ahmet'i gör-dü dün.
Fatma Ahmet-Acc see-Past yesterday.
"Fatma saw Ahmet, yesterday."

Clausal arguments, just like simple NP arguments, can occur anywhere in the matrix sentence as long as they are case-marked, (5)a and b. The structure of subordinate verbs in Turkish is described in the Introduction, page 5. The arguments and adjuncts within most embedded clause can occur in any word order, also seen in (5)a and b.[3] As indicated by the translations, word order variation in complex sentences also affects the interpretation. Embedded clauses can play a role in the information structure of the matrix clause, but they also have their own information structure that is distinct from the matrix clause's IS.

(5) a. Ayşe [dün Fatma'nın git-tiğ-i-ni] bil-iyor.
Ayşe [yest. Fatma-Gen go-Ger-3S-Acc] know-Prog.
"Ayşe knows that yesterday, FATMA left."

b. [Dün git-tiğ-i-ni Fatma'nin] Ayşe bil-iyor.
[Yest. go-Ger-3S-Acc Fatma-Gen] Ayşe know-Prog.
"It's AYŞE who knows that she, Fatma, left YESTERDAY."

In complex sentences with clausal arguments, elements of the embedded clauses can occur in matrix clause positions; this has been called *long distance scrambling* in transformational theories. Long distance scrambling poses a greater problem for a language processing model than local scrambling because the arguments occur out of the domain of their verb. One must be able to recover the appropriate predicate-argument relations of the embedded clause and the matrix clause without ambiguity.

Long distance scrambling is only used by speakers for specific pragmatic functions. Generally, an element from the embedded clause can occur in the sentence initial topic position of the matrix clause (e.g. (6)b) or to the right of the matrix verb as backgrounded information (e.g. (6)c).[4]

[3]The immediately preverbal element in each clause receives stress.

[4]I have put in coindexed traces and italicized the scrambled elements in these examples to help the reader; I am not making the syntactic claim that these traces actually exist. In fact, I will not adopt a transformational theory,

(6) a. Fatma [Esra'nın kitab-ı oku-duğ-u-nu] bil-iyor.
Fatma [Esra-Gen book-Acc read-Ger-3sg-Acc] know-Prog.
"Fatma knows that Esra read the book."

b. *Kitabı*$_i$ Fatma [Esra'nın e$_i$ okuduğunu] biliyor.
Book-Acc$_i$ Fatma [Esra-Gen e$_i$ read-Ger-3Sg-Acc] know-Prog.
"As for the book, Fatma knows that Esra read it."

c. Fatma [Esra'nın e$_i$ okuduğunu] biliyor *kitabı*$_i$.
Fatma [Esra-Gen e$_i$ read-Ger-3sg-Acc] know-Prog *book-Acc*$_i$.
"Fatma knows that Esra read it, the book."

Word orders which do not correspond with some pragmatic function are not used in natural discourse and sound awkward or ungrammatical if used. For example, there is no pragmatic function associated with the second position in the matrix sentence, and long distance scrambling to this position sounds awkward, ex. (7)a, unless commas and pauses are added. In addition, elements of embedded clauses cannot scramble to the position immediately before the matrix verb, ex. (7)b. However, any constituent of the matrix sentence can scramble freely into this position; moreover, elements of the embedded clause can occur between the subordinate verb and the matrix verb when they are locally scrambled within their own clause as in (7)c. This restriction on the immediately preverbal position will be discussed further in the next chapter, page 59.

(7) a. ??Ahmet *bu kitab-ı*$_i$ Fatma'ya [ben-im e$_i$ oku-duğ-um-u] söyle-di.
??Ahmet *this book-Acc*$_i$ Fatma-Dat [I-Gen e$_i$ read-Ger-1Sg-Acc] say-Past.
"Ahmet told Fatma that I read this book."

b. *Ahmet [benim e$_i$ okuduğumu] Fatma'ya *bu kitabı*$_i$ söyledi.
*Ahmet [I-Gen e$_i$ read-Past-Ger-1Sg-Acc] Fatma-Dat *this book-Acc*$_i$ say-Past.

c. Ayşe [benim okuduğumu bu kitabı], biliyor.
Ayşe [I-Gen read-Past-Ger-1Sg-Acc this book-Acc] know-Prog.
"Ayşe knows that I read this book."

The information structure (IS) is distinct from predicate-argument structure (AS) in languages such as Turkish because adjuncts and elements long distance scrambled from embedded clauses can take part in the IS of the matrix clause even though they are not arguments in the AS of the matrix clause.

Native speakers can understand sentences in which elements have been long distance scrambled over an unbounded distance and sentences in which more than one element from a clause has been long distance scrambled. The more one scrambles things around, the harder the sentence is to

but I may use the terms 'scrambling' and 'movement' as an easy way of speaking about word order variation.

process, but there is no clear cut-off point in which sentences become ungrammatical.

(8) Bu kitab-ı$_j$ Fatma [t$_i$ [t$_j$ oku-mak] iste-diğ-im-i] bil-iyor ben-im$_i$.
This book-Acc$_j$ Fatma [t$_i$ [t$_j$ read-Inf] want-Ger-1S-Acc] know-Prog I-Gen$_i$.
"As for this book, Fatma knows that I want to read it."

Long distance scrambling can occur out of almost all embedded clauses in Turkish. However, it is harder to extract elements from some adjunct clauses, as seen in (9) below.

(9) a. Berna [PRO ödev-in-i bit-ir-ince] ban-a yardım ed-ecek.
Berna [PRO hw-3Poss-Acc finish-Aor-Ger] I-dat help do-Fut.
"When (she) finishes (her) homework, Berna is going to help me."

b. *Ödevini bana Berna [bitirince] yardım edecek.
*$Hw\text{-}3P\text{-}Acc_i$ I-dat Berna [finish-ger] help do-3sg.
"As for her homework, Berna is going to help me when she finishes it."

c. *[Berna bitirince] bana yardım edecek *ödevini*.
*[Berna finish-ger] I-dat help do *hw-3Ps-Acc*.
*"When she finishes it, Berna is going to help me, her homework."

The syntactic restrictions on scrambling in Turkish will be discussed further in the next chapter. Among the word order constraints that must be captured are head-final NP clauses, continuous simple NPs, and island behaviour in some adjunct clauses.

2.1.2 Requirements for the Formalism

As motivated from the data given above, a formalism to capture the syntax of "free" word order languages such as Turkish must be flexible enough to handle:

- the word order variation of the arguments in a clause,
- the word order variation of the adjuncts in a clause,
- the long distance scrambling of elements from embedded clauses into the matrix clause,
- syntactic restrictions in word order (e.g. islands, head-final clauses).

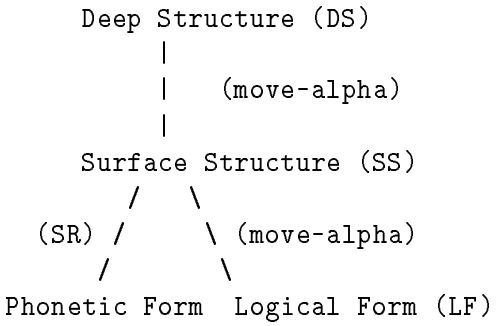
In addition, to capture the context-appropriate use of word order, the formalism must:

- associate information structure components such as topic and focus with the appropriate sentence positions,
- capture the information structure of complex sentences,
- allow elements that are not arguments in the predicate-argument structure of a clause, to take part in the information structure of the clause.

2.2 Previous Approaches

2.2.1 The Syntactic Movement Analysis

Many linguists work within a generative theory of grammar based the following well-known model of Universal Grammar (Chomsky, 1971; Chomsky and Lasnik, 1977).



One of the earliest claims in syntax literature about word order is that it is purely a stylistic variation which has no effect on the meaning of the sentence. According to the Stylistic Rule Hypothesis (SRH) as described by (Rochemont, 1978), some noncanonical word orders are the result of stylistic rules (SR) operating between the Surface Structure (SS) and Phonetic Form (PF) components. These rules do not affect the meaning of the sentence, and thus have no connection to the Logical Form (LF) level which provides the input for semantic interpretation. Rochemont's earlier arguments for SRH include the fact that syntactic operations like wh-movement, subject-aux inversion, and topicalization can not apply in focus constructions in English; they are syntactically frozen as in (10)b. This implies that the stylistic rules apply after Surface Structure and are independent of syntactic operations.

(10) a. A man walked into the room with PURPLE hair.

b. *What color hair did a man walk into the room with?

However in later work, (Rochemont and Culicover, 1990) abandon the SRH and claim that English focus constructions are derived by the application of move-α prior to Surface Structure. They show that the syntactic freezing characteristic can be handled by UG principles such as ECP and subjacency and island restrictions on move-α at Surface Structure. (Culicover, 1980) shows that the so-called Stylistic Rules can influence the scope of negation and other logical operators at the LF level (contradicting (Rochemont, 1978)), and thus, the stylistic movement must occur before the LF level. (Horvath, 1985) among others also argues against the SRH for "free" word order languages. She argues that word order in these languages does affect the interpretation of sentences, as well as binding properties that take place at the LF level.

In the generative grammar of Government and Binding (GB) (Chomsky, 1981), noncanonical word orders are analyzed in terms of a general movement rule, move-α, applying between Deep Structure and Surface Structure.[5] For instance, (Horvath, 1985) claims that word order variation in Hungarian is the result of movement rules similar to wh-movement in English. Recent work on scrambling, (Saito, 1989; Mahajan, 1990; Webelhuth, 1989; Grewendoorf and Sternefeld, 1990), has focused on whether scrambling is movement to argument (A) positions (like NP-movement in English) or movement to operator (A') positions (like wh-movement in English). In English, the landing sites of NP-movement and wh-movement have different binding characteristics. However, scrambling does not fit neatly into either one of these types of movement. Webelhuth, working on scrambling in German, argues that the landing sites of scrambling may exhibit the characteristics of A and A' positions simultaneously, while Mahajan argues that scrambling in Hindi can be A or A' movement but that there are no mixed A/A' positions. Local and long distance scrambling in Turkish can show both A and A' movement characteristics (Kural, 1991; Hoffman and Turan, 1991), just like in German and Korean (Lee and Santorini, 1994). I will not make any claims regarding A vs. A' status of scrambling in Turkish because the facts are far from clear; these issues are discussed to some extent in (Kural, 1991). See also (Bayer and Kornfilt, 1994) for problems with a move-α analysis of scrambling.

Island effects associated with movement are hard to find in Turkish. As seen below, extraction and long distance scrambling out of adjunct and sentential subject clauses are possible in Turkish, even if not very common. This casts a doubt on whether word order variation involves movement at all. Thus, I do not adopt a movement analysis for "free" word order in Turkish.

(11) a. *Ankara'dan*$_i$ sen [e$_i$ dün gel-en] adam-ı tan-ıyor mu-sun?
Ankara-Abl$_i$ you [e$_i$ yest. come-Rel] man-Acc know-Prog Quest-2Sg?
"As for Ankara, do you know the man who came yesterday from (there)?"

b. [e$_i$ Çözmek] zor bu problemi$_i$.
[e$_i$ Solve-Inf] hard this problem-Acc$_i$.
"To solve (it) is hard, this problem."

A similar argument against a movement analysis is made by (Rooth, 1985) for prosodic focus constructions in English. Intonationally marked focus in English sentences can be given an interpretation which is the result of movement at LF. Strong and Weak Crossover Facts have been used to show that Focused NPs behave like wh-phrases in English and thus must involve movement at LF. The example below can be explained by assuming the focused item raises at LF to adjoin next to 'only', and this movement causes the WCO effect.

[5] Under the recent Minimalist Program (Chomsky, 1993), movement occurs only when it is motivated by feature-checking.

(12) *We only expect the woman he$_i$ loves to miss HIM$_i$.

However, Rooth points out that if focused NPs move at LF, this movement is different from wh-movement and quantifier raising because it does not obey island constraints. The first example below shows that a wh-phrase cannot be extracted from a complex-NP (Ross, 1967), however a focused NP 'THE ZONING BOARD' can associate with the adverb 'only' which suggests that it has moved next to "only" at LF even though it is within a complex NP.

(13) a. *Which board did they investigate [the question whether you know the woman who chaired]?

b. They only investigated [the question whether you know the woman who chaired THE ZONING BOARD].

Thus, this data suggests that focused NPs in English are not given an interpretation via movement at LF. Rooth's Alternative Semantics allows focused NPs to be interpreted without resorting to a movement analysis.

In Turkish, the fact that the effects of scrambling on binding do not cleanly or consistently show A or A' movement characteristics, combined with the lack of evidence for islands, casts doubt on whether scrambling is the result of movement rules at all. The optional free word order variation seems to have different characteristics than the obligatory NP-movement and wh-movement observed in English. These observations support the categorial framework I develop in which both local and long distance scrambling can be explained uniformly without resorting to syntactic movement rules. In fact, non-transformational analyses have been adopted by many computational linguists because of their computational efficiency. Transformations such as movement rules greatly increase the formal power of a grammar (Peters and Ritchie, 1973) which may cause problems for efficient parsing and generation.

2.2.2 Integrating Syntax and Information Structure

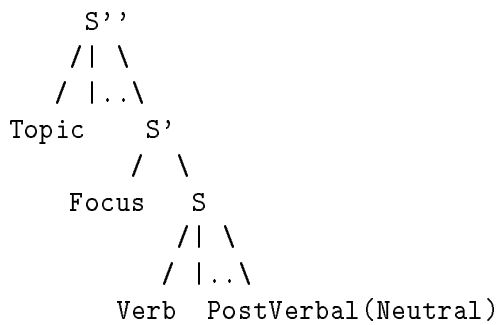
Most syntactic theories do not address the context-dependent meaning reflected by the word order in "free" word order languages. However, there have been a few approaches that integrate information structure (IS) (notions such as topic and focus) with a generative theory of grammar in order to capture the context-dependent interpretation of different word orders. The most common approach is to integrate this information with the syntactic representation at surface structure by allowing constituents to move to certain phrase-structure positions in order to be interpreted as topic or focus (Sgall, Hajicova, and Benesova, 1973; Kiss, 1987; King, 1993). There are also nontransformational theories such as GPSG, HPSG and LFG, that separate the syntactic rules into Immediate Dominance (ID) rules and Linear Precedence (LP) rules, and the LP rules order the topic and focus in the sentence string. Alternatively, we can integrate notions such as

topic and focus into the semantic component of a sentence (Rooth, 1985; Partee, 1991). In this section, I will summarize these methods of adding information structure to the grammar, before presenting the CCG approach.

2.2.2.1 IS at Surface Structure

One of the earliest theories to capture "free" word order languages, specifically Czech, is the Prague school of linguistics (Sgall, Hajicova, and Benesova, 1973; Sgall, Hajicova, and Panevova, 1986). Sgall et al propose a level of Communicative Dynamism (CD), which corresponds to something like deep structure in GB. Communicative dynamism is a semantic property of sentence elements that indicates whether the element is more like a topic (mostly given information) or more like a focus (mostly new information) in the sentence. The CD of elements produces a default ordering (called the systemic ordering) from the most topical element to the most focused element of the sentence. In "free" word order languages, the surface structure is equivalent to this systemic order. In languages such as English, movement rules operate on the systemic order to produce the surface structure. Thus, the surface structure directly reflects the information structure in "free" word order languages, and there is a deep structure which reflects the information structure in fixed word order languages.

Other approaches that integrate the information structure with the surface structure of a sentence have used movement rules to model "free" word order languages (Kiss, 1987; Vallduví, 1990; King, 1993). For example, (Kiss, 1987) proposes a configurational phrase structure for Hungarian where phrase structure positions indicate the information structure, the topic and the focus of a sentence, rather than the grammatical relations, such as subject and object. In the tree seen below, the verb and its complements are base-generated in the S subtree, the propositional component of the sentence. Then, move-α transformations fill the operator positions c-commanding S, and the moved elements are associated with the pragmatic functions of topic and focus.



The formalism that I propose in this dissertation is similar to this approach in associating pragmatic functions with specific sentence positions so that the information structure is directly reflected by the surface structure of the sentence, however my approach does not use movement rules. I use a nontransformational, lexicalist, and compositional formalism that has a more flexible notion of constituency than traditional grammars.

2.2.2.2 IS in LP rules

Many nontransformational approaches model "free" word order languages by separating the grammar into two components: immediate dominance (ID) and linear precedence (LP) rules. Gazdar and Pullum introduce the first ID/LP formalism using Generalized Phrase Structure Grammar (GPSG) in (Gazdar et al., 1985). (Uszkoreit, 1987) presents a GPSG framework that handles German word order variation in simple clauses. His LP rules contain a rule which orders focused information following nonfocused information, (-Focus < +Focus) as well as rules that order syntactic information such as (NP-Nom < NP-Acc). The same ID/LP approach has been used in Head-Driven Phrase Structure Grammar (HPSG) which is a descendent of GPSG (Pollard and Sag, 1987). (Steinberger, 1994) extends Uszkoreit's LP rules for a German HPSG with information about the order of adverbs. (Engdahl and Vallduvi, 1994) propose an HPSG for Catalan, another "free" word order language, using dislocation rules and LP constraints on the information structure components. The ID/LP approach has also been used in Lexical-Functional Grammars (LFGs) (Mohanan, 1982; King, 1993).

The grammar I propose is similar to ID/LP formalisms in that the grammar is divided into two components, the syntactic/semantic component and the ordering component. However, this does not exactly correspond to the ID/LP division. I place syntactic restrictions on word order, such as island behavior or that NPs precede verbs in languages such as German, in the syntactic part of the grammar, not in ordering rules. The ordering part of the grammar is just concerned with ordering the information structure components such as topic and focus. Thus, the division in the grammar components is not between dominance and word order, but between the predicate-argument structure and the information structure. In addition, the current ID/LP approaches do not handle complex clauses with embedded information structures, long distance scrambling, and island restrictions on scrambling. My approach allows for embedded information structures, provides a uniform analysis of local and long distance scrambling, and captures syntactic and pragmatic restrictions on word order as well as word order freeness in Turkish.

2.2.2.3 IS in Semantics

Recent work by semanticists has shown that focus interacts with the scope of quantifiers and focus-particles such as "only" and "even" in the semantic representation (Rooth, 1985; Krifka, 1992). (Partee, 1991), inspired by the Prague School, integrates compositional semantics with the context-dependent notions of topic as well as focus. She notes that the traditional tripartite semantic structure for quantified sentences, *Operator [Restrictor] : Nuclear Scope*, interacts with pragmatic functions such as topic and focus. The generalization is that the topical (presupposed) elements are found in the Restrictor clause while focused elements (the assertion of the sentence) are found in the Nuclear Scope in the semantic representation. This is demonstrated in English sentences where an adverbial quantifier interacts with a focused item, indicated by capital letters in the interpretation in b.

(14) a. Mary usually takes JOHN to the movies.

b. Usually_{op} [when Mary takes x to the movies$]_{restr.}$: $[x = \text{John}]_{n.s.}$

(15) a. Mary usually takes John to the MOVIES.

b. Usually_{op} [when Mary takes John to $x]_{restr.}$: $[x = \text{movies}]_{n.s.}$

Thus, Partee incorporates the pragmatic functions of topic and focus (or presuppositions and assertions) into the tripartite semantic representation and interpretation. In addition, Partee suggests that there are "recursive contexts" such that we interpret the restricter clause (the presuppositions) in the current context and the nuclear scope clause (the assertions) in light of the presuppositions just evaluated.

In Turkish, the interpretation of a sentence must include its information structure as well as the traditional truth-conditional semantic representation. The interaction between the informational topic and focus and LF level semantics (e.g. binding and quantification) is beyond the scope of this study.

2.3 Why CCGs?

I have chosen to adapt Combinatory Categorial Grammars (CCGs) to handle "free" word order languages. CCG is a computationally attractive formalism because it is a lexicalist, non-transformational, mildly context-sensitive, and polynomially parsable formalism. However, the greatest advantage of using a CCG-based formalism is its flexible surface structure which can directly reflect the information structure of sentences. We can easily integrate the information structure with the syntax in a CCG-based formalism, to capture both the syntax and the context-dependent interpretation of "free" word order.

My work is influenced by (Steedman, 1991) in which a theory of prosody, closely related to a theory of information structure, is integrated with CCGs. CCG has a freer constituent structure than traditional grammars, and this flexibility allows syntactic constituency to exactly correspond to prosodic constituency. Steedman assigns each constituent in the CCG a prosodic category based on its pitch accent. The interface between the CCG and the prosodic level is simple: during the derivation of a sentence, two syntactic categories are allowed to combine only if their prosodic categories can also combine.

Intonational phrases often correspond to a unit of planning or presentation with a single discourse function. To capture this, the prosodic categories in CCG are associated with interpretations that are certain pragmatic primitives. For example, the high pitch accent H* is associated with a focused constituent within the rheme of the sentence, while the L+H* pitch accent is associated with the focus of the theme of the sentence. The theme is roughly what the sentence is about, while the rheme is the new information that speaker is communicating about the theme (Halliday, 1967), and these components of the sentence are in different prosodic phrases, separated by boundary tones such as LH%. The example below (from (Prevost and Steedman, 1993)) shows the appropriate prosody reflecting a certain information structure in the answer to a question. Capital letters indicate a high pitch and accent.

(16) Q: I know that the OLD widget had a SLOW processor.
But what processor does the NEW widget include?

A:	(The	NEW	widget includes)		(a	FAST	processor).
		L+H*	LH%			H*	LL%
	Ground	*Focus*	*Ground*		*Ground*	*Focus*	*Ground*
		Theme				*Rheme*	

In the CCG derivation for this sentence, the surface structure directly reflects the prosodic phrasing. The subject and verb form a constituent in the surface structure as well as a constituent in the prosodic information structure. A different derivation of the same sentence would correspond to a different intonational phrasing and thus, a different information structure.

The information structure in Turkish is largely expressed through word order rather than prosody. I have chosen to extend CCGs to capture Turkish word order in order to take advantage of CCG's flexible and compositional notion of syntactic constituency. In such a formalism, we can integrate the information structure with the surface structure of the sentence, without using movement rules and traces which are not motivated in Turkish.

In summary, the advantages of using a Combinatory Categorial formalism are that:

- CCGs provide a compositional and flexible surface structure, which allows syntactic constituents to easily correspond with information structure units.
- CCGs are lexicalist formalisms. This improves the processing efficiency since we only have to look at the grammatical information associated with each word in the input sentence, instead of all the information in the grammar. In addition, the same grammar can be used for both parsing and generation.
- CCGs are non-transformational grammars that do not posit empty categories or movement rules. Transformations greatly increase the power of grammar which may affect its computational efficiency (Peters and Ritchie, 1973), and the Turkish data does not provide sufficient evidence to support a syntactic movement operation.
- CCGs are mildly context-sensitive. (Shieber, 1985b) has shown that competence grammars for natural languages must be more powerful than context-free grammars. The class of mildly context-sensitive grammars are formally adequate for natural languages while still being polynomially parsable (Joshi, 1985).
- CCGs are computationally efficient (e.g. polynomially parsable) (Vijay-Shanker and Weir, 1993).

However, CCGs must be adapted to handle all the characteristics of "free" word order languages. In the next section, I discuss why we must extend CCGs, concentrating on the formal and descriptive power needed to capture long distance scrambling.

2.4 Why Extend CCGs?

As observed by (Becker, Joshi, and Rambow, 1991; Rambow, 1994a), scrambling is "doubly unbounded" in that:

- There is no bound on the distance over which an element may scramble, and
- There is no bound on the number of elements that can scramble in a sentence.

In general, this describes sentences like the following where each V_i subcategorizes for NP_i:

(17) $(NP_1 ... NP_n)_{scrambled}$ V_n ... V_1

The more one scrambles things, the harder the sentence is to process, but there is no clear cut-off point in which the scrambled sentences become ungrammatical for native speakers. Thus, I claim that processing limitations and pragmatic purposes, rather than syntactic competence, restrict such scrambling.

Following (Rambow, 1994a), I assume that there is no bound on the amount of scrambling in the syntactic competence grammar for "free" word order languages. This is similar to (Shieber, 1985b) where it is assumed that there is no bound on the number of clausal embeddings in the competence grammar in order to prove natural languages, specifically Swiss German cross-serial dependencies, require competence grammars that are more powerful than context-free grammars (CFGs). In fact, (Becker, Joshi, and Rambow, 1991; Rambow, 1994a) prove that Tree-Adjoining Grammars (TAGs) cannot derive unbounded long distance scrambling while maintaining the *co-occurrence constraint*, i.e. traditional lexical assignments where each tree associated with a verb in the lexicon subcategorizes only for the verb's own arguments. Since TAGs are weakly equivalent to CCGs (Weir, 1988), we would expect that CCGs cannot derive unbounded long distance scrambling as well.

In this section, I describe various versions of CCGs and their limitations in capturing aspects of "free" word order languages:

1. CCG without type-raising
2. CCG with type-raising
3. CCG with variable type-raising
4. CCG with variable type-raising and unbounded composition (B^n)

I will show that all but the last formalism are formally inadequate to capture *unbounded* long distance scrambling. However, I will show that this last formalism is not descriptively adequate to handle all aspects of "free" word order languages, because of problems with non-order-preserving type-raising. In the next chapter I will propose a formalism called Multiset-CCG which I argue is descriptively as well as formally adequate to handle "free" word order languages such as Turkish.

2.4.1 CCG without Type-raising

Verbal categories in CCGs traditionally indicate the linear order for their arguments. For instance, following traditional lexical assignments in CCG as in (Steedman, 1989), we could assign the functional category $(S\backslash NPnom)\backslash NPacc$ to a transitive verb in an SOV language to indicate that the verb must first find an accusative case-marked NP to its left and then a nominative case-marked NP to its left to result in a complete sentence. This category would rule out any word order other than SOV for a transitive sentence and thus, could not handle free word order. However, it is possible to assign multiple categories to each verb to match any ordering of the arguments within one clause. For instance, we could assign the following set of categories to a transitive verb in order to handle all six permutations of the verb and its arguments:

(18) a. (S\NPnom)\NPacc, for the SOV word order.
b. (S\NPacc)\NPnom, for the OSV word order.
c. (S/NPacc)\NPnom, for the SVO word order.
d. (S/NPnom)\NPacc, for the OVS word order.
e. (S/NPacc)/NPnom, for the VSO word order.
f. (S/NPnom)/NPacc , for the VOS word order.

In fact, we could even associate pragmatic functions with each sentence position, so that the argument in the sentence-initial position is interpreted as the topic of the sentence, and the argument in the immediately preverbal position is interpreted as the focus of the sentence, as seen below.

(19) a. S\Nnom:topic(X)\Nacc:focus(Y)
b. S\Nacc:topic(Y)\Nnom:focus(X)
c. S/Nnom:ground(X)\Nacc:focus(Y)
d. S/Nacc:ground(Y)\Nnom:focus(X), ...

However, this approach does not capture the information structure of all scrambled sentence in Turkish. As seen in Chapter 2, adjuncts and elements from other clauses can take part in the information structure of a clause even though they are not arguments in the predicate-argument structure of that clause. In this approach, there is no way to allow an adjunct to be the topic or the focus of the sentence unless it is made an argument of the verb.

Although this approach captures the local scrambling of arguments within one clause, it cannot handle long distance scrambling at all (without using type-raised categories, which will be discussed in the next section). In the sentence in (21)a, the arguments of the two clauses are interleaved; the subject of the embedded clause marked by the genitive case morpheme occurs in the sentence-initial position of the main clause. As seen in the CCG derivation in (21)b, we can

combine the two verbs together via the backward composition rule, repeated in (20), but then this complex verbal category cannot combine with the NP arguments because it expects them in the opposite order.

(20) **Backward Composition (<B):**

$Y\backslash Z \quad X\backslash Y \Rightarrow X\backslash Z$

(21) a. Ayşe'nin ben gittiğini sandım.
Ayşe-Gen I go-Ger-Acc think-Past-1S.
"As for Ayşe, I thought that she had left."

b.
Ayşe-Gen I		go-Ger-Acc	think-Past-1S.
$NPgen_2$	$NPnom_1$	$S_{ger}\backslash NPgen_2$	$S\backslash NPnom_1\backslash S_{ger}$
		——————————<B	
		$S\backslash NPnom_1\backslash NPgen_2$	
——————XXX——————			

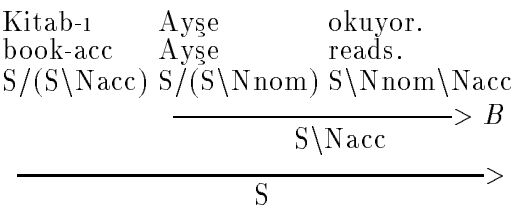
We could imagine assigning a category such as $S\backslash NPgen_2\backslash NPnom_1\backslash S_{ger}$ to the matrix verb; this category orders the subject of the embedded verb with respect to the subject of the matrix verb. However, it does not make sense for the category of the matrix verb to specify all of the arguments of embedded clauses. If we extended this approach to sentences with more than one embedded clause, then we would have to assign a different category to the matrix verb for each word order with two clauses, with three clauses, and in fact an unbounded number of embedded clauses. This approach would be unreasonable. The competence grammar should have a finite lexicon and should capture the recursiveness in language through the grammar rules.[6] Thus, CCGs with traditional lexical assignments and without type-raised categories are formally inadequate to capture long distance scrambling. In the following section, I will discuss type-raised categories in CCGs and how they can be used to handle some of the long distance dependencies that we need to capture for "free" word order languages.

2.4.2 CCGs with Type-raising

Traditionally, CCGs handle some long distance dependencies such as wh-questions and relative clause formations through the use of type-raising and composition (Steedman, 1985; Steedman, 1987; Steedman, 1989). The same method can potentially derive different word orders in Turkish. By representing NPs as type-raised categories, we can derive scrambled sentences, like the OSV sentence below, in which the NPs do not occur in the order that the verb specifies. In fact, we may want to think of the case morphemes triggering type-raising rules in the lexicon which convert a basic category such as NP into a function looking for a verb that is looking for the case-marked

[6] In fact, in traditional lexical assignments in CCGs, lexical categories only refer to categories in same predicate-argument structure. And this constraint is identical to the *co-occurrence constraint* on trees in TAGs (Becker, Joshi, and Rambow, 1991), which states that each clausal tree contains a verb and only its own arguments.

NP e.g. $S/(S\backslash Nnom)$.

(22)


However, we will need to assign each NP more than one type-raised category in the lexicon in order to handle all scrambled sentences. For example, an object NP with accusative case, $Nacc$, will need to be type-raised to the following categories to handle all of the six permutations possible for a transitive sentence:

(23) a. $S\backslash Nnom/(S\backslash Nnom\backslash Nacc)$, for the SOV and OSV word orders,

b. $S/(S\backslash Nacc)$, for the OSV word order,

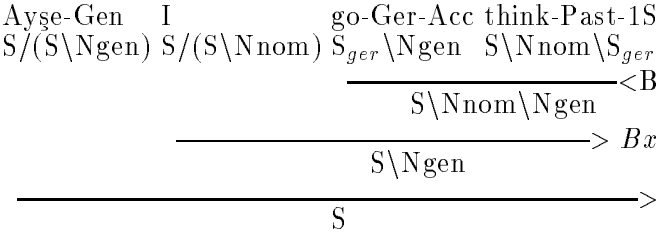
c. $S\backslash(S\backslash Nacc)$, for the SVO and VSO word orders,

d. and $S\backslash Nnom\backslash(S\backslash Nnom\backslash Nacc)$, for the VOS word order.

Once we consider ditransitive sentences, the number of type-raised categories required for each NP grows even larger. This causes some concern for efficient parsing and generation. One way to solve this problem is to use a variable in the type-raised categories, as will be discussed on page 32. An additional problem is that non-order-preserving type-raising is needed in Turkish as seen in (23)c and d in order to handle post-verbal scrambling, and this causes overgeneration problems that will be discussed further in the next section. In this section, I will discuss whether CCGs with type-raising (without variables) can handle unbounded long distance scrambling.

The use of composition and type-raising (without variables) in CCGs allows an analysis for *limited* long distance scrambling in "free" word order languages. For example, we can derive the following long distance scrambled sentence by combining the two verbs together via the backward composition rule and then allowing the type-raised NPs to combine with that complex verb.

(24) a. Ayşe'nin ben gittiğini sandım.
Ayşe-Gen I go-Ger-Acc think-Past-1S.
'As for Ayşe, I thought that she had left.'

b.


However, CCG cannot derive all sentences with long distance scrambling with traditional lexical assignments. For example the sequence "NP_1 NP_2 NP_1 V_2 V_1" cannot be derived because

the arguments of each clause are interleaved.

(25) A: San-a kim gel-diğ-im-i söyle-di?

You-dat who come-Ger-1S-Acc say-Past?

"Who told you that I came here?"

a. Ban-a, sen-in$_i$, kimse [e$_i$ gel-diğ-in-i] söyle-me-di.

I-dat you-Gen$_i$ noone [e$_i$ come-Ger-2S-Acc] say-Neg-Past.

"As for me, and as for you, NOONE told me that you came."

b.

I_1	you_2	$noone_1$	$come_2$	say_1
$S/(S\backslash Ndat)$	$S/(S\backslash Ngen)$	$S/(S\backslash Nnom)$	$S_{ger}\backslash Ngen$	$S\backslash Nnom\backslash Ndat\backslash S_{ger}$

$$\underline{S_{ger}\backslash Ngen \quad S\backslash Nnom\backslash Ndat\backslash S_{ger}}\ {<}B$$
$$S\backslash Nnom\backslash Ndat\backslash Ngen$$
$$\underline{S/(S\backslash Nnom) \quad S\backslash Nnom\backslash Ndat\backslash Ngen}\ {>}Bx$$
$$S\backslash Ndat\backslash Ngen$$
$$\text{———————XXX———————}$$

This derivation does not go through because the composition rules preserve the order of the arguments specified in the verbal categories, however the arguments of each clause do not occur in this order in the string; they are interleaved. For this derivation to be possible the genitive subject of the embedded clause would have to have the type-raised category $S\backslash Ndat_1/(S\backslash Ndat_1\backslash Ngen_2)$. However, this means that it would have to know about an NP which is not in the same clause, and that there would be an infinite number of type-raised categories that would have to be assigned in order to handle all the word order variations in sentences with an unbounded number of embedded clauses.

The scrambled sequence "NP_1 NP_2 NP_1 V_2 V_1" is also grammatical in German, as shown in the following example from (Becker, Joshi, and Rambow, 1991; Rambow, 1994a).

(26) ...dem Kunden den Kühlschrank$_i$ bisher noch niemand

..the client the refrigerator$_i$ so far as yet noone

[e$_i$ zu reparieren] zu versuchen versprochen hat.

[e$_i$ to repair] to try promised has.

"..that so far, noone has yet promised the client to repair the refrigerator."

They show that TAGs with traditional lexical assignments cannot handle this sequence. Since TAGs and CCGs are weakly equivalent, it is not surprising that CCGs with traditional lexical assignments cannot handle this sequence either. Using type-raising and composition, CCGs can handle local scrambling, long distance scrambling with one embedded clause, and long distance scrambling which does not involve the interleaving of arguments from more than one clause, but it cannot handle all word order variations in sentences with an arbitrary number of embedded clauses unless variable type-raising and unbounded composition are added to the grammar; this is discussed further in the next section.

2.4.3 CCGs with Variable Type-Raising

(Steedman, 1985) introduces variables in type-raised categories to handle recursive embedded clauses in Dutch. Since there are many verbal functions which subcategorize for an NP argument, a general type-raised category $v/(v\backslash NP)$ with a variable v ranging over a set of verbal functions captures many generalities. It is not clear whether this use of variables increases the power of CCGs; in (Hoffman, 1993), I present some evidence that a similar use of variables may increase the weak generative power of CCGs. In any case, variables do increase the domain of locality for type-raised categories. Thus, CCGs with variable type-raising can handle more cases of long distance scrambling.

In fact, in the CCG for Dutch (Steedman, 1985; Steedman, 1993), the grammar has to be restricted to disallow such scrambled sentences. Type-raising causes the Dutch grammar to overgenerate certain sentences with a mixed order of NPs. Although Dutch does allow some scrambling of NP arguments (den Besten, 1985), NP arguments belonging to different verbs in the cross-serial constructions cannot be easily scrambled. If the grammar rules are not restricted[7], the use of type-raised categories causes the following ungrammatical sentence to be generated:

(27) *..dat de kinderen$_2$ Jan$_1$ zag$_1$ zwemmen$_2$.

$$\dfrac{\dfrac{\dfrac{v/(v\backslash NP_2) \quad v/(v\backslash NP_1) \quad S\backslash NP_1\backslash NP_2/(S\backslash NP_2)}{(S\backslash NP_1)\backslash NP_2}>}{S\backslash NP_2}>Bx \quad S\backslash NP_2}{S}>$$

For "free" word order languages such as Turkish, we can use the type-raised category $v/(v\backslash NP)$ to handle long distance scrambling to the left of the verbs, even for sequences with interleaved arguments such as "NP_1 NP_2 NP_1 V_2 V_1" that CCGs without variable type-raising cannot handle.

(28) a. Ban-a sen-in$_i$ kimse [e$_i$ gel-diğ-in-i] söyle-me-di.
I-dat you-Gen$_i$ noone [e$_i$ come-Ger-2S-Acc] say-Neg-Past.
"As for me, and as for you, NOONE told me that you came."

[7](Steedman, 1993) restricts the crossing composition rule in the Dutch grammar by stating that the primary constituent X/Y in the rule cannot be bound to a category where $X = T$ as in the type-raised function $T/(T\backslash NP)$. Thus, in the derivation of the scrambled sentence in (27), the use of the forward crossing composition rule is blocked because $X =$ the variable T.

b. $\underset{v/(v\backslash \text{Ndat})}{\text{I}_1}$ $\underset{v/(v\backslash \text{Ngen})}{\text{you}_2}$ $\underset{v/(v\backslash \text{Nnom})}{\text{noone}_1}$ $\underset{S_{ger}\backslash \text{Ngen}}{\text{come}_2}$ $\underset{S\backslash \text{Nnom}\backslash \text{Ndat}\backslash S_{ger}}{\text{say}_1}$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{S\backslash \text{Nnom}\backslash \text{Ndat}\backslash \text{Ngen}}<\text{B}}{S\backslash \text{Ndat}\backslash \text{Ngen}}>B}{S\backslash \text{Ndat}}>\quad (v = S\backslash \text{Ndat})}{S}>$$

Note that the variable in the type-raised category for the embedded verb's subject "you" is bound during the derivation to a verbal function that refers to an NP that is not in the same predicate-argument structure. Since the variables in the type-raised categories can bind verbal functions during the derivation that refer to arguments that are not in the same clause, we can handle long distance scrambling with interleaved arguments while maintaining a finite lexicon where each NP is only associated with the one type-raised category $v/(v\backslash NP)$. Non-order preserving categories such as $v\backslash(v\backslash NP)$ that are needed to handle scrambling to the right of the verb will be discussed in the next sections.

However, even CCGs with variable type-raising cannot handle *unbounded* long distance scrambling. For example, CCGs with variable type-raising cannot derive a string of an unbounded number of scrambled NPs followed by a string of verbs "$(\text{NP}_1 ... \text{NP}_n)_{scrambled}$ V_n ... V_1" where each verb, V_i, subcategorizes for NP_i unless the composition rules are extended.

2.4.4 Adding unbounded composition (B^n)

CCG can only derive the string above if it uses unbounded composition (B^n). This is known to increase the power of CCGs (Weir, 1988). To derive the string above, we first need to combine the verbs together, using backward composition, which results in a complex verb category looking for all of the NPs; next, we need to combine each NP with this complex verb category. However, unbounded composition is necessary to combine any type-raised NP_i with the complex verb regardless of the order specified by the complex verb, as seen below. The variable in the type-raised category binds a verbal category of unbounded size, e.g. $(v = S\backslash Np_1\backslash ...\backslash Np_{i-1})$.

$$\dfrac{v/(v\backslash Np_i) \qquad S\backslash Np_1\backslash Np_2 ...\backslash Np_i ...\backslash Np_m}{S\backslash Np_1\backslash ...\backslash Np_{i-1}\backslash Np_{i+1} ...\backslash Np_m}>\text{Bx}^n$$

CCGs cannot capture unbounded scrambling without resorting to the use of variable type-raising and the more powerful generalized composition rule (B^n) unbounded composition. Thus, we can confirm the findings of (Becker, Joshi, and Rambow, 1991) that unbounded scrambling requires

a greater weak generative capacity than what the weakly-equivalent formalisms CCG, TAG, and LIG can provide.

Even though CCGs using variable type-raising and B^n can handle unbounded long distance scrambling, I argue that they are not descriptively adequate to handle other aspects of "free" word order languages. They cannot account for post-verbal scrambling or coordination in scrambled sentences without using non-order preserving type-raising. And as we will see in the next section, non-order preserving type-raising can cause grammars to overgenerate unwanted word orders. Thus, I propose that the strict ordering of arguments in the verbal categories should be relaxed.

The order-preserving type-raised categories $v/(v\backslash NP)$ or $v\backslash(v/NP)$ cannot handle post-verbal scrambling in Turkish. What is needed is a non-order preserving type-raised category $v\backslash(v\backslash NP)$ or a verbal category that does not specify that its argument should be to its left, i.e. $S|Nnom$.

(29) a. Uyan-dı Ayşe.
awake-Past Ayşe.
"She, Ayşe, woke up."

b. Awake Ayşe.
S\Nnom $v/(v\backslash Nnom)$
———XXX———

Order preserving type-raising also cannot handle gapping and coordination in Turkish where the NPs are scrambled. Coordination and gapping in CCGs is handled by allowing the type-raised NPs to form a syntactic constituent through the composition rules. However, if the NPs are scrambled, the constituent they form cannot then combine with the verb which specifies a strict ordering of its arguments. In the derivation below, the variable binding in the composition rule is $(v_2 = v_1\backslash Nacc)$.

(30) a. Kitab-ı Ayşe, gazete-yi de Fatma oku-yor.
Book-Acc Ayşe, newspaper-Acc too Fatma read-Prog.
"As for the book, Ayşe is reading it, and the newspaper, Fatma."

b. Book Ayşe , newspaper Fatma read.
$v_1/(v_1\backslash Nacc)$ $v_2/(v_2\backslash Nnom)$ and $v/(v\backslash Nacc)$ $v/(v\backslash Nnom)$ S\Nnom\Nacc

$$\dfrac{\dfrac{}{v_1/(v_1\backslash Nacc\backslash Nnom)}>B \qquad \dfrac{}{v/(v\backslash Nacc\backslash Nnom)}>B}{\dfrac{}{v/(v\backslash Nacc\backslash Nnom)}(\&)}$$

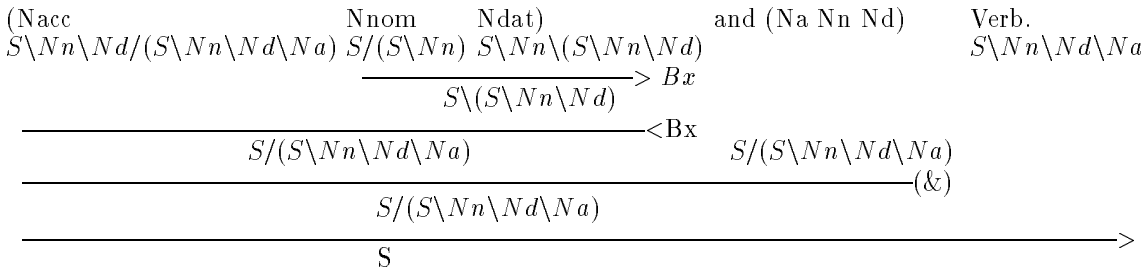
———XXX———

In the next section, I show how non-order preserving type-raising can handle post-verbal scrambling and gapping constructions, but I also show that there are many problems with using non-order preserving type-raising. The greatest problem is for languages like Korean which allow scrambling but are strictly verb-final. Non-order preserving type-raising overgenerates word orders involving post-verbal constituents for Korean. The root of these problems is that a verbal category such as $S\backslash Nn\backslash Nd\backslash Na$ specifies its NP arguments in a strict order and direction. Thus, I argue

that the strict order of NP arguments in a verbal category such as $S\backslash Nn\backslash Nd\backslash Na$ needs to be relaxed instead of generalizing the type-raising scheme for NPs.

2.4.5 Problems with Non-Order Preserving Type-raising

(Lee and Niv, 1989) develop a CCG analysis for scrambling in Korean which captures many sentences in which constituents have been scrambled and coordinated by using non-order preserving type-raising. Their approach does not use variables in type-raised categories. The non-order preserving type-raising is needed, for instance, to compose three scrambled NPs together that can be coordinated with another group of NPs before combining with the verb. The NPs are not in the order that the verb expects them to be, but non-order preserving type-raising can be used to create a constituent that matches the order of the arguments in the verb's category.



Notice that the NP with dative case ($Ndat$) is type-raised to a category that is not order-preserving. This is necessary because the verb has the category, $S\backslash Nn\backslash Nd\backslash Na$, with both the order and the directionality of its arguments specified, and the three NPs must combine together in such a way to form a functor category with the arguments in the correct order and directionality to exactly match the lexical category of the verb.

CCG with non-order preserving type-raising can provide an analysis of most gapping constructions in "free" word order languages. However, there are certain sequences of coordinated NPs this grammar cannot handle. For example, the derivation for a sequence such as in (31) is blocked because the type-raised NPs must combine together in a certain order to match the ordering of arguments specified by the verb.

(31) (Nacc Nnom) and (Nacc Nnom) Ndat Verb.

In (Lee and Niv, 1989), they note that a more general type-raising scheme is needed to handle coordination with scrambled NPs. For example, a dative marked NP needs to be assigned multiple type-raised categories (e.g. $S|(S\backslash Nd)$, $S\backslash Nn|(S\backslash Nn\backslash Nd)$, etc.) in order to handle all strings of coordinated NPs. However, this use of multiple types leads to an increase in the number of possible derivations for a sentence.

Another problem with this approach is that the use of a non-order preserving category such as $S\backslash(S\backslash NP)$ does not match our intuitions about what the direction slash means. In the example

above, the non-order preserving category $S\backslash Nn\backslash(S\backslash Nn\backslash Nd)$ is assigned to the dative marked NP. Although this type indicates that the NP should be looking for its verb, $S\backslash Nd$, on its left, it eventually finds and combines with its verb on the right. The directionality indicated in the category does not match the the placement of the NP and verb in the string. I disagree with the use of non-order preserving type-raising because I believe we should maintain the intuitive meaning of the slash in lexical categories.

The most serious problem with using non-order preserving type-raising is the danger of over-generation. In strictly verb-final languages like Korean, NPs generally cannot occur behind the verb. However, in this analysis, the NPs must be able to type raise to order-preserving and non-order preserving categories in order to handle scrambling in just the positions to the left of the verb, and thus we cannot restrict NPs from occurring post-verbal positions by restricting what kind of type-raised category they are assigned. The only way to keep this grammar from over-generating these sentences is by stipulating categorial restrictions on the backward composition and application rules.

At first glance, the last problem does not seem to apply to Turkish because generally Turkish nouns can occur behind the verb. However, question words in Turkish (as well as discourse-new elements) cannot occur behind the verb although they can scramble freely to positions to the left of the verb. Thus, certain characteristics of the noun can determine whether or not it is free to occur in all sentence positions. This suggests that the restriction on whether an NP can occur to the right of the verb is a lexical one, and I propose that we capture this restriction in the lexicon. However, a grammar like the one above places this restriction in the combinatory rules, instead of capturing it in the lexical entries of the NPs.

Given the problems with non-order preserving type-raising, I argue that "free" word order should not be captured by generalizing the type-raising scheme for NPs but by relaxing the strict specification of argument order in the categories assigned to the verbs.

2.5 Summary

In Section 1, I presented the data on Turkish word order. This data motivates the need for a formalism which captures the following characteristics of "free" word order languages:

- the free word order of arguments and adjuncts a clause,
- the long distance scrambling of elements from embedded clauses into the matrix clause,
- the syntactic restrictions on word order (e.g. islands, head-final clauses),

- the context-dependent interpretations associated with certain sentence positions (e.g. topic with the sentence-initial and focus with the immediately preverbal positions in Turkish),
- the recursive nature of information structures (embedded ISs in complex sentences),
- the ability of elements that are not arguments in the predicate-argument structure of a clause to take part in the information structure of the clause (e.g. adjunct scrambling and long distance scrambling).

After discussing previous approaches to "free" word order in Section 2, I discussed why I have chosen to adapt Combinatory Categorial Grammars (CCGs) to handle these characteristics of "free" word order languages in Section 3. CCG is a computationally attractive formalism in that it is a lexicalist, non-transformational, mildly context-sensitive, and polynomially parsable formalism. The formalism that I develop for Turkish "free" word order preserves these computational qualities. In addition, CCGs has a flexible notion of syntactic constituency. The surface structure derived by CCGs can directly reflect the information structure of sentences. We can integrate information structure with syntax in a CCG-based formalism to capture both the syntax and the context-dependent interpretation of "free" word order.

In Section 4, I showed that CCGs must be extended to provide a uniform analysis of local and long distance scrambling. I described various versions of CCGs and their limitations in capturing characteristics of "free" word order languages.

- CCGs with traditional lexical assignments and without type-raised categories are formally inadequate to capture long distance scrambling.
- CCGs with order-preserving type-raising and composition can handle local scrambling, long distance scrambling with one embedded clause, and long distance scrambling which does not involve the interleaving of arguments from more than one clause, but they cannot handle unbounded scrambling where an unbounded number of elements can be extracted and scrambled an unbounded distance away from their clause.
- CCGs with variable type-raising and unbounded composition ($> B^n$) can handle unbounded long distance scrambling, but they are not descriptively adequate for "free" word order languages. They cannot account for post-verbal scrambling or coordination in scrambled sentences without using non-order preserving, and there are many overgeneration problems with using non-order preserving type-raising.

I argue that instead of generalizing the type-raising scheme for NPs, we need to relax the strict order of NP arguments in a verbal category such as $S\backslash Nnom\backslash Nacc$. In the next chapter, I present

a categorial formalism, Multiset-CCG, where verbs subcategorize for a multiset of arguments without specifying their relative ordering. I argue that this formalism is formally and descriptively adequate in handling free word order as well as the appropriate restrictions on word order. I prove that this new formalism retains the computationally-attractive properties of CCG in Chapter 4: it is a lexicalist, non-transformational, mildly context-sensitive, and polynomially parsable formalism. And in Chapter 6, I integrate information structures, determined for Turkish in Chapter 5, with Multiset-CCG to capture the context-dependent interpretation of word order in Turkish.

Chapter 3

A Categorial Syntax for Turkish

In this chapter, I present a competence grammar for a fragment of Turkish that captures the basic syntactic and semantic relationships between predicates and their arguments while allowing "free" word order. This grammar, which derives a *predicate-argument structure*, will then be integrated with an *information structure*, capturing pragmatic information such as topic and focus, in Chapter 6.

I present a formalism called Multiset Combinatory Categorial Grammars (Multiset-CCGs) that can capture the syntax of languages with freer word order than English (Hoffman, 1992; Hoffman, 1995). Multiset-CCGs relaxes the subcategorization requirements of a predicate such that it requires a set of arguments without specifying their order. This formalism is based on Combinatory Categorial Grammars, CCGs, (Ades and Steedman, 1982; Steedman, 1985; Steedman, 1987; Steedman, 1989), a lexicalist and compositional grammar in which syntactic and semantic parallelism is maintained. An introduction to CCGs has already been given in the introduction; in addition, I have already discussed why CCGs must be extended in order to handle "free" word order languages in the previous chapter.The compositionality and flexibility in structure that CCGs provide are very advantageous for my approach to capture "free" word order. These properties allow a uniform approach in handling local and long distance scrambling. In addition, they allow us to easily integrate discourse information into the grammar which makes the grammar very useful in a computational approach as will be seen in Chapter 6.

3.1 Local Scrambling

As we saw in the last chapter, the arguments of a verb in Turkish (as well as many other "free" word order languages) do not have to occur in a fixed word order. All six permutations of this transitive sentence have the same propositional interpretation *see(Ayşe,Fatma)*.

(1) a. Ayşe Fatma'yı gördü.
Ayşe Fatma-Acc see-Past-(3Sg).
"Ayşe saw Fatma."
b. Fatma'yı Ayşe gördü.
c. Ayşe gördü Fatma'yı.
d. Fatma'yı gördü Ayşe.
e. Gördü Fatma'yı Ayşe.
f. Gördü Ayşe Fatma'yı.

To capture the "free" word order of arguments in a clause, Multiset CCGs relaxes the linear ordering information in the subcategorization specifications of the verbs.[1] In Multiset-CCGs, each verb is assigned a function category in the lexicon which specifies a *multiset* of arguments, so that it can combine with its arguments in any order. For instance, a transitive verb has the following category $\boldsymbol{S|\{Nn, Na\}}$ which defines a function looking for a set of arguments, a nominative case noun phrase (Nn) and an accusative case noun phrase (Na), and resulting in the category S, a complete sentence, once it has found these arguments.

The syntactic category for verbs provides no hierarchical or precedence information. However, it is associated with a propositional interpretation that does express the hierarchical ranking of the arguments. For example, the verb "see" is assigned the lexical category in (2).

(2) $\boldsymbol{S : see(X, Y)|\{Nn : X, Na : Y\}}$

A proper noun such as "Fatma" is assigned $\boldsymbol{Nn : Fatma}$, where the semantic interpretation is separated from the syntactic representation by a colon. These categories are a shorthand for the many syntactic and semantic features associated with each lexical item. The verbal functions can also specify a *direction* feature for each of their arguments, following (Zeevat, Klein, and Calder, 1987) who first introduced direction as a property of arguments. Verb-final languages such as Korean can be modeled by using this direction feature in verbal categories, notated as an arrow above the argument e.g. $S|\{\overleftarrow{Nn}, \overleftarrow{Na}\}$.

The lexical categories above can easily be transformed into a DAG (directed acyclic graph), also called feature-structure or attribute-value matrix (Shieber, 1985a; Johnson, 1988)), representation like the following where coindices, x and y, are indicated by italicized font. Feature structures and unification has been used in other categorial formalisms as well, (Wittenburg, 1986; Zeevat, Klein, and Calder, 1987; Karttunen, 1989).

[1] This is similar to the approach of (Gunji, 1987; Karttunen, 1989), as will be discussed further in the next chapter.

$$\left[\begin{array}{ll} \text{Result} : & \left[\begin{array}{ll} \text{Syn} : & \left[\begin{array}{ll} \text{Cat} : & \text{S} \\ \text{Tense} : & \text{Past} \\ \text{Agr} : & \left[\begin{array}{ll} \text{Number} : & n \\ \text{Person} : & 3 \end{array}\right] \end{array}\right] \\ \text{Sem} : & \text{see}(x,y) \end{array}\right] \\ \text{Args} : & \left\{ \left[\begin{array}{ll} \text{Syn} : & \left[\begin{array}{ll} \text{Cat} : & \text{np} \\ \text{Case} : & \text{nom} \\ \text{Agr} : & \left[\begin{array}{ll} \text{Number} : & n \\ \text{Person} : & 3 \end{array}\right] \end{array}\right] \\ \text{Sem} : & x \end{array}\right], \left[\begin{array}{ll} \text{Syn} : & \left[\begin{array}{ll} \text{Cat} : & \text{np} \\ \text{Case} : & \text{acc} \end{array}\right] \\ \text{Sem} : & y \end{array}\right] \right\} \end{array}\right]$$

Multiset-CCG contains a small set of rules to combine these categories to create larger constituents. The following application rules allow a function such as a verbal category to combine with one of its arguments to its right (>) or left (<). The functions can specify a *direction* feature for each of their arguments, notated in the rules as an arrow above the argument.[2] We assume that a category $X|\{\ \}$ where there are no arguments left in the multiset rewrites by a clean-up rule to just X.

(3) a. **Forward Application (>):**

$X|(Args \cup \{\vec{Y}\})$ $\quad Y \quad \Rightarrow \quad X|Args$

b. **Backward Application (<):**

$Y \quad X|(Args \cup \{\overleftarrow{Y}\}) \quad \Rightarrow \quad X|Args$

Using these new rules, a verb can apply to its arguments in any order. For example, the following is a derivation of a sentence with the word order Object-Subject-Verb: For example, the following is a derivation of a transitive sentence with the word order Object-Subject-Verb; variables in the semantic interpretations are italicized.[3]

(4)

Fatma'yı	Ayşe	gördü.
Fatma-Acc	Ayşe	saw.
Na:Fatma	Nn:Ayşe	S: see(X, Y)|{ Nn:X, Na:Y}

S:see(Ayşe, Y)|{ Na:Y} <

S: see(Ayşe, Fatma) <

[2]Since Turkish is not strictly verb-final, most verbs will not specify the direction features of their arguments and can match either function in the application rules. The direction feature is unified in when the rules are applied.

[3]In my implementation of this grammar, DAG-unification (Shieber, 1985a) instead of term-unification is used in the rules. To improve the efficiency of unification and parsing, the arguments DAGS can be associated with feature labels that indicate their category and case, e.g. { N(nom) : DAG1, N(acc) : DAG2 }.

In fact, all six permutations of this sentence can be derived by the Multiset-CCG rules and are assigned the same propositional interpretation, *see(Ayşe,Fatma)*.

Local scrambling also occurs in embedded clauses in Turkish. The arguments and adjuncts within an embedded clause can also occur in almost any order. The sentences below demonstrate the local scrambling of a gerundive clause within the matrix sentence and the local scrambling of arguments within the embedded clause.

(5) a. Ayşe [kedi-yi Fatma'nin sev-diğ-i-ni] bil-iyor.
Ayşe [cat-Acc Fatma-Gen like-Ger-3S-Acc] know-Pres.
"Ayşe knows that as for the cat, Fatma likes it."

b. [Fatma'nin kedi-yi sev-diğ-i-ni] Ayşe bil-iyor.
[Fatma-Gen cat-Acc like-Ger-3S-Acc] Ayşe know-Pres.
"As for Fatma's liking the CAT, it's AYŞE who knows that."

c. Ayşe biliyor [Fatma'nin sevdiğini kediyi].
Ayşe know-Pres [Fatma-Gen like-Ger-3S-Acc cat-Acc].
"Ayşe knows that, that Fatma likes it, the cat."

Table 3.1 summarizes the scrambling behavior of embedded argument clauses in Turkish; this is based on (Erguvanlı, 1984; Hoffman, 1991; Kural, 1993). The types of subordinate verbs according to their morphology have already been described in the introduction, page 6. Subordinate verbs can occur with or without the tense morphemes, with or without subject-agreement morphemes, and with or without case-markings. For example, in (5), the subordinate verb is marked with the gerundive morpheme -dik which is given a past or present tense reading; it occurs with agreement markings and is always assigned accusative case if it is the direct object of the sentence.

In Table 3.1 under the heading *Scrambling Behaviour*, the first column refers to whether constituents of the embedded clause can occur in any order within the embedded clause. The second column refers to whether the clause as a whole can occur anywhere within the matrix sentence. As can be seen, this is determined by whether the clause is case-marked or not. Clauses that are case-marked can occur anywhere within the matrix sentence, while those without case-marking are restricted to the immediately preverbal positions (this will be discussed further in later sections). The final column refers to long distance scrambling of constituents out of the embedded clause into the matrix sentence, which will be discussed further in the next section.

In Multiset-CCGs, subordinate verbs are assigned a category similar to matrix verbs. For example, $S_{ger-acc}|\{Ngen, Nacc\}$ is a subordinate verb that is marked with a gerundive morpheme and accusative case and subcategorizes for a genitive case-marked subject and an accusative case-marked object. This category allows the arguments of a subordinate verb to occur in any order

Types of Embedded Clauses					Scrambling Behavior		
Type	morph	tense	agr	case	within clause	in matrix S	long distance
Gerundive	(-dik)	past/pres	+agr	case	Yes	Yes	Yes
	(-ecek)	future	+agr	case	Yes	Yes	Yes
Infinitive	(-mek)	-tense	-agr	case	Yes	Yes	Yes
	(-mek)	-tense	-agr	none	verb final	No	Yes
	(-me)	-tense	+agr	case	Yes	Yes	Yes
Complement	$\emptyset$	+tense	-agr	none	Yes	No	Yes
	$\emptyset$	+tense	+agr	none	Yes	No	Yes

Table 3.1: Scrambling Behaviour of Embedded Clauses

within the subordinate clause. The verb-final behaviour for infinitive clauses is captured by specifying leftward direction arrows on the arguments of infinitive verbs. In DAG notation, a subordinate verb such as "sevdiğini" is assigned the following category:

$$\begin{bmatrix} \text{Result} : \begin{bmatrix} \text{Syn} : \begin{bmatrix} \text{Cat} : \text{S}_{ger} \\ \text{Case} : \text{Acc} \\ \text{Tense} : \text{Pres} \\ \text{Agr} : \begin{bmatrix} \text{Number} : n \\ \text{Person} : 3 \end{bmatrix} \end{bmatrix} \\ \text{Sem} : \text{like}(x,y) \end{bmatrix} \\ \text{Args} : \left\{ \begin{bmatrix} \text{Syn} : \begin{bmatrix} \text{Cat} : \text{np} \\ \text{Case} : \text{Gen} \\ \text{Agr} : \begin{bmatrix} \text{Number} : n \\ \text{Person} : 3 \end{bmatrix} \end{bmatrix} \\ \text{Sem} : x \end{bmatrix}, \begin{bmatrix} \text{Syn} : \begin{bmatrix} \text{Cat} : \text{np} \\ \text{Case} : \text{Acc} \end{bmatrix} \\ \text{Sem} : y \end{bmatrix} \right\} \end{bmatrix}$$

Instead of using the set notation, we could imagine assigning Turkish verbs multiple lexical entries, one for each possible word order permutation; for example, a transitive verb could be assigned the curried categories $S\backslash Nn\backslash Na$, $S\backslash Na\backslash Nn$, $S\backslash Na/Nn$, etc., instead of the one entry $S|\{Nn, Na\}$. However, the set notation is more than a shorthand representing multiple entries because it allows us to handle long distance scrambling, where arguments of an embedded clause are placed in matrix clause positions.

3.2 Long Distance Scrambling

As seen in Table 3.1, long distance scrambling is allowed out of all embedded argument clauses in Turkish, whether or not the embedded verb is marked with the tense, agreement, or case morphemes. Examples are shown below of long distance scrambling to the sentence-initial and post-verbal positions for every type of clause seen in Table 3.1:

(6) a. Ayşe'nin ben [gel-diğ-i-ni] bil-iyor-um ev-e.
Ayşe-Gen I [come-Ger-3S-Acc] know-PresProg-1S house-Dat.
"As for Ayşe, I know that she came home."

b. Ayşe'nin ben [gel-eceğ-i-ni] bil-iyor-um ev-e.
Ayşe-Gen I [come-Ger-3S-Acc] know-PresProg-1S house-Dat.
"As for Ayşe, I know that she will come home."

c. Sinema-ya Ahmet [PRO yarın git-me-yi] çok ist-iyor.
Movie-Dat Ahmet [PRO tomorrow go-Inf-Acc] very want-Prog.
"As for the movies, Ahmet wants to go to them tomorrow very much."

d. Sinema-ya Ahmet [PRO yarın git-mek] ist-iyor.
Movie-Dat Ahmet [PRO tomorrow go-Inf] want-Prog.
"As for the movies, Ahmet wants to go to them tomorrow."

e. Fatma'nın ben [git-me-si-ni] ist-iyor-um ev-e.
Fatma-Gen I [go-Inf-3S-Acc] want-Prog-1Sg house-Dat.
"As for Fatma, I want her to go home."

f. Ev-e Ali [ben git-ti-m] san-ıyor.
House-Dat Ali [I go-Past-1sg] think-Prog-(3SG).
"As for home, Ali thinks that I went there."

g. Ali beni [git-ti] san-ıyor ev-e.
Ali I-Acc [go-Past] think-Prog house-dat.
"Ali thinks of me that I went home."

In Multiset-CCGs, we represent Turkish subordinate verbs as functions similar to the matrix verbs. Two functions with multisets of arguments, e.g. two verbs or a verb and an adjunct, can combine using the following composition rules:

(7) a. **Forward Composition ($> B$):**
$X|(Args_X \cup \{\vec{Y}\}) \quad Y|Args_Y \;\Rightarrow\; X|(Args_X \cup Args_Y)$

b. **Backward Composition ($< B$):**
$Y|Args_Y \quad X|(Args_X \cup \{\overleftarrow{Y}\}) \;\Rightarrow\; X|(Args_X \cup Args_Y)$

These composition rules allow two verb categories with multisets of arguments to combine together. For example, the two verbs can syntactically and semantically combine together as shown in (8); the semantic interpretation of a category is given following the syntactic category and the colon.

(8) a. Fatma [Esra'nın kedi-ler-i sev-dig-i-ni] bil-iyor.
Fatma [Esra-Gen cat-pl-Acc like-Ger-Acc] know-Pres.
"Fatma knows that Esra likes cats."

b. sevdiğini — like-gerund-acc — $S_{ger} : like(y,z) \mid \{Ng : y, Na{:}z\}$
biliyor — knows — $S : know(x,p) \mid \{Nn{:}\ x, S_{ger}{:}\ p\}$

$$\frac{S_{ger} : like(y,z) \mid \{Ng : y, Na{:}z\} \quad S : know(x,p) \mid \{Nn{:}\ x, S_{ger}{:}\ p\}}{S : know(x,\ like(y,z)) \mid \{Nn : x, Ng : y, Na{:}z\}}\ <B$$

As the two verbs combine, their arguments collapse into one argument set in the syntactic representation. However, the verbs' respective arguments are still distinct within the semantic representation of the sentence. The predicate-argument structure of the subordinate clause is embedded into the semantic representation of the matrix clause.

Multiset-CCG can derive sentences with long distance scrambling without resorting to the use of empty categories or movement rules. The composition rules only combine *adjacent* and *linguistically realized* strings. Long distance scrambling can easily be handled by first composing the verbs together to form a complex verbal function which can then apply to all of the arguments in any order.

(9) a. Esra'nın Fatma [gittiğini] biliyor.
Esra-Gen Fatma [go-Ger-3sg-Acc] know-Pres.
"As for Esra, Fatma knows that she left."

b. Esra-gen Fatma go-ger-acc knows.
Ngen Nnom $S_{ger-acc}|\{Ngen\}$ $S|\{Nnom, S_{ger-acc}\}$

$$\frac{\dfrac{\dfrac{}{S|\{Ngen, Nnom\}}<B}{S|\{Ngen\}}<}{S}<$$

Note that sentence above cannot be derived using traditional CCG categories that indicate the linear order of their arguments.

(10) Esra-gen Fatma go-gerund-acc knows.
Ngen Nnom $S_{ger-acc}\backslash Ngen$ $S\backslash Nnom\backslash S_{ger-acc}$

$$\frac{\dfrac{}{S\backslash Nnom\backslash Ngen}<B}{XXX}$$

We must relax the order of the arguments in order to derive sentences with long distance scrambling. In fact, Multiset-CCGs can derive a string of any number of scrambled NPs followed by a string of verbs where each verb, V_i, subcategorizes for NP_i:

(11) $(\mathrm{NP}_1 ... \mathrm{NP}_m)_{scrambled}\ \mathrm{V}_m\ ...\ \mathrm{V}_1$

The Multiset-CCG composition rules allow the verbs to compose together to form a complex verbal constituent looking for all of the NP arguments in any order. As discussed in the previous chapter, the more one scrambles things, the harder the sentence is to process, but there is no clear cut-off point in which the scrambled sentences become ungrammatical for native speakers. Thus, I claim that processing limitations and pragmatic purposes, rather than syntactic competence, restrict such scrambling. Multiset CCG, as a competence grammar, must allow the scrambling of elements over an unbounded distance and the extraction of more than one element from embedded clauses.

The amount of scrambling in a sentence will affect its processing time. As we will see in the next chapter, Multiset-CCG is a grammar that is context-sensitive and polynomially parsable. The parsing time is primarily affected by the size of the multisets in derived categories, because the multisets can grow unboundedly through the use of the composition rules. The composition rules, which increase the size of the multiset of arguments, are more costly in terms of processing a sentence than the application rules, which decrease the size of the multiset. Notice that a sentence with no long distance scrambling like "Fatma thinks that as for the book, I read it" in (12)a is derived by just using application rules in Multiset-CCG.[4] However, if there is long distance scrambling in a sentence such as "As for the book, Fatma thinks that I read it." in (12)b, we must use the composition rules in the Multiset CCG derivation. As we compose verbs together (indicated by the rule $< B$), we create a derived category which contains a larger multiset of arguments to keep track of, and this is what increases the processing time.

(12) a.

Fatma	[kitab-ı	ben-im	oku-dğu-m-u]	san-ıyor.
Fatma	[book-Acc	I-Gen	read-Gerund-1S-Acc]	think-Prog.
Nnom	Nacc	Ngen	$S_{ger-acc}\|\{Ngen, Nacc\}$	$S\|\{Nnom, S_{ger-acc}\}$

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{Ngen}\quad \text{Nacc}}{S_{ger-acc}|\{Nacc\}}<}{S_{ger-acc}}<\quad \text{Nnom}}{S|\{Nnom\}}<}{S}<$$

[4] If there are sentential adjuncts in the sentence, we do use the composition rules, but the size of the multiset does not increase because the adjuncts only have one argument, e.g. $S|\{S\}$.

b. Kitabı Fatma okuduğumu sanıyor benim.
book-Acc Fatma read-gerund-acc thinks I-Gen.
Nacc Nnom $S_{ger-acc}|\{Ngen, Nacc\}$ $S|\{Nnom, S_{ger-acc}\}$ Ngen

$$\frac{\frac{\frac{\frac{}{S|\{Nnom,Ngen,Nacc\}}<B}{S|\{Ngen,Nacc\}}<}{S|\{Ngen\}}<}{S}>$$

3.3 Adjuncts

Adjuncts can also occur in different sentence positions in Turkish sentences depending on their discourse function. For instance, the sentential adjunct "yesterday" can occur in different positions in a sentence, resulting in different discourse meanings, much as in English sentences:

(13) a. Fatma Ahmet'i dün gördü.
Fatma Ahmet-Acc dün see-Past.
"Fatma saw Ahmet YESTERDAY."

b. Dün Fatma Ahmet'i gördü.
Yesterday Fatma Ahmet-Acc see-Past.
"Yesterday, Fatma saw Ahmet."

c. Fatma Ahmet'i gördü dün.
Fatma Ahmet-Acc see-Past yesterday.
"Fatma saw Ahmet, yesterday."

In Multiset-CCG, sentential adjuncts are assigned the functor category $S|\{S\}$. Through the use of the composition rules, this category can combine with any functor which will also result in S a complete sentence. This allows the adjunct to occur almost anywhere in the sentence. For example,

(14) a. Ayşe dün geldi.
Ayşe dün came.
Nnom $S|\{S\}$ $S|\{Nnom\}$

$$\frac{\frac{}{S|\{Nnom\}}<B}{S}<$$

b. Ayşe geldi dün.
Ayşe came yesterday.
Nnom $S|\{Nnom\}$ $S|\{S\}$

$$\frac{\frac{}{S}<}{S}<B$$

This syntactic category is associated with a semantic interpretation like the following, where the argument S must contain an event variable e, and the result S must add a predicate modifying e to the semantic interpretation of the sentence:

(15)
$$\left[\begin{array}{lcl} \text{Result} & : & \left[\begin{array}{lcl} \text{Syn} & : & S \\ \text{Sem} & : & \text{yesterday}(e)\ \&\ P \end{array}\right] \\ \text{Args} & : & \left\{\left[\begin{array}{lcl} \text{Syn} & : & S \\ \text{Sem} & : & \left[\begin{array}{lcl} \text{Event} & : & e \\ \text{LF} & : & P \end{array}\right] \end{array}\right]\right\} \end{array}\right]$$

In sentences with more than one clause, the adjunct is attached to the closest clause. This predicts the correct semantic interpretation for the following sentences, where the adjunct "yesterday" can modify the going event and/or the saying event depending on its position in the sentence.

(16) a. Dün Ayşe [Ali'nin gittiğini] söyledi.
Yesterday Ayşe Ali-Gen go-Ger-Acc say-Past.
"Yesterday, Ayşe said that Ali left."
*"Ayşe said that yesterday Ali left."

b. Ayşe [Ali'nin dün gittiğini] söyledi.
Ayşe [Ali-Gen yesterday go-Ger-Acc] say-Past.
"Ayşe said that yesterday Ali left."
*"Yesterday, Ayşe said that Ali left."

c. Ayşe dün Ali'nin gittiğini söyledi.
Ayşe yesterday Ali-Gen go-Ger-Acc say-Past.
"Yesterday, Ayşe said that Ali left."
"Ayşe said that yesterday Ali left."

This approach cannot capture the proper semantic interpretation if adjuncts are long distance scrambled, because it attaches the adjunct to the closest clause. For example, in the following sentence, the predicate $place(E, Istanbul)$ must modify the going event instead of the knowing event.

(17) a. Istanbul'a Ayşe [Ali'nin gittiğini] biliyor].
Istanbul-loc Ayşe Ali-Gen go-Ger-Acc say-Past.
"As for Istanbul, Ayşe knows that Ali has gone there."

The Multiset-CCG derivation will not capture this reading because it will attach the adjunct "to Istanbul" to the matrix sentence instead of the embedded clause. However, the fact that knowing events do not take locative modifiers whereas going events do, as in the example above,

must be somehow encoded in the lexical entries. Some adjuncts act as if they are optional arguments and perhaps should be represented as such. I believe Multiset-CCG can be extended with recursive lexical rules as in (Bouma and van Noord, 1994; van Noord and Bouma, 1994) that add adjuncts to a verb's subcategorization set. Long distance scrambling of adjuncts would then be possible in those cases where the embedded verb subcategorizes for the adjunct in its lexical category. I will leave this as a topic of future research.

Not all adjuncts exhibit free word order in Turkish. Certain adjuncts only modify verbs and must be placed immediately next to the verb in a sentence. For example,

(18) a. Ben çikolatayı çok severim.
I chocolate-Acc much like.
"I like chocolate very much."

b. *Çok ben çikolatayı severim.
*Much I chocolate-Acc like.

In Multiset-CCG, this word order restriction is specified in the lexical entry for verbal adjuncts like "çok"; they are assigned a category $S|\{(S|Args)_{lex:+}\}$ that indicates that the argument must be a lexical verb. This argument will not match any derived category because after a grammar rule is applied, each derived category is assigned the feature $[lex:-]$.

Adjectives in Turkish can occur in different word orders within an NP, but they must occur to the left of the noun they modify. In order to capture the head-final nature of Turkish NPs, adjectives in Multiset-CCGs must be assigned a functor category which specifies linear precedence, such as $NP|\{\vec{NP}\}$. There is no definite determiner in Turkish. Bare nouns can act as NPs or can be modified by adjectives. The numeral one, "bir", is used as an indefinite determiner, but since it can occur in any position to the left of the head noun in the NP, it is assigned the same category as adjectives in Multiset-CCG. The following derivations show how the head-final nature of Turkish NPs is handled in Multiset-CCG.

(19) a. sarı kedi
yellow cat
$NP|\{\vec{NP}\}$ NP
——————>
NP

b. *kedi sarı
*cat yellow
NP $NP|\{\vec{NP}\}$
——XXX——

NPs and adjunct clauses often act as islands. Their restrictiveness in long distance scrambling will be discussed in the next section.

3.4 Syntactic Restrictions on Word Order

We have seen that Multiset-CCG is flexible enough to derive sentences with rampantly free word order. However, what is important is that this formalism does not just generate word salad; it can also capture the correct syntactic restrictions on word order in these languages. Although word order in Turkish is quite free, there are some syntactic restrictions. For example, simple NPs must be continuous and head-final in Turkish. We can capture this restriction in Multiset-CCG by restricting the directionality of arguments in the lexical categories and by restricting the composition rules. We will see that Multiset CCG is flexible enough to handle a variety of "free" word order languages with varying degrees of word order freeness. If the composition rules are unrestricted, we can capture languages such as Warlpiri which, unlike Turkish, can have discontinuous NPs. Multiset-CCG can also handle languages more restrictive in word order than Turkish. Strictly verb-final languages, such as Japanese, Korean, and German, can be captured by restricting the directionality of arguments in the lexical categories. Finally, island restrictions on long distance scrambling can be handled by introducing lexical functions that have prioritized multisets of arguments.

In the following sections, I discuss syntactic restrictions on word order in Turkish and briefly discuss how restrictions in other languages could be handled in Multiset-CCG. As we will see, the syntactic restrictions on word order are captured in Multiset-CCG either in the lexical categories (by restricting the directionality of arguments or by allowing two prioritized multisets of arguments in lexical functions) or by restricting the composition rules.

3.4.1 Lack of Case-Marking

In relatively free word order languages, hearers can process sentences with different word order permutations as long as the predicate-argument structures of the sentences can be unambiguously inferred. In Turkish and many other "free" word order languages, case marking provides the essential information for inferring the correct predicate-argument structure of a sentence. Thus, elements with overt case marking generally can scramble freely, even out of embedded clauses. However, arguments of the verb which are not case-marked cannot scramble. For example, Turkish direct objects (NPs and embedded clauses) are normally accusative case marked, but they can quite often occur without any case marking. Object noun phrases which are singular, occurring without a determiner and without case marking are given a nonreferential reading. Although they are not phonologically "incorporated" into the verb (Baker, 1988), a distinct contrast in scrambling behavior of the case-marked and unmarked direct objects is observed.[5] The unmarked

[5](Aissen, 1979; Hankamer, 1979) argue for object-incorporation in the Turkish lexicon, but I am not taking sides on the controversy over whether these unmarked objects are truly incorporated or not.

object is restricted to the immediately preverbal position whereas the case-marked object can occur in any position. Intuitively, we can see that there is no restriction on the word order of case-marked objects because we can infer the predicate-argument structure through the case-markings, however we must rely on word order in order to distinguish an object without case-marking from the subject in sentences like the following.

(20) a. Esra gazete oku-yor.
Esra newspaper read-Prog.
"Esra is reading (some) newspaper(s)."

b. Gazete okuyor Esra.
Newspaper read-Prog Esra.
"She is reading (some) newspaper(s), Esra."

c. ∗Gazete Esra okuyor.
*Newspaper Esra read-Prog.

d. ∗Esra okuyor gazete.
*Esra read-Prog newspaper.

Direct objects occurring with the indefinite article "bir" can also occur without case-marking, and they too are restricted to the immediately preverbal position in most contexts. (Comrie, 1978; Dede, 1986; Enç, 1991) among others associate accusative case-marking in Turkish with the referentiality and specificity of the discourse referent. The accusative case-marking imposes the specific reading on indefinite objects as seen in (21)d, and allows the NP to scramble. The lack of case-marking is associated with a nonspecific reading as seen in (21)a and the inability to scramble as seen in (21)b,c.

(21) a. Fatma ban-a bir kitap ver-di.
Fatma I-Dat one book give-Past.
"Fatma gave me a (nonspecific) book."

b. *Fatma bir kitap bana verdi.
*Fatma one book I-Dat give-Past.

c. ??Fatma bana verdi bir kitap.
??Fatma I-Dat give-Past one book.

d. Fatma bir kitab-ı ban-a verdi.
Fatma one book-Acc I-Dat give-Past.
"Fatma gave me a (specific) book."

However, it is possible to scramble these indefinite NPs without case-marking in contrastive gapping constructions as can be seen in (22)a, unlike the nonreferential bare nouns seen in (22)b.[6] However, in most contexts, objects without case-marking cannot scramble.

(22) a. Bir gömlek san-a, bir gömlek de kardeş-in-e al-dı-m.
One shirt you-Dat, one shirt too sibling-Poss2sg-Dat bought-Past-1Sg.
"(I) bought a shirt for you and another shirt for your sibling."

b. *Gömlek san-a, gömlek de kardeş-in-e al-dı-m.
*shirt you-Dat, shirt too sibling-Poss2sg-Dat bought-Past-1Sg.
"(I) bought some shirt(s) for you and some for your sibling."

In Multiset CCG, verbs which can have a bare noun argument are assigned a function category, e.g. $S|\{Nnom, Ndat\}|\{\overleftarrow{N}\}$ for a ditransitive verb, which has two multisets of arguments that are *prioritized*, i.e. ordered with respect to one another. This category forces the verb to first combine with an unmarked noun on its immediate left before combining with the rest of its arguments in any order. Most transitive verbs will have this extra category since objects without case-marking are very common in Turkish discourse.[7] For example, the following derivation is for (20)b:

(23) Gazete okuyor Esra.
Newspaper reads Esra.
N $S|\{Nnom\}|\{\overleftarrow{N}\}$ $Nnom$
$S|\{Nnom\}$ <
S >

This approach correctly rules out the word orders in (20)c and d, as seen below for sentence c. We do not need an extra verbal category to handle the freedom of word order for indefinites NPs in the special gapping constructions, because these constructions are handled by type-raising the NPs. As will be discussed on page 63, once type-raising is introduced into the grammar, the composition rules must be restricted to disallow the ungrammatical word orders.

(24) Gazete Esra okuyor.
Newspaper Esra reads.
N Nnom S|{Nnom}|{$\overleftarrow{N}$}.
XXX

Word order restrictions with respect to case-marking are also seen in embedded infinitival clauses. Subordinate verbs in Turkish usually occur with a case-marking morpheme that indicates the grammatical function of the whole subordinate clause in the matrix sentence. A small number of verbs subcategorize for embedded clauses which can drop the accusative case-marking as nouns

[6](Erguvanlı, 1984; Knecht, 1986) have pointed out that these indefinite NPs without case-marking cannot be incorporated with the verb, since it is possible to interpose an item between the verb and the object. They claim that bare object nouns are incorporated.

[7]It should be possible to generate this extra lexical category for the appropriate verbs in the lexicon using a lexical reduncy rule.

can. Like the scrambling patterns seen above for object NPs, infinitival clauses with accusative case-marking can occur anywhere in the matrix sentence, example (25), whereas those without the accusative case-marking must remain immediately before the matrix verb, example (26).

(25) a. Ahmet [PRO sinema-ya git-me-yi] çok ist-iyor.
Ahmet [PRO movie-Dat go-Inf-Acc] very want-Prog.
"Ahmet wants to go to the movies very much."

b. [PRO sinema-ya git-me-yi] Ahmet çok ist-iyor.
[PRO movie-Dat go-Inf-Acc] Ahmet very want-Prog.
"To go to the movies, Ahmet wants very much."

c. Ahmet çok ist-iyor [PRO sinema-ya git-me-yi]
Ahmet very want-Prog [PRO movie-Dat go-Inf-Acc]
"Ahmet wants that very much, to go to the movies."

(26) a. Ahmet [PRO sinemaya gitmek] istiyor.
Ahmet [PRO movie-Dat go-Inf] want-Prog.
"Ahmet wants to go to the movies."

b. *[PRO sinemaya gitmek] Ahmet istiyor.

c. *Ahmet istiyor [PRO sinemaya gitmek].

In Multiset-CCG, matrix verbs such as "want" are assigned the categories $S|\{Nn\}|\{\overleftarrow{S_{inf}}\}$ as well as $S|\{Nnom, S_{inf-acc}\}$. The first category captures the word order restrictions that involve infinitive clauses without case-marking, while the second category captures the "free" word order of the infinitive clause with accusative case marking.

3.4.2 Simple and Complex NPs

Simple NPs in Turkish must be continuous in the sentence and head final as seen in (27).

(27) a. [Siyah kedi] gel-di.
[Black cat] come-Past.
"The black cat came in."

b. *Kedi siyah geldi.
*Cat black come-Past.

c. *Siyah geldi kedi.
*Black come-Past cat.

The Multiset-CCG composition rules as presented overgenerate the ungrammatical orders for simple NPs. For example, the ungrammatical sentence below with a discontinuous NP can be generated by using the composition rules. The category $Nx|\{\overrightarrow{Nx}\}$ in the following derivation is

assigned to an adjective which is looking for a noun or noun phrase with any case-marking to its right.

(28)

∗siyah	geldi	kedi.
∗black	came	cat.
$Nx\|\{\vec{N}x\}$	S$\|${Nnom}	Nnom
	$\overline{\qquad\qquad}$<B	
	S$\|\{\vec{\text{N}}\text{nom}\}$	
	$\overline{\qquad\qquad\qquad\qquad}$>	
	S	

The following restriction on the composition rules is needed to prevent verbs and adjectives in Turkish from composing together in the derivation above.

(29) **Backward Composition' ($< B$):**
$Y|Args_Y \quad X|(Args_X \cup \{\overleftarrow{Y}\}) \;\Rightarrow\; X|(Args_X \cup Args_Y)$
(except when X = S and $Y = NP$).

By restricting which categories can take part in the composition rule, we prevent the categories for verbs, $S|\{NP\}$, and for adjectives, $NP|\{\vec{N}P\}$, from combining together before combining with a bare noun. This correctly rules out the first composition in the derivation above and captures the fact that simple NPs must be continuous and head-final in Turkish. However, note that a language such as Warlpiri does allow discontinuous NPs, and Multiset CCG is flexible enough to handle languages that are freer in word order than Turkish, such as Warlpiri, through the use of unrestricted composition rules.

There are also some complex NPs in Turkish that can be discontinuous in a sentence. In the possessive construction in Turkish, the 'possessed' NP contains number and person markings that agree with the 'possessor' NP. Local as well as long distance scrambling is possible for such complex NPs as seen below.

(30) a. Ben [ev-in kapı-sı-nı] boya-dı-m.
I [house-Gen door-Poss3S-Acc] paint-Past-1sg.
"I painted the house's door."

b. Ben [kapısını evin] boyadım.
I [door-Poss3S-Acc house-Gen] paint-Past-1sg.
"I painted its, the house's, door."

c. Evin$_i$ ben [e$_i$ kapısını] boyadım.
House-Gen I door-Poss3S-Acc paint-Past-1sg.
"As for the house, I painted its door."

d. Ben [e_i kapısını] boyadım evin$_i$.
I door-Poss3S-Acc paint-Past-1sg house-Gen.
"I painted its door, the house's."

e. ?Ben evin boyadım kapısını.
I house-Gen paint-Past-1sg door-Poss3S-Acc.

f. ?Kapısını ben evin boyadım.
Door-Poss3S-Acc I house-Gen paint-Past-1sg.

I assign the head (possessed) noun in these constructions a function category that subcategorizes for a genitive cased (possessor) noun in Multiset-CCG because the head noun contains agreement markings similar to verbs in Turkish.

(31)
ev-in	kapı-sı
house-Gen	door-Agr3S
Ngen	Nacc|{Ngen}
	——————<
	Nacc

To allow discontinuous possessive constructions, we must modify the restriction on the composition rule as below.

(32) **Backward Composition" ($< B$):**
$Y|Args_Y \quad X|(Args_X \cup \{\overleftarrow{Y}\}) \;\Rightarrow\; X|(Args_X \cup Args_Y)$
(except when X = S and $Y|Args_Y = NP|\{\overrightarrow{NP}\}$).

Now, possessed NPs can compose with verbs in order to allow long distance scrambling although adjectives in simple NPs cannot. For example, the following discontinuous possessive construction can be derived.[8]

(33)
Ben	kapı-sın-ı	boya-dı-m	ev-in.
I	door-P3-Acc	paint-Pst-1sg	house-Gen.
Nnom	Nacc|{Ngen}	S|{Nnom,Nacc}	Ngen
	——————<B		
	S|{Nnom,Ngen}		
——————————<			
	S|{Ngen}		
——————————>			
	S		

3.4.3 Islands

In some languages, certain clauses act as islands that strictly do not allow extraction for relativization or for long distance scrambling. For example, in German, certain clauses (e.g. finite clauses) can be identified as islands for long distance scrambling. In Turkish, as in many "free"

[8]All the permutations involving the possessive NP can be derived by Multiset-CCG except for (30)f. It is not clear to me whether this word order is grammatical or not, but if it is, type-raising of the genitive-cased noun is necessary to account for this word order, i.e. the type-raised category $Nx|\{Nx|Ngen\}$.

word order languages, island effects are very hard to find.

Extraction for relative clause formation in Turkish has been investigated by (Kornfilt, Kuno, and Sezer, 1980; Sezer, 1986) who show that Turkish does not obey the same island constraints as English in relative clause formation. They present a functional explanation of the data; a relative clause must be a statement about its head noun in order for the extraction to be felicitous in Turkish. (Kuno, 1976; Erteschik-Shir and Lappin, 1979) show that relative clause formation in Japanese and English is also affected by functional constraints concerning thematicity and prominence. I will not discuss islands with respect to relativization in this dissertation, because the constraints for long distance scrambling are usually different than in extraction with respect to relative clause formation.

Sentential subjects in Turkish, and many other SOV languages such as Japanese (Kuno, 1976) and Hungarian (Kiss, 1987), do not show island effects in relativization (Kornfilt, Kuno, and Sezer, 1980; Sezer, 1986) and in long distance scrambling. The following examples show that case-marked elements can be extracted from sentential subjects and long distance scrambled in the matrix sentence. Long distance scrambling in Turkish is even freer than extraction for relative clause formation, since we can freely long distance scramble elements from subject infinitival clauses that do not allow relativization.

(34) a. Bu problemi zor çözmek.
This problem-Acc hard solve-Inf.
"As for this problem, to solve (it) is hard."

b. Çözmek zor bu problemi.
Solve-Inf hard this problem-Acc.
"To solve (it) is hard, this problem."

c. Ali'nin şüpheli [tezini bitireceği].
Ali-Gen doubtful [thesis-3Sg-Acc finish-FutGer-Acc].
"As for Ali, that (he) will finish (his) thesis is doubtful."

d. [Tezini bitireceği] şüpheli Ali'nin.
[Thesis-3Sg-Acc finish-FutGer-Acc] doubtful Ali-Gen.
"That he, Ali, will finish (his) thesis is doubtful."

Moreover, although the strongest island constraint in English is for adjunct clauses, even elements from relative clauses can be extracted for long distance scrambling in Turkish, as seen below.

(35) a. *Ankara'dan*$_i$ sen [e$_i$ dün gel-en] adam-ı tan-ıyor mu-sun?
Ankara-Abl$_i$ you [e$_i$ yest. come-Rel] man-Acc know-Prog Quest-2Sg?
"As for Ankara, do you know the man who came yesterday from there?"

b. Bu kitab-ı ben [e$_i$ yaz-an] kadın-ı tan-ıyor-um.
This book-Acc I [e$_i$ write-Re]l woman-Acc know-Prog-1Sg.
"As for this book, I know the woman who wrote it."

However, long distance scrambling is not completely free. Some adjunct clauses in Turkish do not allow long distance scrambling as seen in (36). My intuitions are that long distance scrambling is not allowed out of adjunct clauses that do not have close semantic links to the matrix clause. Following (Kuno, 1976; Erteschik-Shir and Lappin, 1979), I believe some island phenomena can be explained through functional rather than syntactic means. However, further research is necessary to determine why certain adjunct clauses in Turkish are islands and others are not.

(36) a. Berna [PRO ödev-in-i bit-ir-ince] ban-a yardım ed-ecek.
Berna [PRO hw-3Ps-Acc finish-Aor-Ger] I-dat help do-Fut.
"When (she) finishes (her) homework, Berna is going to help me."

b. *Ödevini bana Berna [bitirince] yardım edecek.
*$Hw\text{-}3P\text{-}Acc_i$ I-dat Berna [finish-ger] help do-3sg.
"As for her homework, Berna is going to help me when she finishes it."

c. *[Berna bitirince] bana yardım edecek *ödevini*.
*[Berna finish-ger] I-dat help do *hw-3Ps-Acc*.
*"When she finishes it, Berna is going to help me, her homework."

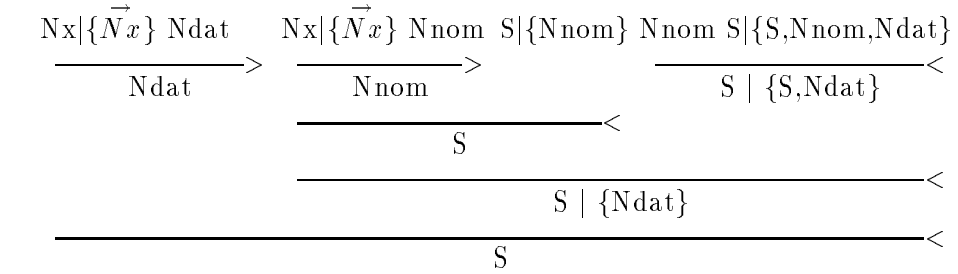
I account for clauses exhibiting island behaviour in Multiset-CCG within the lexicon. I assign the head of the island clause a category with two prioritized multisets of arguments such as $S|\{S\}|\{Nnom, Nacc\}$. This function makes certain that the head combines with all of its NP arguments before combining with the matrix clause, S. As demonstrated in (37) below, long distance scrambling out of such an adjunct clause is thus prohibited.

(37) *[Berna bitirince] bana yardım edecek *ödevini*.
*Berna finish-ger I-dat help do hw-3Ps-Acc
$Nnom$ $S|\{S\}|\{Nnom, Nacc\}$ ———S——— $Nacc$
———————————< ————XXX————
$S|S|\{Nacc\}$
————————XXX————————

In contrast, heads of adjunct clauses which are not islands are assigned categories such as $S|\{S, Nnom, Nacc\}$. Since this category can combine with the matrix verb even before it has combined with all of its arguments, it allows long distance scrambling of its arguments. The example below is from (Erguvanlı, 1984), p.27.

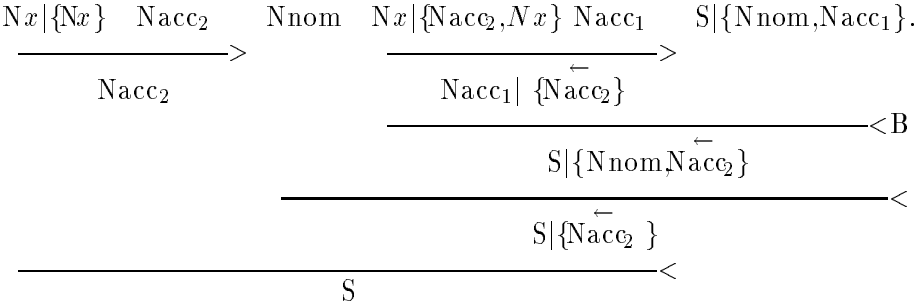
(38) a. *Bu şehr-e$_i$* iki ay ol-du [biz e$_i$ gel-eli].
This city-Dat two month be-Past [we come-Ger].
"To this city, it's been two months [since we came]."

b. Bu şehr-e iki ay ol-du biz gel-eli.
This city-Dat two month be-Past we come-Ger.



To handle relative clauses that allow extraction for long distance scrambling, the head of the relative clause is assigned the category $Nx|\{Arguments, \vec{N}x\}$ instead of the island preserving category $Nx|\{\vec{N}x\}|\{Arguments\}$ with two prioritized multisets of arguments.

(39) Bu kitab-ı ben yaz-an adam-ı tan-ıyor-um.
This book-Acc I write-Rel man-Acc know-Prog-1Sg.



Although it is harder to extract an element from an adjunct clause, this is not true of all adjunct clauses. Thus, we cannot state that every adjunct clause is an island. Instead, I argue that island characteristics should be captured in individual lexical categories for Turkish. This lexical control is very advantageous for capturing the island behaviour in Turkish. However, further research is needed to determine what types of adjunct clauses exhibit island behaviour in order to specify the appropriate categories in the lexicon.

Some languages with relatively free word order are more restricted in long distance scrambling than Turkish. For example in German, long distance scrambling is not allowed out of finite (tensed) clauses (40)a, but is allowed out of infinitival clauses (40)b. The following examples are from (Rambow, 1994a).

(40) a. *Peter hat den Kühlschrank versprochen, daß er reparieren wird.
*Peter has the refrigerator promised, that he repair will.
"Peter has promised that he will repair the refrigerator."

b. ...den Kühlschrank$_i$ niemand [t$_i$ zu reparieren] versprochen hat.
...the refrigerator$_i$ noone [t$_i$ to repair] promised has.
"...that no-one has promised to repair the refrigerator."

This behaviour can be captured by restricting the composition rules in Multiset-CCG.[9] The restriction would state that tensed verbs cannot combine with other verbs using the composition rules, because there rules are what allow long distance scrambling in Multiset-CCG. They would only be allowed to combine via the application rules, which do not merge the argument sets of the two verbs.

3.4.4 The Immediately Preverbal Position

Another syntactic restriction on long distance scrambling in Turkish is that elements from embedded clauses cannot be placed in the immediately preverbal position of other clauses. Since the immediately preverbal position is associated with the focus, this position in each clause must be occupied by a constituent which is an integral part of the event described by that clause's verb. Any constituent that is a part of the matrix sentence's predicate-argument structure can scramble freely into this position (41)a. However, elements of embedded clauses cannot scramble to the position immediately before the matrix verb (e.g. (41)b), even though they can occur between the subordinate verb and the matrix verb when they are locally scrambled within their own clause as in (41)c. Crucially, "this book" in (41)c does not receive stress or a high pitch accent; this is how we know that it is a part of the embedded clause and not in the immediately-preverbal focus position with respect to the matrix verb.

(41) a. Ahmet [ben-im kitab-ı oku-duğ-um-u] Fatma'ya söyle-di.
Ahmet [I-Gen book-Acc read-Ger-1Sg-Acc] Fatma-Dat say-Past.
"Ahmet told FATMA that I read the book."

b. *Ahmet [benim e_i okuduğumu] Fatma'ya *kitabı*$_i$ söyledi.
*Ahmet [I-Gen e_i read-Past-Ger-1Sg-Acc] Fatma-Dat *book-Acc*$_i$ say-Past.
*"It was the BOOK that Ahmet told Fatma that I read."

c. Ayşe [benim okuduğumu bu kitabı], bil-iyor.
Ayşe [I-Gen read-Past-Ger-1Sg-Acc this book-Acc] know-Pres.
"Ayşe knows that I read it, this book."

In theories involving syntactic movement and traces, the sentence in b can be ruled ungrammatical because the moved element cannot c-command its trace.[10] Similarly in Multiset-CCG, this sentence cannot be derived because the NP "book" cannot combine with the constituents on either side.

[9]Further research is needed to capture the German V2 restriction in CCGs. (Hepple, 1990) captures V2 phenomena in a Lambek Calculus by separating the specification of the order of the complements in a clause from the position of the head in the clause.

[10]Such an approach may have problems in handling Hungarian, since Hungarian allows elements from the embedded clause to scramble into the immediately preverbal position in the matrix clause.

(42) Ahmet [I-Gen read-Ger-1Sg-Acc] Fatma-Dat *book-Acc* say-Past.
Nn Ng $S_{ger}|\{Ng,Na\}$ Nd Na $S|\{Nn,Nd,S_{ger}\}$
$\frac{\qquad\qquad\qquad}{S_{ger}|\{Na\}}<$ $\frac{\qquad\qquad\qquad}{XXX}$

However, if type-raising of NPs is added to Multiset-CCGs, then the grammar will overgenerate this sentence. I will discuss how the grammar rules can be restricted to disallow this in the following sections, page 63.

3.4.5 Ambiguity in Long Distance Scrambling

There is a great potential for ambiguity in long distance scrambling in recovering the appropriate predicate-argument structures of each clause. This potential for ambiguity may be the reason that long distance scrambling is much less common than local scrambling in natural discourse. For instance, scrambling a case-marked NP out of an embedded clause is generally blocked if there is an NP with the same case-marking already in the matrix clause (e.g. (43)c). This makes sense because when the arguments are not uniquely case-marked, it is hard to determine the appropriate predicate-argument structure of each clause. For example, a uniquely case-marked NP can be long distance scrambled out of its clause in (43)b, but an NP cannot be long distance scrambled into a clause that has another NP with the same nominative-case (43)d. Speakers prefer a reading of these ambiguous sentences where each NP is interpreted as the argument of the closest center-embedded verb, i.e. in the canonical word order.

(43) a. Fatma [Ali ev-e git-ti] san-dı.
Fatma [Ali house-Dat go-Past] think-Past.
"Fatma thought Ali went home."

b. *Eve*$_i$ Fatma [Ali e$_i$ gitti] sandı.
House-Dat$_i$ Fatma [Ali e$_i$ go-Past] think-Past.
"To the house, Fatma thought that Ali went there."

c. Ali Fatma eve gitti sandı.
Ali Fatma house-Dat go-Past think-Past.
"Ali thought that Fatma went home."
*"As for Ali, Fatma thought that he went home."

d. $*$*Ali*$_i$ Fatma [e$_i$ eve gitti] sandı.
Ali$_i$ Fatma [e$_i$ house-Dat go-Past] think-Past.

At first glance, this seems to be a syntactic restriction on long distance scrambling; however, we can find some exceptions, for instance, if the two NPs that have the same case marking are far enough apart as in (44)a. In sentences with a direct complement clause whose subject has been raised to be an object in the matrix clause (44)b and c, the case of the raised object does not

seem to interfere with long distance scrambling.

(44) a. *Esra'ya*$_i$, Ahmet [ben-im e$_i$ yardım et-tiğ-im-i], Fatma'ya söyle-di.
Esra-Dat$_i$ Ahmet [I-Gen e$_i$ help do-Ger-1sg-Acc] Fatma-Dat say-Past.
"As for Esra, Ahmet told Fatma that I helped her."

b. *Bu kitab-ı*$_i$ Ahmet ben-i [e$_i$ oku-du] san-ıyor.
This book-Acc$_i$ Ahmet I-Acc [e$_i$ read-Past] think-Prog.
"As for this book, Ahmet thinks of me that I read it."

c. Ben-i Ahmet [e$_i$ oku-du] san-ıyor *bu kitab-ı*$_i$.
I-Acc Ahmet [e$_i$ read-Past] think-Prog *this book-Acc*$_i$.
"Ahmet thinks of me that I read it, this book."

These exceptions support the hypothesis that the restriction on unique case in long distance scrambling is a processing limitation rather than a syntactic one. The intuition is that we have difficulty processing these sentences with two NPs with the same case marking because we cannot easily disambiguate the predicate-argument structures of each clause and figure out which NP belongs to which verb. Thus, I argue that the competence grammar should allow long distance scrambling even in situations where the NPs cannot be distinguished by case-marking.

Multiset-CCG treats sentences where more than one argument has the same case-marking as ambiguous, unless there are other features that distinguish the two arguments. For example, the string below can be interpreted as a sentence with long distance scrambling by using the composition rules, even though Turkish speakers cannot process this reading.

Ali$_i$	Fatma	[t$_i$ ev-e	git-ti]	san-ıyor.
Ali$_i$	Fatma	[t$_i$ house-Dat	go-Past]	think-Prog.
Nnom:Ali	Nnom:Fatma	Ndat:home	S:go(x,y)|{Nnom:x,Ndat:y}	S:think(z,p)|{Nnom:z, S:p}

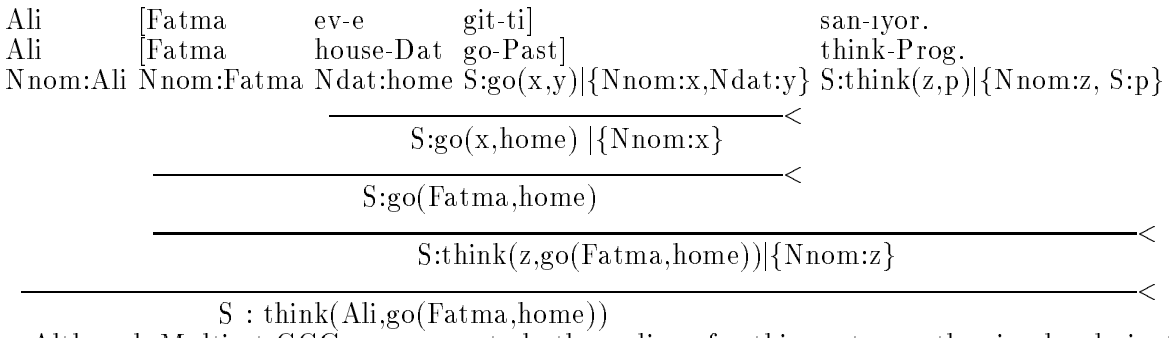
S:go(x,home) |{Nnom:x} <

S:think(z,go(x,home)) |{Nnom:z, Nnom:x} <B

S:think(Fatma,go(x,home)) |{Nnom:x} <

S : think(Fatma,go(Ali,home)) <

However, there is a much simpler derivation possible for the same string that uses only application rules, not the more powerful composition rules. This derivation gives us the preferred reading, where there is no long distance scrambling.



Although Multiset-CCG can generate both readings for this sentence, the simpler derivation that does not use the more costly composition rules would be preferred by a system that has processing constraints. The composition rules allow the multiset of arguments to grow during the derivation and thus are more costly. We can model why speakers of Turkish strongly prefer the second reading over the first for these ambiguous sentences by adding processing constraints to the Multiset-CCG formalism. However, further research is necessary to develop an incremental parsing strategy for Multiset-CCG that captures all of the performance constraints in long distance scrambling.

3.5 Type-Raising and Coordination

Type-raising has been used by (Steedman, 1985; Steedman, 1989) to handle coordination and gapping constructions in English and Dutch within CCGs. Similarly, we can add type-raised categories to Multiset-CCG to capture the coordination of NP sequences in gapping constructions in Turkish. For example, the gapping constructions 'SO and SOV' in (45)a and 'SOV and SO' in (45)b are possible in Turkish. In addition, gapping of more than one verb is possible as in English. In (45)c, the NPs in the coordinated sequences are not from the same clause.

(45) a. Ayşe kitabı, Fatma da gazeteyi okuyor. (SO and SOV)
Ayşe book-Acc, Fatma too newspaper-Acc reads.
"Ayşe is reading the book and Fatma the newspaper."

b. Ayşe kitabı okuyor, Fatma da gazeteyi. (SOV and SO)
Ayşe book-Acc reads, Fatma too newspaper-Acc.
"Ayşe is reading the book and Fatma the newspaper."

c. Ali Fatma'nın, Ahmet de Ayşe'nin gittiğini görmüş.
Ali Fatma-Gen, Ahmet too Ayşe-Gen go-Nom-Acc saw.(NP_1 NP_2 and NP_1 NP_2 V_2 V_1)
"Ali saw that Fatma went away, and Ahmet Ayşe."

In most "free" word order languages, the NPs in the gapping construction do not have to be in the canonical word order. The NP sequences can be scrambled in Turkish as seen in (46)a for a single clause, and in (46)b in a complex sentence. However, the coordination of differently scrambled NP sequences is not acceptable in Turkish, as seen in (46)c, although these are marginally acceptable in other "free" word order languages such as German (Rambow, personal communication).

(46) a. Kitabı Ayşe, gazeteyi de Fatma okuyor. (OS and OSV)
Book-Acc Ayşe, newspaper-Acc too Fatma read-Prog.
"As for the book, Ayşe is reading it, and the newspaper, Fatma."

b. Fatma'nın Ali, Ayşe'nin de Ahmet gittiğini görmüş.
Fatma-Gen Ali, Ayşe-Gen too Ahmet go-Nom-Acc saw.(NP_2 NP_1 and NP_2 NP_1 V_2 V_1)
"As for Fatma, Ali saw that she went away, and as for Ayşe, Ahmet did."

c. *Ayşe kitabı, gazeteyi de Fatma okuyor. (SO and OSV)
*Ayşe book-acc, newspaper-acc too Fatma reads.
"*Ayşe is reading the book and the newspaper Fatma."

Type-raising in CCGs converts NPs into functions over verbal categories. Then, these type-raised NPs can combine together to form constituents that can be coordinated. In Multiset CCG, we can adapt type-raised categories for functions that have a multiset of arguments, as seen

in (47). Case-marked nouns in Turkish are assigned the following order-preserving type-raised categories, as well as the basic NP category in the lexicon. The simpler basic category is always tried first during a derivation, as a strategy suggested in (Partee and Rooth, 1983) to decrease the processing load. If the parse is unsuccessful, all the NPs in the sentence are assigned the following type-raised categories and the parsing process is restarted.[11]

(47) a. $(S|Args)|\{(S|(Args \cup \{\overleftarrow{\text{NP}}\}))^{\rightarrow}\}$
b. $(S|Args)|\{(S|(Args \cup \{\overrightarrow{\text{NP}}\}))^{\leftarrow}\}$

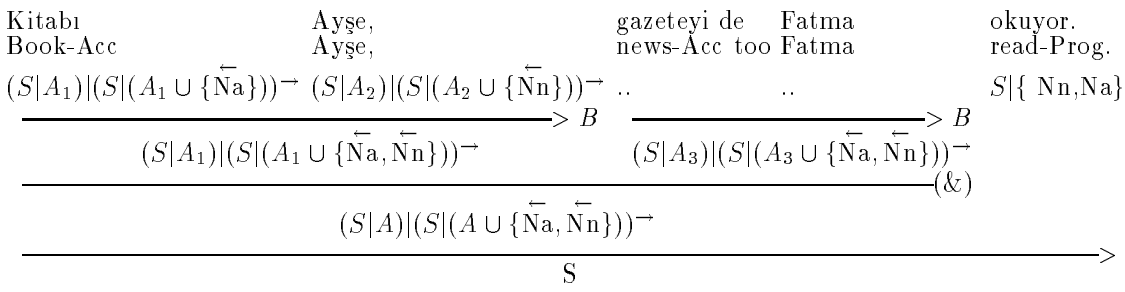
The first category is a function that is looking for a verb on its right; this verb is also a function looking for the original NP (with the appropriate case) on its left as well as any number of other arguments. Once this type-raised NP has combined with a verbal category, the result is a function which is looking for the remaining arguments of the verb. The second type-raised category in (47)b is for NPs that are placed in post-verbal positions; it is a function that is looking for a verb on its left which is looking for the NP on its right.

Multiset-CCGs can model a strictly verb-final language like Korean by only assigning the first type-raised category to the noun phrases of that language. Since most case-marked nouns in Turkish can occur behind the verb, both type-raised categories are necessary. Some NPs, for example question words, in Turkish can only occur in preverbal positions, as seen in (48). Thus, theey assigned only the rightward looking type-raised category, $(S : quest(X)|Args)|\{(S|(Args \cup \{\overleftarrow{\text{NP:X}}\}))^{\rightarrow}\}$, which changes a declarative sentence into a interrogative.

(48) a. Fatma kim-i ara-dı?
Fatma who-Acc seek-Past?
"Who did Fatma call?"
b. *Fatma ara-dı kim-i?
*Fatma seek-Past who-Acc?

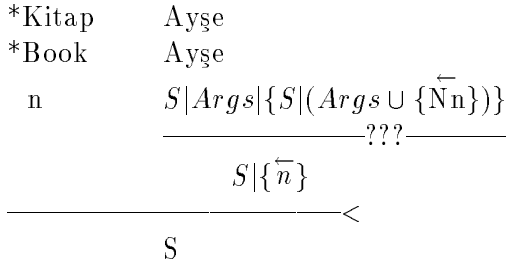
Since type-raised categories are functions, two type-raised noun phrases can combine together using the forward and backward composition rules and a coordination rule (&). For example, a sentence with the word order 'OS and OSV' can be incrementally derived as below. The composition rule binds $S|A_2$ to $(S|(A_1 \cup \{\overleftarrow{\text{Na}}\}))^{\rightarrow}\}$. Note that the grammar can capture the scrambled word order without resorting to non-order-preserving type-raising.

[11] This may cause problems for an incremental parsing strategy. Further research is necessary to determine how type-raised categories should be used by the grammar.



However, this approach also allows the coordination of strings such as 'S O and O S V' where the coordinated NP strings do not have the same word order. These strings are not grammatical in Turkish, but they have been reported to be grammatical in other "free" word order languages, e.g. German. I argue that these coordinations are ungrammatical in Turkish because the pragmatic information structure of the two coordinated strings are not the same. The coordination rule (&) must be restricted to only apply if the information structure as well as the syntactic/semantic category of each constituent is the same. I am leaving the question of how to do this for future research. Further research is also needed to determine whether the use of these type-raised categories that contain variables, e.g. $Args$, changes the weak generative capacity of Multiset-CCG.

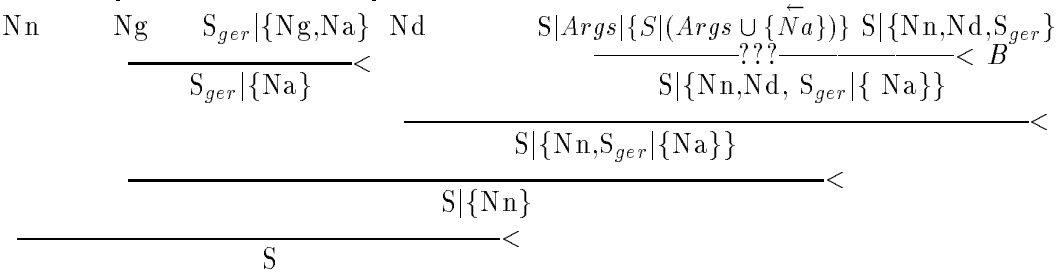
One problem with using type-raised categories for NPs is that they may cause the grammar to overgenerate, for instance, the ungrammatical word orders with incorporated objects. As mentioned before, object nouns or clauses without case marking are forced to remain in the immediately pre-verbal position. A matrix verb can have a curried category such as $S|\{Nn\}|\{\overleftarrow{n}\}$ to allow it to combine with a noun without case-marking (n) to its immediate left. However, to restrict a type-raised Nn from interposing in between the matrix verb and the subordinate clause, we must restrict type-raised noun phrases and verbs from composing together.



The exact same problem is seen with overgenerating the ungrammatical long distance scrambling into the immediately preverbal focus position.

(50) a. *Ahmet [benim e_i okuduğumu] Fatma'ya *kitabı*$_i$ söyledi.
*Ahmet [I-Gen e_i read-Past-Ger-1Sg-Acc] Fatma-Dat *book-Acc*$_i$ say-Past.
*"Ahmet told Fatma that I read the BOOK."

b. *Ahmet [I-Gen read-Ger-Acc] Fatma-Dat *book-Acc* say-Past.



We need to restrict the composition rules in order to disallow the composition between type-raised NPs and verbs. Type-raised NPs and verbal categories are allowed to combine using the application rules but not the composition rules. In fact, the same restriction is placed on the Dutch grammar by (Steedman, 1985). The restriction below on the forward composition rule will disallow example (49), and the restrictions on the backward composition rule will disallow example (50) as well as preventing verbs and adjectives from combining together to disallow discontinuous NPs as determined on page 54.

(51) a. **Forward Composition' ($> B$):**

$X|(Args_X \cup \{\overrightarrow{Y}\})$ $Y|Args_Y$ $\Rightarrow$ $X|(Args_X \cup Args_Y)$

except when X = (S|{...}) and Y = S.

b. **Backward Composition"' ($< B$):**

$Y|Args_Y$ $X|(Args_X \cup \{\overleftarrow{Y}\})$ $\Rightarrow$ $X|(Args_X \cup Args_Y)$

except when X = S and Y = NP or (S|{...}).

3.6 Summary

In this chapter, I have presented a competence grammar which captures the syntax of "free" word order in Turkish. Multiset-CCG is a categorial grammar in which verbs are represented as functions which subcategorize for a multiset of arguments without specifying their relative ordering. The directionality of the arguments can be specified in the lexical function. A set of application and composition rules are defined to combine these lexical functions and their arguments. Multiset CCG is able to derive the predicate-argument structure of sentences regardless of the word order. The grammar uniformly handles free word order among arguments and adjuncts in a clause, as well as permutation of elements from more than one clause, i.e. long distance scrambling, by the use of the composition rules. As we will see in Chapter 6, the flexibility of the Multiset-CCG surface structure is crucial for integrating the information structure of Turkish "free" word order with Multiset-CCG.

Multiset-CCG can also capture syntactic restrictions on word order. The composition rules, which allow long distance scrambling, can be restricted to disallow the composition of certain categories. In addition, many syntactic restrictions can be lexicalized. As discussed in this chapter, the following syntactic restrictions can be captured in Multiset-CCG:

- **Head final clauses:** by restricting the directionality of subcategorized arguments in the lexical category for the head.
- **Adjunct Islands, Incorporated Objects:** by allowing two prioritized multisets of arguments in the lexical category of the head. (Having more than one multiset of arguments in lexical functions allows the location of some arguments to be frozen with respect to the other arguments.)
- **Continuous NPs:** by restricting the composition rules so that they cannot compose adjectives with verbs.

Multiset CCG is flexible enough to handle a variety of "free" word order languages with varying degrees of word order freeness. If the composition rules are unrestricted, we can capture languages such as Warlpiri which, unlike Turkish, can have discontinuous NPs. Multiset-CCG can also handle languages more restrictive in word order than Turkish. Strictly verb-final languages, such as Japanese, Korean, and German, can be captured by restricting the directionality of arguments in the lexical categories. Finally, island restrictions on long distance scrambling can be handled by introducing lexical functions that have prioritized multisets of arguments. In addition, the composition rules can be restricted in different ways to limit the long distance scrambling to certain constituents in each language.

Chapter 4

A Formal Analysis of Multiset-CCG

We have seen in Chapter 2 that CCGs are not expressive enough to handle all word order variations in Turkish. In chapter 3, I presented a formalism, Multiset-CCG, which can capture the characteristics of "free" word order languages. In this chapter, I present a formal analysis of Multiset-CCGs in Section 1. I first review the weak generative capacity of CCGs. Then, I provide formal descriptions of three different versions of Multiset-CCGs. I discuss their weak generative capacity and prove that Multiset CCGs are within the class of context-sensitive grammars. I present a polynomial time parsing algorithm for Multiset CCGs on page 91. In section 2, I compare the Multiset-CCG formalism with other computational approaches to the syntax of "free" word order languages.

4.1 The Formal Properties of Multiset-CCGs

4.1.1 The Weak Generative Capacity of CCGs

CCGs are an extension of Categorial Grammars (Ajdukiewicz, 1935; Bar-Hillel, 1953). Pure categorial grammars only contain application rules that combine functions with their arguments. (Hillel, Gaifman, and Shamir, 1960) prove that Pure CGs are weakly equivalent to context-free grammars. However, (Shieber, 1985b) has shown that context-free grammars are not adequate to handle natural languages, specifically cross-serial dependencies in Swiss German or Dutch. CCGs extend CGs by adding composition and type-raising rules that are able to handle cross-serial dependencies. These rules increase the weak generative capacity of CCGs, beyond context-free grammars.

CCGs are *mildly context-sensitive grammars*. As defined by (Joshi, 1985), these are context-sensitive grammars that have the following characteristics:

- Ability to produce limited cross-serial dependencies
- Constant Growth Property (related to the semilinearity property)
- Polynomial Parsability

A language L has the constant growth property if for all $w \in L$ where $|w| > c_0$, a constant, there is a $w' \in L$ s.t. $|w| = |w'| + c$ for some $c \in C$, a finite set of constants. The linguistic intuition behind this property is that sentences in a natural language are built from a set of clauses with bounded structure using linear operations. (Weir, 1988) believes that the slightly stronger property of semilinearity is closer to this intuition. A language has the semilinearity property if the number of occurrences of each symbol in any string in the language is a linear combination of the occurrences of the symbols in a fixed finite set of strings. As shown by (Parikh, 1966), context-free languages are known to be semilinear.

In the formal definition of CCGs in (Weir, 1988), a CCG, G, is denoted by (V_T, V_N, S, f, R), where

- V_T is a finite set of terminals (lexical items),
- V_N is a finite set of nonterminals (atomic categories),
- S is a distinguished member of V_N,
- f is a function that maps elements of $V_T \cup \{\epsilon\}$ to finite subsets of $C(V_N)$, the set of categories, where,
  - $V_N \subseteq C(V_N)$ and
  - if c_1 and $c_2 \in C(V_N)$, then $(c_1 \backslash c_2)$ and $(c_1 / c_2) \in C(V_N)$.
- R is a finite set of combinatory rules where $X, Y, Z_1, \ldots, Z_n$ are variables over the set of categories $C(V_N)$. Certain restrictions may be placed on the possible instantiations of the variables in the rules. The slash variable $|_i$ in the composition rules can bind to $\backslash$ or $/$.
  - **Forward Application** (>):
    $X/Y \;\; Y \;\; \rightarrow X$
  - **Backward Application** (<):
    $Y \;\; X\backslash Y \;\; \rightarrow X$
  - **Generalized Forward Composition** (>B(n) or >Bx(n)): For some $n \geq 1$,
    $X/Y \quad Y|_1 Z_1|_2 \ldots|_n Z_n \;\; \rightarrow X|_1 Z_1|_2 \ldots|_n Z_n$
  - **Generalized Backward Composition** (<B(n) or <Bx(n)): For some $n \geq 1$,
    $Y|_1 Z_1|_2 \ldots|_n Z_n \quad X\backslash Y \;\; \rightarrow X|_1 Z_1|_2 \ldots|_n Z_n$

The derives relation in a CCG is defined as $\alpha c\beta \Rightarrow \alpha c_1 c_2 \beta$ if R contains the rule $c_1 c_2 \rightarrow c$. The language generated by this grammar is defined as

$$L(G) = \{a_1, \ldots, a_n \mid S \stackrel{*}{\Rightarrow} c_1, \ldots, c_n, c_i \in f(a_i), a_i \in V_T \cup \{\epsilon\}, 1 \leq i \geq n\}$$

(Weir and Joshi, 1988; Weir, 1988; Vijay-Shanker and Weir, 1990) have proven that:

Theorem 4.1 *CCGs are weakly equivalent to the following mildly context-sensitive formalisms: Tree-Adjoining Grammars (TAGs), Head Grammars (HGs), and Linear Indexed Grammars (LIGs).*

It is informative to compare CCGs to the stack-based LIG formalism, first considered by (Gazdar, 1988). An Indexed Grammar is an extension of a context-free grammar where each nonterminal is associated with a stack of unbounded size; Indexed Grammars are context-sensitive grammars (Aho, 1968). A Linear Indexed Grammar

CCGs are an extension of Categorial Grammars (Ajdukiewicz, 1935; Bar-Hillel, 1953). Pure categorial grammars only contain application rules that combine functions with their arguments. (Hillel, Gaifman, and Shamir, 1960) prove that Pure CGs are weakly equivalent to context-free grammars. However, (Shieber, 1985b) has shown that context-free grammars are not adequate to handle natural languages, specifically cross-serial dependencies in Swiss German or Dutch. CCGs extend CGs by adding composition and type-raising rules that are able to handle cross-serial dependencies. These rules increase the weak generative capacity of CCGs, beyond context-free grammars.

CCGs are *mildly context-sensitive grammars*. As defined by (Joshi, 1985), these are context-sensitive grammars that have the following characteristics:

- Ability to produce limited cross-serial dependencies
- Constant Growth Property (related to the semilinearity property)
- Polynomial Parsability

A language L has the constant growth property if for all $w \in L$ where $|w| > c_0$, a constant, there is a $w' \in L$ s.t. $|w| = |w'| + c$ for some $c \in C$, a finite set of constants. The linguistic intuition behind this property is that sentences in a natural language are built from a set of clauses with bounded structure using linear operations. (Weir, 1988) believes that the slightly stronger property of semilinearity is closer to this intuition. A language has the semilinearity property if the number of occurrences of each symbol in any string in the language is a linear combination of the occurrences of the symbols in a fixed finite set of strings. As shown by (Parikh, 1966), context-free languages are known to be semilinear.

In the formal definition of CCGs in (Weir, 1988), a CCG, G, is denoted by (V_T, V_N, S, f, R), where

- V_T is a finite set of terminals (lexical items),
- V_N is a finite set of nonterminals (atomic categories),
- S is a distinguished member of V_N,
- f is a function that maps elements of $V_T \cup \{\epsilon\}$ to finite subsets of $C(V_N)$, the set of categories, where,
  - $V_N \subseteq C(V_N)$ and
  - if c_1 and $c_2 \in C(V_N)$, then $(c_1 \backslash c_2)$ and $(c_1 / c_2) \in C(V_N)$.
- R is a finite set of combinatory rules where $X, Y, Z_1, \ldots, Z_n$ are variables over the set of categories $C(V_N)$. Certain restrictions may be placed on the possible instantiations of the variables in the rules. The slash variable $|_i$ in the composition rules can bind to $\backslash$ or $/$.
  - **Forward Application** (>):
    $X/Y \;\; Y \;\; \rightarrow X$
  - **Backward Application** (<):
    $Y \;\; X\backslash Y \;\; \rightarrow X$
  - **Generalized Forward Composition** (>B(n) or >Bx(n)): For some $n \geq 1$,
    $X/Y \quad Y|_1 Z_1|_2 \ldots|_n Z_n \;\; \rightarrow X|_1 Z_1|_2 \ldots|_n Z_n$
  - **Generalized Backward Composition** (<B(n) or <Bx(n)): For some $n \geq 1$,
    $Y|_1 Z_1|_2 \ldots|_n Z_n \quad X\backslash Y \;\; \rightarrow X|_1 Z_1|_2 \ldots|_n Z_n$

The derives relation in a CCG is defined as $\alpha c \beta \Rightarrow \alpha c_1 c_2 \beta$ if R contains the rule $c_1 c_2 \rightarrow c$. The language generated by this grammar is defined as

$$L(G) = \{a_1, \ldots, a_n \mid S \stackrel{*}{\Rightarrow} c_1, \ldots, c_n, c_i \in f(a_i), a_i \in V_T \cup \{\epsilon\}, 1 \leq i \geq n\}$$

(Weir and Joshi, 1988; Weir, 1988; Vijay-Shanker and Weir, 1990) have proven that:

Theorem 4.2 *CCGs are weakly equivalent to the following mildly context-sensitive formalisms: Tree-Adjoining Grammars (TAGs), Head Grammars (HGs), and Linear Indexed Grammars (LIGs).*

It is informative to compare CCGs to the stack-based LIG formalism, first considered by (Gazdar, 1988). An Indexed Grammar is an extension of a context-free grammar where each

nonterminal is associated with a stack of unbounded size; Indexed Grammars are context-sensitive grammars (Aho, 1968). A Linear Indexed Grammar

CCGs are an extension of Categorial Grammars (Ajdukiewicz, 1935; Bar-Hillel, 1953). Pure categorial grammars only contain application rules that combine functions with their arguments. (Hillel, Gaifman, and Shamir, 1960) prove that Pure CGs are weakly equivalent to context-free grammars. However, (Shieber, 1985b) has shown that context-free grammars are not adequate to handle natural languages, specifically cross-serial dependencies in Swiss German or Dutch. CCGs extend CGs by adding composition and type-raising rules that are able to handle cross-serial dependencies. These rules increase the weak generative capacity of CCGs, beyond context-free grammars.

CCGs are *mildly context-sensitive grammars*. As defined by (Joshi, 1985), these are context-sensitive grammars that have the following characteristics:

- Ability to produce limited cross-serial dependencies
- Constant Growth Property (related to the semilinearity property)
- Polynomial Parsability

A language L has the constant growth property if for all $w \in L$ where $|w| > c_0$, a constant, there is a $w' \in L$ s.t. $|w| = |w'| + c$ for some $c \in C$, a finite set of constants. The linguistic intuition behind this property is that sentences in a natural language are built from a set of clauses with bounded structure using linear operations. (Weir, 1988) believes that the slightly stronger property of semilinearity is closer to this intuition. A language has the semilinearity property if the number of occurrences of each symbol in any string in the language is a linear combination of the occurrences of the symbols in a fixed finite set of strings. As shown by (Parikh, 1966), context-free languages are known to be semilinear.

In the formal definition of CCGs in (Weir, 1988), a CCG, G, is denoted by (V_T, V_N, S, f, R), where

- V_T is a finite set of terminals (lexical items),
- V_N is a finite set of nonterminals (atomic categories),
- S is a distinguished member of V_N,
- f is a function that maps elements of $V_T \cup \{\epsilon\}$ to finite subsets of $C(V_N)$, the set of categories, where,
  - $V_N \subseteq C(V_N)$ and
  - if c_1 and $c_2 \in C(V_N)$, then $(c_1 \backslash c_2)$ and $(c_1 / c_2) \in C(V_N)$.

- R is a finite set of combinatory rules where $X, Y, Z_1, \ldots, Z_n$ are variables over the set of categories $C(V_N)$. Certain restrictions may be placed on the possible instantiations of the variables in the rules. The slash variable $|_i$ in the composition rules can bind to \ or /.

  - **Forward Application** (>):
    $X/Y \;\; Y \;\; \rightarrow X$
  - **Backward Application** (<):
    $Y \;\; X\backslash Y \;\; \rightarrow X$
  - **Generalized Forward Composition** (>B(n) or >Bx(n)): For some $n \geq 1$,
    $X/Y \quad Y|_1 Z_1|_2 \ldots|_n Z_n \;\; \rightarrow X|_1 Z_1|_2 \ldots|_n Z_n$
  - **Generalized Backward Composition** (<B(n) or <Bx(n)): For some $n \geq 1$,
    $Y|_1 Z_1|_2 \ldots|_n Z_n \quad X\backslash Y \;\; \rightarrow X|_1 Z_1|_2 \ldots|_n Z_n$

The derives relation in a CCG is defined as $\alpha c \beta \Rightarrow \alpha c_1 c_2 \beta$ if R contains the rule $c_1 c_2 \rightarrow c$. The language generated by this grammar is defined as

$$L(G) = \{a_1, \ldots, a_n \mid S \stackrel{*}{\Rightarrow} c_1, \ldots, c_n, c_i \in f(a_i), a_i \in V_T \cup \{\epsilon\}, 1 \leq i \geq n\}$$

(Weir and Joshi, 1988; Weir, 1988; Vijay-Shanker and Weir, 1990) have proven that:

Theorem 4.3 *CCGs are weakly equivalent to the following mildly context-sensitive formalisms: Tree-Adjoining Grammars (TAGs), Head Grammars (HGs), and Linear Indexed Grammars (LIGs).*

It is informative to compare CCGs to the stack-based LIG formalism, first considered by (Gazdar, 1988). An Indexed Grammar is an extension of a context-free grammar where each nonterminal is associated with a stack of unbounded size; Indexed Grammars are context-sensitive grammars (Aho, 1968). A Linear Indexed Grammar

CCGs are an extension of Categorial Grammars (Ajdukiewicz, 1935; Bar-Hillel, 1953). Pure categorial grammars only contain application rules that combine functions with their arguments. (Hillel, Gaifman, and Shamir, 1960) prove that Pure CGs are weakly equivalent to context-free grammars. However, (Shieber, 1985b) has shown that context-free grammars are not adequate to handle natural languages, specifically cross-serial dependencies in Swiss German or Dutch. CCGs extend CGs by adding composition and type-raising rules that are able to handle cross-serial dependencies. These rules increase the weak generative capacity of CCGs, beyond context-free grammars.

CCGs are *mildly context-sensitive grammars*. As defined by (Joshi, 1985), these are context-sensitive grammars that have the following characteristics:

- Ability to produce limited cross-serial dependencies
- Constant Growth Property (related to the semilinearity property)
- Polynomial Parsability

A language L has the constant growth property if for all $w \in L$ where $|w| > c_0$, a constant, there is a $w' \in L$ s.t. $|w| = |w'| + c$ for some $c \in C$, a finite set of constants. The linguistic intuition behind this property is that sentences in a natural language are built from a set of clauses with bounded structure using linear operations. (Weir, 1988) believes that the slightly stronger property of semilinearity is closer to this intuition. A language has the semilinearity property if the number of occurrences of each symbol in any string in the language is a linear combination of the occurrences of the symbols in a fixed finite set of strings. As shown by (Parikh, 1966), context-free languages are known to be semilinear.

In the formal definition of CCGs in (Weir, 1988), a CCG, G, is denoted by (V_T, V_N, S, f, R), where

- V_T is a finite set of terminals (lexical items),
- V_N is a finite set of nonterminals (atomic categories),
- S is a distinguished member of V_N,
- f is a function that maps elements of $V_T \cup \{\epsilon\}$ to finite subsets of $C(V_N)$, the set of categories, where,
  - $V_N \subseteq C(V_N)$ and
  - if c_1 and $c_2 \in C(V_N)$, then $(c_1 \backslash c_2)$ and $(c_1 / c_2) \in C(V_N)$.
- R is a finite set of combinatory rules where $X, Y, Z_1, \ldots, Z_n$ are variables over the set of categories $C(V_N)$. Certain restrictions may be placed on the possible instantiations of the variables in the rules. The slash variable $|_i$ in the composition rules can bind to $\backslash$ or $/$.
  - **Forward Application** (>):
    $X/Y \;\; Y \;\; \rightarrow X$
  - **Backward Application** (<):
    $Y \;\; X\backslash Y \;\; \rightarrow X$
  - **Generalized Forward Composition** (>B(n) or >Bx(n)): For some $n \geq 1$,
    $X/Y \quad Y|_1 Z_1|_2 \ldots|_n Z_n \;\; \rightarrow X|_1 Z_1|_2 \ldots|_n Z_n$
  - **Generalized Backward Composition** (<B(n) or <Bx(n)): For some $n \geq 1$,
    $Y|_1 Z_1|_2 \ldots|_n Z_n \quad X\backslash Y \;\; \rightarrow X|_1 Z_1|_2 \ldots|_n Z_n$

The derives relation in a CCG is defined as $\alpha c \beta \Rightarrow \alpha c_1 c_2 \beta$ if R contains the rule $c_1 c_2 \rightarrow c$. The language generated by this grammar is defined as

$$L(G) = \{a_1, \ldots, a_n \mid S \stackrel{*}{\Rightarrow} c_1, \ldots, c_n, c_i \in f(a_i), a_i \in V_T \cup \{\epsilon\}, 1 \leq i \geq n\}$$

(Weir and Joshi, 1988; Weir, 1988; Vijay-Shanker and Weir, 1990) have proven that:

Theorem 4.4 *CCGs are weakly equivalent to the following mildly context-sensitive formalisms: Tree-Adjoining Grammars (TAGs), Head Grammars (HGs), and Linear Indexed Grammars (LIGs).*

It is informative to compare CCGs to the stack-based LIG formalism, first considered by (Gazdar, 1988). An Indexed Grammar is an extension of a context-free grammar where each nonterminal is associated with a stack of unbounded size which can be copied from parent to daughters by the rules; Indexed Grammars are context-sensitive grammars (Aho, 1968). In a Linear Indexed Grammar only one of the nonterminal daughters on the right hand side of a rule can inherit the unbounded stack from the parent, the left-hand side nonterminal. LIGs are defined as $G = (V_T, V_N, V_S, S, P)$, where

- V_T is a finite set of terminals,
- V_N is a finite set of nonterminals,
- V_S is a finite set of stack symbols,
- S is a distinguished member of V_N,
- P is a finite set of productions having the form
  (i) $A\,[] \rightarrow a$
  (ii) $A\,[..l] \rightarrow A_1\,[] \ldots A_i[..] \ldots A_n\,[]$ (pop)
  (iii) $A\,[..] \rightarrow A_1\,[] \ldots A_i[..l] \ldots A_n\,[]$ (push)
  where $A_1, \ldots, A_n \in V_N, l \in V_S$, and $a \in V_T \cup \{\epsilon\}$

(Weir and Joshi, 1988) prove that having an unbounded number of composition rules (for all n, allowing an arbitrarily large category $Y|_1 Z_1 \ldots|_n Z_n$) increases the weak generative capacity of CCGs beyond mildly context-sensitive Tree-Adjoining languages. If we compare such a grammar to LIGs, we see that an arbitrary number of Z arguments would mean having a rule which splits the stack associated with the nonterminal on the left hand side into two unbounded parts for each of the nonterminal daughters on the right hand side (i.e. $A\,[..X..Z] \rightarrow A_1\,[..X, Y] \quad Y\,[..Z]$). Since it is not possible to split an unbounded stack in a LIG, the number of Z arguments in the

composition rules of a CCG must be restricted to some particular n in order to be equivalent to a LIG.

The conversion of a CCG into a LIG relies on the fact that the combinatory rules in the CCG are linear. To enforce linearity, only the category X in the combinatory rules can be unbounded in size; the variables Y and Z must be bounded in their possible instantiations. In other words, only a finite number of categories can fill the secondary constituent of each combinatory rule. The secondary constituent is the second of the pair in the forward rules or the first of the pair in the backward rules (i.e. Y in the application rules and $Y|Z_1...|Z_n$ in the composition rules).

Weir and Joshi point out that we can substitute all the useful instances of the secondary constituent and expand the combinatory rules to a larger but finite set. In the expanded set of combinatory rules in CCGs, only one variable, X, can match a category of unbounded size. Similarly, the rules in a LIG involve a single unbounded stack. Thus, any of the expanded combinatory rules can be converted to a LIG production. For example, the LIG productions corresponding to an instance of the forward application and forward composition rules in CCG are as following; the nonterminals are indicated by capital letters and the terminals with small letters, and we assume that the stack symbols include the set of terminals and nonterminals.

(1) a. For all possible terminals y_i and some $n \geq 0$,
$A[..] \rightarrow A[..(Y|_1 y_1 ...|_n y_n)] \quad Y[|_1 y_1 ...|_n y_n]$

b. For all possible terminals y_i and z_i, and some $n, m \geq 0$,
$A[..|_1 z_1 ...|_n z_n] \rightarrow A[..(Y|_1 y_1 ...|_n y_n)] \quad Y[|_1 y_1 ...|_n y_n|_1 z_1 ...|_m z_m]$

It is not neccessary to restrict the size of the secondary constituents matching Y and $Y|Z_1|...Z_n$ in the formal definition of the CCG rules, because the following lemma holds of the grammar, as proven by (Weir and Joshi, 1988).

Lemma 4.1 *There is a bound (determined by the grammar G) on the number of useful categories that can match the secondary constituent of a rule.*

The set of derivable categories in CCGs is infinite in size, however Weir and Joshi show that the set of the components of all derivable categories is bounded in size. The components of a category $c = (c_0|_1 c_1|_2 ...|_n c_n)$ are its immediate components $c_0, ..., c_n$ and the components of these immediate components. A finite set $D_C(G)$ can be defined that contains all derivable components of every *useful* category. A category c is *useful* if $c \stackrel{*}{\Rightarrow} w$ for some w in ${V_T}^*$:

$$c \in D_C(G) \text{ if } c \text{ is a component of } c' \text{where } c' \in f(a) \text{ for some } a \in V_T \cup \{\epsilon\}.$$

Given that every useful category matching the secondary constituents Y and $Y|Z_1|...Z_n$ in the combinatory rules has components which are in $D_C(G)$, the lemma given above holds.

(Vijay-Shanker and Weir, 1993) give a polynomial time $O(n^6)$ parsing algorithm for CCGs based on the LIG-equivalency proof above.

4.1.2 The Weak Generative Capacity of Multiset-CCGs

A Multiset-CCG, G, is defined as (V_N, V_T, S, f, R), where

- V_N is a finite set of nonterminals.
  These are basic categories that can be components of lexical categories; they include atomic categories such as S and NP and basic functional categories such as $S\backslash NP$ that are nonterminals in many other grammar, e.g. V or VP. There are also nonterminals that are marked with the direction feature such as $\vec{NP}$ or $\overleftarrow{NP}$.
- V_T is a finite set of terminals (lexical items),
- S is a distinguished member of V_N,
- f is a lexical assignment function that maps elements of V_T to finite subsets of $C(V_N)$, the set of categories possible in G.
- and R is a finite set of combinatory rules.

There are different ways in which the set of categories, $C(V_N)$, and the combinatory rules R can be defined. I make a distinction between three versions of Multiset-CCG: Pure Multiset-CCG, Prioritized Multiset-CCG, and Curried Multiset CCG. In Pure Multiset-CCG, the result and the arguments of a lexical function category are basic nonterminal categories. In the last chapter, we saw that it is useful to extend Multiset-CCG with functions that specify two multisets of arguments prioritized in a particular linear order in order to handle island restrictions in long distance scrambling. I will call this extension Prioritized Multiset-CCG; this is the version of the grammar that I am using to capture Turkish. However, we can define an even more general version of Prioritized Multiset-CCG such that functions are allowed an unbounded number of prioritized multisets, in fact a stack of multisets. I call this formalism Curried Multiset-CCG; it can be used to simulate a CCG for English or a Prioritized Multiset-CCG for Turkish. Examples of function categories in each type of Multiset-CCG are shown below:

- **Pure Multiset-CCG:** S|{NP,NP}
- **Prioritized Multiset-CCG:** S|{NP,NP}|{NP}
- **Curried Multiset-CCG:** S|{NP,NP}|{NP}|{NP,NP}...

In the next sections, I will discuss in detail how Pure, Prioritized, and Curried Multiset CCGs each define the set of categories and rules used in the grammar in a slightly different way and have increasing generative capacities.

4.1.2.1 Pure Multiset-CCGs

A Pure Multiset Combinatory Categorial Grammar, Multiset-CCG, is denoted by $G = (V_N, V_T, S, f, R)$ as defined above. The set of categories, $C(V_N)$, possible in G is defined as follows:

- $V_N \subseteq C(V_N)$
- if $A_0, A_1, ..., A_n \in V_N$, then $A_0|\{A_1, ..., A_n\} \in C(V_N)$.[1]

R in G is a finite set of combinatory rules defined as follows:

- $A|(U \cup \{\vec{B}\}) \quad B \rightarrow A|U.$
- $B \quad A|(U \cup \{\overleftarrow{B}\}) \rightarrow A|U.$
- $A|(U \cup \{\vec{B}\}) \quad B|V \rightarrow A|(U \cup V).$
- $B|V \quad A|(U \cup \{\overleftarrow{B}\}) \rightarrow A|(U \cup V).$

$A, B \in V_N$, and U and V are multisets of categories in V_N. General restrictions may be placed on the possible instantiations of the variables in the rules, e.g. the direction feature indicated by $\rightarrow$ or $\leftarrow$ above the categories.
The derives relation in a Multiset-CCG is defined as

- If R contains the rule $c_1 c_2 \rightarrow c$, then $\alpha c \beta \Rightarrow \alpha c_1 c_2 \beta$.
- If $c \in f(a)$ for some $a \in V_T$, then $\alpha c \beta \Rightarrow \alpha a \beta$.

The language generated by this grammar is $L(G) = \{w \mid S \stackrel{*}{\Rightarrow} w, w \in V_T^*\}$.
We can show that Pure Multiset-CCG can generate languages that CCGs cannot.

Theorem 4.5 *Pure Multiset-CCLs $\not\subseteq$ CCLs.*

The Bach language or MIX-5 = { w | w is a string of an equal number of a's, b's, c's, d's, and e's but mixed in any order} intersected with the regular language $a^*b^*c^*d^*e^*$ generates the language $\{a^n b^n c^n d^n e^n | n \geq 0\}$ which is known not to be a Combinatory Categorial Language (CCL) (Weir and Joshi, 1988). Since CCLs are closed under intersection with regular languages, MIX-5 cannot be a CCL. However, Multiset-CCG below generates MIX-5 using the rules of forward composition and application:[2]

$f(\epsilon) = \{S|\{B, C, D, E, S\}, S\}$

[1] $A_1, ...A_n$ may be nonterminals that are marked by a direction feature such as $\vec{A_i}$.

[2] Including ϵ in V_T, following (Weir, 1988), simplifies the grammar for these formal languages. However, the empty string is not a part of the grammar for natural languages. It should be possible to replace the use of ϵ with some terminal marker in these grammars for formal languages as well.

$f(a) = A$, $f(b) = B$, $f(c) = C$, $f(d) = D$, $f(e) = E$.

Thus, Pure Multiset-CCGs can generate languages that CCGs cannot.

However, Pure Multiset-CCGs are not strictly of greater generative capacity than CCGs, because CCGs can also generate languages that Pure Multiset-CCGs cannot.

Theorem 4.6 *CCLs* $\not\subseteq$ *Pure Multiset-CCLs.*

For example, COUNT-4 = $\{a^n b^n c^n d^n | n \geq 0\}$ can be generated by CCGs but not by Pure Multiset-CCGs. This is because Pure Multiset-CCG nonterminals are only associated with multisets and not stacks; there is no way to ensure the linear order that all the a's occur before the b's and that the b's occur before the c's. The direction arrows can only distinguish the order of two categories, not three or more. Thus, Pure Multiset-CCGs and CCGs can each generate languages that the other cannot.

4.1.2.2 Prioritized Multiset-CCGs

A Prioritized Multiset-CCG is denoted by $G = (V_N, V_T, S, f, R)$ where $C(V_N)$, the set of categories, is defined as:

- $V_N \subseteq C(V_N)$
- for $\exists k1, k2$, if $A_0, A_1, ...A_{k1}, ...A_{k2} \in V_N$,
  then $A_0|\{A_1, ...A_{k1}\}|\{A_{k1+1}, ...A_{k2}\} \in C(V_N)$.

Function categories in Prioritized Multiset-CCGs can have up to two argument multisets. These multisets are prioritized such that all the arguments in one of the multisets must be found before the arguments in the other multiset.

R in G is a finite set of combinatory rules, where $Y \in V_N$ and X is either a nonterminal or of the form $A|W$ where $A \in V_N$ and W is a multiset of nonterminals. U, V_1, and V_2 are multisets of categories in V_N.

- $X|(U \cup \{\vec{Y}\}) \quad Y \rightarrow X|U$
- $Y \quad X|(U \cup \{\overleftarrow{Y}\}) \rightarrow A|U$
- $X|(U \cup \{\vec{Y}\}) \quad Y|V_1|V_2 \rightarrow X|(U \cup V_1)|V_2$.
- $Y|V_1|V_k \quad X|(U \cup \{\overleftarrow{Y}\}) \rightarrow X|(U \cup V_1)|V_2$.

We assume that $A|U_1|U_2$ where $U_2 = \emptyset$ reduces to $A|U_1$. General restrictions may be placed on the possible instantiations of the variables in the rules, e.g. the direction feature indicated by $\rightarrow$ or $\leftarrow$ above the categories.

The derives relation in Prioritized Multiset-CCGs is defined in the same way as Pure Multiset-CCGs.

We can show that:

Theorem 4.7 *Pure Multiset CCLs* $\subset$ *Prioritized Multiset CCLs.*

Prioritized Multiset CCG can simulate Pure Multiset CCG by just restricting all categorial functions to one multiset of arguments. In addition, Prioritized Multiset-CCG has a greater generative capacity than Pure Multiset-CCGs, because it can derive the COUNT-3 language. The grammar below generates (COUNT-3)* using the rules of forward composition and application:

$f(a) = A, f(b) = S|\{\overleftarrow{A}, \overrightarrow{C}\}|\{S\}, f(c) = C, f(\epsilon) = S.$

However, Prioritized Multiset-CCG still cannot derive every language that CCGs can derive.

Theorem 4.8 *CCLs* $\not\subseteq$ *Prioritized Multiset CCLs.*

The categories in Prioritized Multiset-CCG have a bounded number of prioritized multisets; thus they cannot simulate a stack of arguments. I conjecture that Prioritized Multiset-CCG cannot derive the languages $\{ww|a, b, c \in w\}$ or COUNT-4.

4.1.2.3 Curried Multiset-CCGs

Curried Multiset-CCG is the generalized form of Prioritized Multiset-CCG. A Curried Multiset-CCG is denoted by $G = (V_N, V_T, S, f, R)$, where $C(V_N)$, the set of categories, is defined as:

- $V_N \subseteq C(V_N)$
- if $c_0 \in C(V_N)$ and $c_1, \ldots, c_n \in V_N$, then $(c_0|\{c_1, \ldots, c_n\}) \in C(V_N)$.

Thus, function categories in Curried Multiset-CCGs can have a unbounded stack of multisets. The set of rules R is defined such that all the arguments in the multiset at the top of the stack must be found before the arguments in the other multisets. R is a finite set of combinatory rules, where $X \in C(V_N)$, $Y \in V_N$, and U and $V_1, ...V_k$ are multisets of categories in V_N which may be empty sets. We assume that $A|U_1|...U_{k-1}|U_k$ where $U_k = \emptyset$ reduces to $A|U_1...|U_{k-1}$.

- $X|(U \cup \{\overrightarrow{Y}\}) \quad Y \rightarrow X|U.$
- $Y \quad X|(U \cup \{\overleftarrow{Y}\}) \rightarrow X|U.$
- $X|(U \cup \{\overrightarrow{Y}\}) \quad Y|V_1|...|V_k \rightarrow X|(U \cup V_1)|V_2|...V_k.$
- $Y|V_1|...|V_k \quad X|(U \cup \{\overleftarrow{Y}\}) \rightarrow X|(U \cup V_1)|V_2|...V_k.$

Theorem 4.9 Prioritized Multiset-CCLs $\subset$ Curried Multiset-CCLs

Curried Multiset-CCG can simulate a Prioritized Multiset-CCG by restricting all categorial functions to two multisets of arguments in the lexicon and rules. In addition, Curried Multiset-CCG can generate languages such as WW that Prioritized Multiset-CCG cannot. This is because categories in Curried Multiset-CCG are associated with a stack of multisets and can simulate CCGs.

Unlike Pure and Prioritized Multiset-CCGs, we can show that:

Theorem 4.10 *CCLs $\subset$ Curried Multiset-CCLs.*

Each CCG function containing a stack of arguments such as $X\backslash Y/Z/W$ can be simulated by the Curried Multiset-CCG function with single element sets such as $X|\{\overleftarrow{Y}\}|\{\overleftarrow{Z}\}|\{\overleftarrow{W}\}$.[3] Thus, Curried Multiset-CCG can generate languages that CCGs can, e.g. WW and COUNT-4. Curried Multiset-CCG can also generate all of the languages that Pure and Prioritized Multiset-CCG can, such as MIX-5, that CCGs cannot generate.

In the next section I will show that all three versions of Multiset-CCG are within context-sensitive grammars, however they do not have the full power of context-sensitive grammars.

Conjecture 4.1 *Context-Sensitive Languages $\not\subseteq$ Curried Multiset-CCLs.*

Indexed Grammars are known to be context-sensitive (Aho, 1968) and can generate languages such as $\{www|w \in \Sigma^*\}$ which Multiset-CCGs cannot. IGs can generate such languages by copying a stack of indices to more than one daughter. Although Curried Multiset-CCG also has a stack associated with its categories, it can only pass this stack to one of the daughters in the combinatory rules as in CCGs (Weir, 1988). It can only generate languages that either CCGs can or Prioritized Multiset-CCG can.

It is also possible to extend Multiset-CCGs with type-raising. In the last chapter, page 63, I introduced a polymorphic type-raising category for Multiset-CCGs to handle coordination of NPs, and I pointed out that the addition of type-raising allows the grammar to generate Turkish sentences that pure Multiset-CCGs cannot. Further research is necessary to determine whether type-raising with polymorphic categories increases the power of CCGs. Regardless of the power of type-raising with variables, (Weir, 1988) has shown that a general coordination rule does increase the power of CCGs to the weak generative capacity of Indexed Grammars. I assume that the Turkish grammar will only use the potential extra power of the type-raised categories and the general coordination rule as a last resort.

[3] A type-raised category such as $S/(S\backslash NP)$ in CCGs corresponds to $S|\{S|\{\overrightarrow{\overleftarrow{NP}}\}\}$ in Curried Multiset-CCG, where lexical components such as $S|\{\overrightarrow{\overleftarrow{NP}}\}$ are defined as members of V_N in Curried Multiset CCG.

4.1.2.4 Summary of the Weak Generative Capacity

In summary, the set of categories, $C(V_N)$, is defined in the following ways for each version of Multiset-CCG.

- Pure Multiset-CCG: $V_N \subseteq C(V_N)$ and
  if $A_0, A_1, ..., A_n \in V_N$, then $A_0|\{A_1, ..., A_n\} \in C(V_N)$.
- Prioritized Multiset-CCG: $V_N \subseteq C(V_N)$ and for $\exists k1, k2$,
  if $A, A_1, ...A_{k1}...A_{k2} \in V_N$, then $A|\{A_1, ...A_{k1}\}|\{A_{k1+1}, ...A_{k2}\} \in C(V_N)$.
- Curried Multiset-CCG: $V_N \subseteq C(V_N)$ and
  if $c_0 \in C(V_N)$ and $c_1, ...c_n \in V_N$, then $c_0|\{c_1, ..., c_n\} \in C(V_N)$.

Given these definitions, it is clear that:

Theorem 4.11 Pure Multiset-CCLs $\subset$ Prioritized Multiset-CCLs $\subset$ Curried Multiset-CCLs

We have also seen that *CCLs* $\subseteq$ *Curried Multiset-CCLs*. In addition, I conjecture that Curried Multiset-CCG cannot generate some context-sensitive languages that IGs can. Thus, *ILs* $\not\subseteq$ *Curried Multiset-CCLs*.

Table 4.1 describes the formal languages that each formalism can generate.

	CFG	CCG	Pure M-CCG	Prior. M-CCG	Curried M-CCG	IG
COUNT-2	Yes	Yes	Yes	Yes	Yes	Yes
COUNT-3	No	Yes	No	Yes	Yes	Yes
COUNT-4	No	Yes	No	No(?)	Yes	Yes
COUNT-5	No	No	No	No	No	Yes
COUNT-K	No	No	No	No	No	Yes
MIX-3	No	No(?)	Yes	Yes	Yes	No(?)
MIX-5	No	No	Yes	Yes	Yes	No(?)
MIX-K	No	No	Yes	Yes	Yes	No(?)
WW	No	Yes	No	No	Yes	Yes
WWW	No	No	No	No	No	Yes

Table 4.1: Formal Languages

In the next sections, we will see that the Multiset-CCG formalisms are within the class of context-sensitive grammars and that they are polynomially parsable. They do not have the full power of context-sensitive grammars because they cannot generate the WWW or COUNT-K languages.

4.1.3 Formal Equivalence to {}-LIGs

Multiset-CCGs is very similar to {}-LIG, (Rambow, 1994a), an LIG based formalism which has a multiset instead of a stack associated with each nonterminal. In fact, in this section I will show that:

Theorem 4.12 {}*-LIG is weakly equivalent to a restricted version of Multiset-CCG.*

As defined by (Rambow, 1994a; Rambow, 1994b), a *multiset-valued Linear Index Grammar* ({}-LIG) is a 5-tuple (V_N, V_T, V_I, P, S), where V_N and V_T are disjoint sets of terminals and nonterminals, respectively, V_I is the set of indices, S is the start symbol and P is a set of productions of the following form:

$As \longrightarrow v_0 B_1 s_1 v_1 \ldots B_n s_n v_n$
where $A, B_1, ..., B_n \in V_N$, $v_0, ..., v_n \in V_T^*$,
and $s, s_1, ..., s_n$ are multisets of members in V_I.

Each nonterminal in a {}-LIG is associated with a multiset. The multiset s associated with the lefthand side nonterminal A in the rule above consists of a finite number of indices that will be removed from A's complete multiset. The multiset s_i associated with each nonterminal B_i consists of a finite number of indices that will be added to the multisets associated with each B_i. The operations involving the addition and removal of indices is specified in the derivation rule below. In addition, the unbounded multiset associated with the nonterminal on the lefthand side can be distributed in any way among the nonterminal daughters on the righthand side during the derivation.

The *derivation relation* $\Longrightarrow$ for a {}-LIG is defined as follows. Let $\beta, \gamma \in (V_N V_I^* \cup V_T)^*$, and $t, t_1, ...t_n$ are multisets of members of V_I:

$\beta A t \gamma \Longrightarrow \beta v_0 B_1 t_1 v_1 \ldots B_n t_n v_n \gamma$ where $t = \cup_{i=1}^{n} (t_i \backslash s_i) \cup s$.

A *linearly restricted* derivation in a {}-LIG is a derivation $S \stackrel{*}{\Rightarrow} w$ with $w \in V_T^*$ such that:

- The number of index symbols added during the derviation is linearly bounded by $|w|$.
- The number of ϵ-productions used during the derivation is linearly bounded by $|w|$.

(Rambow, 1994a) proves that the linearly restricted {}-LILs,
where L_R({}-LIG) = { w | there is a linearly restricted derivation $S \stackrel{*}{\Rightarrow} w$},
are context-sensitive and polynomially parsable. In fact, polynomially restricted {}-LIGs are also polynomially parsable, in time $O(n^2 + q(n)^{|V_I|})$.

We can show that

Lemma 4.2 *Pure Multiset-CCLs* $\subseteq$ *{}-LILs.*

Let $G = (V_N, V_T, S, f, R)$ be a Pure Multiset-CCG; we can construct an equivalent {}-LIG, $G' = (V_N, V_T, V_I', P', S)$, where $V_I' = \{A', \overrightarrow{A'}, \overleftarrow{A'} \mid A \in V_N\}$, and P' is defined as below.
For each lexical item $a \in V_T \cup \{\epsilon\}$, the following rule is added to P':

- Basic elements:
  If $f(a) = A \in V_N$ then $(A \rightarrow a) \in P'$.
- Functors:
  If $f(a) = A_0|\{A_1, \ldots, A_k\}$ where $A_0, A_1, ..., A_k \in V_N$ and may have a direction feature associated with them, then $(A_0\{A_1', \ldots, A_k'\} \rightarrow a) \in P'$.

P' also includes rules corresponding to the combinatory rules in G.
For all $A, B \in V_N$:

- $(A \rightarrow A\{\overrightarrow{B'}\} \quad B) \in P'$.
- $(A \rightarrow B \quad A\{\overleftarrow{B'}\}) \in P'$.

The derivation relation in {}-LIGs can simulate function composition with these productions by freely distributing index symbols associated with the nonterminal on the left-hand side among the daughters.

For example, the following Pure Multiset-CCG generates the MIX (or Bach) language using the forward application and composition rules:

$f(\epsilon) = \{S|\{A, B, C, S\}\ ,\ S\ \}$,
$f(a) = A$, $f(b) = B$, $f(c) = C$.
We can construct an equivalent {}-LIG by using the methods outlined above:

1. $S\{A', B', C', S'\} \rightarrow \epsilon$, $S \rightarrow \epsilon$,
2. $A \rightarrow a$, $B \rightarrow b$, $C \rightarrow c$, $D \rightarrow d$, $E \rightarrow e$,
3. $S \rightarrow S\{A'\}\ \ A$,
4. $S \rightarrow S\{B'\}\ \ B$,
5. $S \rightarrow S\{C'\}\ \ C$,
6. $S \rightarrow S\{S'\}\ \ S$

This {}-LIG can generate the string "baacbc" in MIX:

$S \stackrel{5}{\Rightarrow} S\{C'\}\ \ C \stackrel{4}{\Rightarrow} S\{C', B'\}\ \ B\ \ C$
$\stackrel{*}{\Rightarrow} S\{C', B', C', A', A', B'\}\ \ B\ \ A\ \ A\ \ C\ \ B\ \ C$
$\stackrel{6}{\Rightarrow} S\{A', B', C', S'\}\ \ S\{A', B', C'\}\ \ B\ \ A\ \ A\ \ C\ \ B\ \ C$
$\stackrel{1}{\Rightarrow} \epsilon\ \ S\{A', B', C'\}\ \ B\ \ A\ \ A\ \ C\ \ B\ \ C$
$\stackrel{6}{\Rightarrow} \epsilon\ \ S\{A', B', C', S'\}\ \ S\ \ B\ \ A\ \ A\ \ C\ \ B\ \ C$
$\stackrel{*}{\Rightarrow} \epsilon\ \epsilon\ \epsilon\ baacbc.$

Note that when production 6 is used, the indices are freely distributed to the daughters; this mimics the composition of such daughter constituents in Multiset-CCGs.

We can prove that $L(G) = L(G')$, specifically $C \stackrel{*}{\Rightarrow}_G w$ for $A \in C(V_N)$ iff $C' \stackrel{*}{\Rightarrow}_{G'} w$ for $C' \in V_N V_I^*$ and $w \in V_T$, by induction on the length of the derivations. Thus, Pure Multiset-CCLs $\subseteq$ {}-LILs. However, {}-LIGs can generate the COUNT-k languages whereas Pure Multiset-CCG cannot generate COUNT-3. Thus,

Lemma 4.3 *{}-LILs $\not\subseteq$ Pure Multiset-CCLs.*

Rambow conjectures that {}-LIGs cannot generate the language $\{ww | a, b, c \in w\}$ and thus LILs are not contained in {}-LILs. Since Curried Multiset-CCGs can simulate CCGs, we know that they cannot be equivalent to {}-LIGs either. Thus,

Conjecture 4.2 *{}-LILs $\not\subseteq$ Curried Multiset-CCLs.*

There is a version of Multiset-CCGs, which I will call Restricted {}-CCG, that are weakly equivalent to {}-LIGs.

Theorem 4.13 *Restricted {}-CCLs = {}-LILs.*

A restricted {}-CCG is defined like a Pure Multiset-CCG except that the combinatory rules are slightly different. Composition is restricted so that crossing (nonharmonic) composition is not allowed, and a new rule called "partial" application is allowed for complex arguments. These rules allow a type of composition where we can match a part of a function that includes the result as well as some of the arguments in the multiset. Composition in Pure Multiset-CCG only allows a match on the result part of the secondary function, but the partial application shown below matches the result as well as one of the arguments in the multiset.

$$\frac{S|\{(S|\{\overleftarrow{NP}\})\} \quad S|\{\overleftarrow{NP},\ \overrightarrow{NP}\}}{S|\{\overleftarrow{NP}\}} > \text{(partial)}$$

The combinatory rules for Restricted {}-CCG are defined in an abbreviated format as follows, where $A, B \in V_N$, and U, V, and W are multisets of categories in V_N or the empty set; V is further restricted such that all arguments in V have a direction feature to the right or the left as indicated by the subscripts r and l.

- $A|(U \cup \{B|\overrightarrow{W}\}) \quad B|(W \cup V_r) \rightarrow A|(U \cup V)$.
- $B|(W \cup V_l) \quad A|(U \cup \{B|\overleftarrow{W}\}) \rightarrow A|(U \cup V)$.

We can prove that:

Lemma 4.4 *Restricted* {}*-CCLs* $\subseteq$ {}*-LILs.*

The proof is the same as the proof that shows Pure Multiset-CCCLs $\subseteq$ {}-LILs except that the {}-CCG rules are replaced with the following {}-LIG productions: For all $A, B, and (B|\{B_1 ... B_k\}) \in V_N$:

- $(A \rightarrow A\{(B|\{B_1, \overrightarrow{..., B_k\})'\} \quad B\{B'_1 .. B'_k\}) \in P'$.
- $(A \rightarrow B\{B'_1 ... B'_k\} \quad A\{(B|\{B_1, \overleftarrow{..., B_k\})'\}) \in P'$.

In addition,

Lemma 4.5 {}*-LILs* $\subseteq$ *Restricted* {}*-CCLs.*

For every {}-LIG G in extended two (normal) form, an equivalent Restricted {}-CCG G' can be constructed. Let $G = (V_N, V_T, V_I, P, S)$, where the productions, P, are of the following types where $A, B, B_1, B_2 \in V_N, a \in V_T, i \in V_I$:

1. $As \rightarrow a$
2. $As \rightarrow Bt$
3. $As \rightarrow B_1 u \;\; B_2 v$

An equivalent Restricted {}-CCG, $G' = (V'_N, V_T, S, f, R)$, where $V'_N = V_N \cup V_I$. The function f and the rules R are defined as follows.

1. For each production of type 1 in P, f in G' is defined such that $A|s \in f(a)$.
2. For each production of type 2 in P,
   $A|(s \cup \{B|t\}) \in f(\epsilon)$.
3. For each production of type 3 in P,
   $(A|(s \cup \{(\overleftarrow{B_1}|u), (\overrightarrow{B_2}|v)\})) \in f(\epsilon)$.

The rules in Restricted {}-CCG have already been defined above.
For example, the following {}-LIG generates the language COUNT-3:

(2) a. $S \rightarrow S\{a, b, c\}$
    b. $S \rightarrow AT$

c. $T \rightarrow BC$

d. $A\{a\} \rightarrow A' \quad A$

e. $B\{b\} \rightarrow B' \quad B$

f. $C\{c\} \rightarrow C' \quad C$

g. $A'\{\} \rightarrow a$,

h. $B'\{\} \rightarrow b$,

i. $C\{\} \rightarrow c$

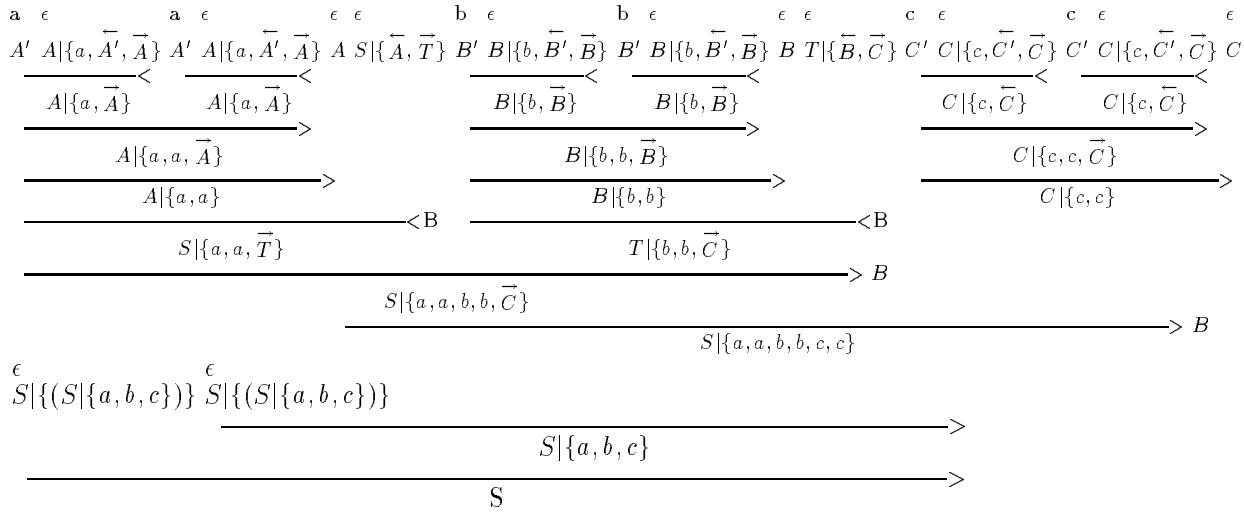
j. $A \rightarrow \epsilon$, $B \rightarrow \epsilon$, $C \rightarrow \epsilon$,

The equivalent Restricted {}-CCG, G', is as follows.

(3) a. $f(\epsilon) = \{S|\{(S|\{a,b,c\})\}, S|\{\overleftarrow{A}, \overrightarrow{T}\}, T|\{\overleftarrow{B}, \overrightarrow{C}\}, A|\{a, \overleftarrow{A'}, \overrightarrow{A}\}, B|\{b, \overleftarrow{B'}, \overrightarrow{B}\}, C|\{c, \overleftarrow{C'}, \overrightarrow{C}\}, A, B, C\}$

b. $f(a) = A', f(b) = B', f(c) = C'$

A sample derivation in G' follows:



Thus, Restricted {}-CCGs are weakly equivalent to {}-LIGs.

4.1.4 Context Sensitivity of Multiset CCG

A grammar is *context-sensitive* if and only if it can be accepted by some nondeterministic but linearly bounded Turing machine (Lewis and Papdimitriou, 1981) such that if $(s, \#w\underline{\#}) \models_M (q, u\underline{a}v)$ then $|uav| \leq |\#w\#|$.

We can show that:

Theorem 4.14 *A Curried Multiset-CCG G such that $G = (V_N, V_T, f, R)$ where $\epsilon \notin V_T$, is context-sensitive.*

Since the Curried Multiset-CCG above is a lexicalized grammar that does not include the empty string, we know that all bottom-up derivations in the grammar are linearly bounded in length with respect to the length of the input string, (Greibach, 1965). Thus, we can construct a nondeterministic Turing machine M which accepts $w \in L(G)$.

Let the tape symbols be $V_N \cup V_T \cup \{\{, \}, |, @\}$. For an input string $a_1...a_n$, M first writes the lexical categories, $@f(a_1)@...@f(a_n)@$, on the tape. Lexical categories in G are of finite size. By definition of a Curried CCG, $|f(a_i)| \leq (kj + 3k + 1)$ symbols because each a_i is a category of the form $A|\{A_1, ...A_j\}|\{A_{j+1}, ..., A_{2j}\}|...|\{..., A_{kj}\}$ where j is the maximum number of arguments in each multiset in the lexical categories of G and k is the maximum number of multisets in the lexical categories of G. Thus, M will use a linearly bounded tape length of $n(kj+3k+1)$ to write the lexical categories of each word.[4]

Then, M nondeterministically chooses pairs of categories on the tape and applies the rewriting rules in R. For example, a pair matching the left hand side of the rule below is rewritten as the category on the right hand side of the rule.

$$X|(U \cup \{\vec{Y}\}) \quad Y|V_1|V_2|...V_k \;\rightarrow\; X|(U \cup V_1)|V_2|...V_k$$

Notice that the length of the category on the right hand side of the rules is always less than the length of the pair of categories on the left hand side because it does not include the nonterminal Y. Thus, the length of tape needed decreases every time M rewrites a pair of categories. If M reaches the state $(q, \#@S@\#)$ then it halts and accepts the string. If M reaches a state where no further rules in R can apply, then it halts and rejects the string. We know that M will eventually halt because G is a lexicalized grammar and thus the length of any derivation in G is linearly bounded by the $|w|$. In addition, we have shown that M will halt using a linearly bounded length of tape. Thus, Curried Multiset-CCG is context sensitive. Since the weak generative capacity of Pure and

[4] If there is lexical ambiguity, all the categories for each word can be written on the tape, and the tape length will have to be multiplied by a constant, the maximum number of lexical categories for each word in the lexicon.

Prioritized Multiset-CCGs is strictly less than Curried Multiset-CCGs, they are context-sensitive grammars as well.

4.1.5 Polynomial Time Parsing for Multiset CCG

In this section, I present a polynomial-time parsing algorithm for Pure and Prioritized Multiset-CCGs. I conjecture that Curried Multiset-CCG can also be parsed in polynomial time by combining my method with the method used by (Vijay-Shanker and Weir, 1993) to parse CCGs.

I extend a CKY algorithm (Kasami, 1965; Younger, 1967) to allow parsing in polynomial space and time for Multiset-CCGs. Although the multisets in Multiset-CCG categories are unbounded in size, they can be stored in finite space if the indices in the multiset come from a fixed alphabet, (Parikh, 1966; Rambow, 1994a).
Given some fixed (finite) alphabet $\Sigma = \{a_1, ..., a_n\}$ and a multiset s over this alphabet, a one-dimensional array $Inds_s$ can be defined such that:

$$Inds_s[a_i] = m_i, \text{ where } m_i \text{ is the number of occurrences of the symbol } a_i \text{ in } s \text{ for } 1 \leq i \leq n.$$

For example, if $\Sigma = \{NP, N, S\}$, the Multiset-CCG category $S|\{NP, NP\}$ can be represented in the CKY matrix as $< S, 200 >$, and $S|\{S, NP\}$ as $< S, 101 >$.

Multiset-CCG is defined as (V_N, V_T, S, f, R), where V_N and V_T are finite sets of nonterminals (basic categories) and terminals (words) respectively, and f is a lexical assignment function that maps elements of V_T to finite subsets of $C(V_N)$, the set of categories. The lexical categories can be functions that specify a multiset of arguments. However, in the lexicon, the multisets are of a fixed size. They contain a maximum of k arguments that are members of V_N, a finite set.[5] Thus, we can use a subset of V_N, those nonterminals that are arguments in lexical categories, $args(V_N)$, as the finite alphabet to index the $Inds$ vectors in the CKY algorithm.

Figure 4.1 shows the CKY algorithm for Pure Multiset CCG. The CKY algorithm uses dynamic programming to construct an upper triangular matrix t indexed 0 through n where each entry contains a partial bottom-up derivation of the input. Given an input $a_1, ...a_n$, entry $t_{i,j}$ contains the set $\{A | A \stackrel{*}{\Rightarrow} a_{i+1}...a_j\}$. As in all CKY algorithms, there are three main loops. First, we place the lexical categories for each input symbol along the diagonal. Then, we try to combine these categories using the forward combination rules (at a) or the backward rules (at b). We can place restrictions on which categories can use the rules, e.g. the direction arrows, but I have omitted these for the sake of simplicity. The rules check if B is an argument in the multiset of arguments associated with A, i.e. $Inds_U$. If so, we delete the B constituent from the multiset and take the union of the multisets of both constituents. The parse is complete when we find a complete sentence, as indicated by the entry $< S, Inds_\emptyset >$ representing the category S with no

[5] The nonterminals include atomic categories such as NP as well as simple functions such as $S|\{NP\}$ which is often represented as an atomic category VP in other grammars.

arguments left to find.

For j := 1 to n do
  $t_{j-1,j} = \{< A, Inds_s > |$ where $f(a_j) = A|s\}$
  For i := j-2 downto 0 do
    For k = i+1 to j-1 do
      a. For every $< A, Inds_U > \in t_{i,k}$ and
          For every $< B, Inds_V > \in t_{k,j}$ where $Inds_U(B) > 0$,
            Let $Inds_U(B) = Inds_U(B) - 1$.
            If $Inds_V = \emptyset$,
              Put $< A, Inds_U > \in t_{i,j}$, if it is not already there, $(>)$.
            Else Put $< A, Inds_U + Inds_V > \in t_{i,j}$, if it is not already there, $(> B)$.
      b. For every $< A, Inds_U > \in t_{k,j}$ and
          For every $< B, Inds_V > \in t_{i,k}$ where $Inds_U(B) > 0$,
            Let $Inds_U(B) = Inds_U(B) - 1$.
            If $Inds_V = \emptyset$,
              Put $< A, Inds_U > \in t_{i,j}$, if it is not already there, $(<)$.
            Else Put $< A, Inds_U + Inds_V > \in t_{i,j}$, if it is not already there, $(< B)$.
If $< S, Inds_\emptyset > \in t_{0,n}$ then accept; else reject.

Figure 4.1: A CKY Algorithm for Pure Multiset-CCG.

The three outer loops in this algorithm are each executed n times, as in a standard CKY algorithm.[6] The inner loops involve three searches among the entries at a certain position in the chart. Thus, the complexity of this algorithm crucially depends on the number of different entries that are possible in one position in the chart. This depends on the combinatorics of the possible categories during a Multiset-CCG derivation. At first glance, it seems as though there are infinitely many different derived categories possible in the grammar because the multiset of arguments can be unbounded in size. However, we can show that the multisets are linearly restricted by the size of the input to the algorithm.

Since Multiset-CCG is a lexicalized grammar where lexical categories can have a maximum of k arguments, the number of elements added to a multiset associated with S during a bottom-up derivation is linearly bounded by k times the length of the input n, in other words $O(kn)$. Thus, the maximum number of pairs we need to save in each entry in the CKY algorithm's matrix for Pure Multiset-CCG is $|V_N|(kn)^{|args(V_N)|}$, because there are kn choices for each of $|args(V_N)|$ positions in the vector that represents the multiset of arguments. We need to look through all pairs in the $t_{i,k}, t_{k,j}$, and $t_{i,j}$ entries in the chart whenever we aply the rules in the inner two loops of the algorithm. Thus, the worst case time complexity for the CKY algorithm for Pure Multiset-CCGs is $\boldsymbol{O(n^{3+3|args(V_N)|})}$ with a constant $|V_N|^3 k^{3|args(V_N)|}$ that is determined by the grammar.

[6] For context-free grammars, the CKY algorithm runs in $O(n^3)$.

This is the worst-case time for wildly scrambled sentences with an unbounded number of clauses. For many grammars and inputs, the algorithm will actually be much faster because the multisets will contain much less than $|args(V_N)|$ arguments. The multisets in the lexical categories start out with at most k arguments each. During the derivation, the composition rules can lead the multisets in the derived categories to grow unboundedly, but the composition rules are only necessary to handle long distance scrambling. In sentences without long distance scrambling, only the application rules, which decrease the size of the multiset, are used. For these inputs, the multisets stored in the chart are always less than or equal to k in size, giving a worst case runtime of $O(n^{3+3k})$. Thus, the average runtime for the algorithm will be much faster than the worst case runtime which handles unbounded long distance scrambling.

We can easily extend this algorithm for Prioritized Multiset-CCG by having the entries keep track of two multisets, $< A, [Inds_1, Inds_2] >$. Then, the maximum number of pairs we need to save in each entry in the CKY algorithm's matrix is $|V_N|(kn)^{2|args(V_N)|}$. The time complexity of this algorithm is $O(n^{3+6|args(V_N)|})$. Curried Multiset-CCG cannot be handled by this algorithm because it can have an unbounded number of multisets in the categories. I conjecture that Curried Multiset-CCGs can be parsed in polynomial time by combining the method above with the structure-sharing method used by (Vijay-Shanker and Weir, 1993) to parse CCGs containing categories with an unbounded stack of arguments.

This algorithm is very similar to the (Rambow, 1994a) algorithm for Restricted {}-LIG, an LIG based formalism in which each nonterminal is associated with a multiset instead of a stack. Rambow proves that if the {}-LIG is polynomially restricted such that the number of index symbols added during the derivation is polynomially bounded by n, the size of the input, the number of possible pairs to be saved in the CKY chart is also polynomially bounded by n, i.e. $|V_N|q(n)^{|V_I|}$, where V_I is the finite set of indices defined for {}-LIG. The overall time complexity of the CKY algorithm for polynomially restricted {}-LIGs is $O(n^2 + q(n)^{|V_I|})$, (Rambow, 1994a). In the next section, we will see that there is a version of Multiset-CCG that has the same weak generative capacity as {}-LIG.

4.2 Comparison to Other Formalisms

In this section, I compare Multiset-CCGs with previous syntactic formalisms proposed to handle "free" word order languages in computational linguistics. In Section 1, I compare Multiset-CCG with various ID/LP formalisms (that distinguish between Immediate Dominance and Linear Precedence relations) used in computational linguistics, i.e. GPSG, HPSG, CG, LFG, and TAGs. In section 2, I compare Multiset-CCG with other lexicalist formalisms. I concentrate on the categorial formalisms of (Bouma, 1985; Karttunen, 1989) that have been proposed for "free" word order languages, and Tree-Adjoining Grammars.

4.2.1 ID/LP Approaches

Many approaches to "free" word order languages separate their grammar into two components: immediate dominance (ID) and linear precedence (LP) rules. In this section, I discuss the use of the ID/LP distinction in GPSG, HPSG, categorial formalisms, LFG, and TAGs. In Chapter 6 page 166, I also compare the ID/LP formalisms that capture pragmatic information such as the ordering of topic and focus with the extended Multiset-CCG which also captures the ordering of these information structure components.

4.2.1.1 Generalized and Head-Driven Phrase Structure Grammars

(Pullum, 1982; Gazdar et al., 1985) introduce the first ID/LP formalism using Generalized Phrase Structure Grammar (GPSG). (Uszkoreit, 1987) presents a GPSG framework that handles German word order variation in simple clauses. He develops ID metarules, rule extension principles, and LP rules that successively apply to generate a grammar of context-free phrase-structure rules. The ID rules are order-free, while the LP rules contain syntactic as well as pragmatic ordering information for sisters. Uszkoreit introduces disjunctive LP rules such as $+NOM < +ACC$, *+Pronoun < -Pronoun, -Focus < +Focus* which can be violated, as long as at least one of the rules hold true. Uszkoreit's constrained ID and LP rules do not increase the context-free weak generative capacity of the formalism, since they are used to generate a finite (albeit large) set of context-free rules. However, Uszkoreit does not investigate long distance scrambling which would require a more powerful formalism.

(Gunji, 1987) develops a similar extension of GPSG for Japanese, called JPSG. In his approach, the subcategorization list of a verb is represented as an unordered set to capture local scrambling; however, the grammar cannot handle long distance scrambling because there is no way to combine the argument sets of two verbs so that their arguments can occur in a mixed order. Gunji resorts to using the SLASH feature, which can be passed across S boundaries, to capture long distance

scrambling in JPSG. Thus, two separate mechanisms are used to handle local and long distance scrambling. Furthermore, the SLASH mechanism can only handle the extraction of one argument. Thus, JPSG cannot handle unbounded scrambling where more than one NP occurs out of its clause (e.g."$(NP_1...NP_n)_{\text{scrambled}}$ V_n ... V_1").

The same ID/LP approach has been used in Head-Driven Phrase Structure Grammar (HPSG) which is a descendent of GPSG (Pollard and Sag, 1987). (Steinberger, 1994) extends Uszkoreit's use of a disjunctive LP rule for a German HPSG. (Engdahl and Vallduvi, 1994) present an HPSG for Catalan, another "free" word order language, using dislocation rules and LP constraints ordering the information structure components.

(Zwicky, 1986) extends the ID/LP formalism introduced by (Pullum, 1982) for GPSG by adding *liberation* rules. The indirect liberation rules can be seen as metarule that operate on ID rules to flatten the constituent structure by tree-pruning. (Zwicky, 1986) introduces direct liberation rules that are a part of the grammar proper; these are like wrapping rules (Bach, 1988), but without ordering restrictions. For example, local scrambling of subjects and objects in transitive sentences can be handled by eliminating/liberating the VP node by collapsing two ID rules (S $\rightarrow$ NP, VP, and VP $\rightarrow$ NP, V) into one order-free ID rule, (S $\rightarrow$ NP, NP, V). Similarly, long distance scrambling can be handled by liberating the embedded S node. This formalism can handle island behavior by deriving the standard hierarchical structure for certain clauses, as well as discontinuous constituents through the use of the liberation rules to derive a flat order-free structure for certain constituents.

Multiset-CCG is similar to these GPSG and HPSG approaches in that linear order is usually not specified in verbal subcategorization. However, I place syntactic restrictions on word order, such as island behavior or that NPs precede verbs in languages such as German, in the syntactic part of the grammar. The ID/LP approaches generally place these restrictions in the LP part of the grammar. They divide the grammar into dominance and precedence relations, while I argue that the division should be between the predicate-argument structure and the information structure. In addition, most ID/LP approaches, except for (Zwicky, 1986) and followers, do not handle unbounded long distance scrambling and island restrictions. Multiset-CCG can uniformly capture local and unbounded long distance scrambling and syntactic restrictions upon word order.

4.2.1.2 ID/LP Categorial Formalisms

(Reape, 1991) develops an ID/LP formalism for German which combines HPSG and Categorial Grammars so that both the functional and the semantic head of constituents are specified. He defines the *word order domain* of a constituent as the sequence of leaves in the constituent's subtree. His LP constraints are well-formedness constraints on the word order domains. He

allows a feature [unioned: +/-] to indicate whether two word order domains can be collapsed into one; a [unioned: -] feature assigned to certain heads allows the grammar to capture island behavior in German.

(Hoeksema, 1991) investigates adding liberation rules (which allows flat structure for certain constituents) as in (Zwicky, 1986) to a categorial grammar in order to capture free word order and complex predicates. (Dowty, 1989 revised 1991) takes this idea a step further with a categorial grammar in which flat structure is the default; LP rules order the categories, and hierarchical structure is only added by attachment rules postulated to keep certain categories together and a list of bounding nodes that do not allow discontinuity. This is very similar to Reape's [union: -] feature and to the integrity constraints in FO-TAGs that will be discussed in the next section.

Multiset-CCG is similar to these approaches in that syntactic restrictions on "free" word order, such as island phenomena are captured by the syntactic "ID" part of the grammar, instead of LP rules. Multiset-CCG prevents the arguments in the island clause from scrambling into the matrix clause by assigning curried function categories to the head of an island clause; this imposes a certain hierarchical structure during the derivation. (Zwicky, 1986; Hoeksema, 1991; Dowty, 1989 revised 1991; Reape, 1991) also capture island phenomena by imposing a hierarchical structure for certain constituents. However, Multiset-CCG provides a lexicalist representation for these grammatical restrictions.

4.2.1.3 Lexical Functional Grammar

Lexical Functional Grammar (LFG) (Bresnan and Kaplan, 1982) is a lexicalist formalism which separates phrase structure information about constituents, the *c-structure*, from the information about grammatical functions, the *f-structure*. C-structure is built from language-specific phrase-structure rules which encode dominance and precedence information. Annotations on these rules indicate features of the possible f-structures associated with the sentence. For a sentence to be derived, it must form a tree rooted at S in the c-structure and have a well-formed f-structure. An f-structure is well-formed if it has found all of the required arguments, i.e. is *complete*, and has no extra grammatical functions that are not governed by a predicate, i.e. is *coherent*.

(Mohanan, 1982) develops an LFG for Malayalam, a "free" word order language. He captures local word order variation in Malayalam by the c-structure rule $S \rightarrow X'^{*} \quad V$. This rule expands S into any number of constituents followed by a verb. The X constituents are restricted to the set {*N', P', ADV', S'*}; they cannot be an adjective or another verb. This rule produces a flat structure for clauses in Malayalam. Lexical categories for the verbs specify the obligatory arguments needed to form a complete f-structure, but do not specify the linear order of these arguments.

(King, 1993) develops an LFG (as well as a GB) approach to capture word order variation

in Russian. She associates phrase structure positions with discourse functions (much like (Kiss, 1987) except within a nontransformational theory, LFG) through the use of more complex phrase-structure rules and linear precedence rules. The annotations on her c-structure rules create f-structures which represent discourse functions such as topic and focus. The LP rules specify the ordering of discourse functions, e.g. Topic < Focus; they ensure that the constituents matching these discourse functions are ordered correctly in the c-structure of the sentence.

The Multiset-CCG approach is similar to the LFG approach in that the linear order of a verb and its arguments as well as the adjuncts is not specified in the syntax rules or in the lexicon. Although the LFG approaches do not deal with long distance scrambling, (King, 1993) mentions the use of functional uncertainty to handle long distance dependencies. For example, topicalized constituents in English which are placed in the sentence-initial position even though they belong in an embedded clause can be captured with the following rule:

(4) S → XP S
(↑ TOP) = (↑ {COMP,XCOMP}*(GF-COMP)) ↑=↓

The f-structure annotation on XP uses functional uncertainty to indicate that it can be the complement (GF-COMP) of a clause embedded (arbitrarily many times) in S. Thus, LFG can handle long distance scrambling through the use of functional uncertainty.

The LFG approach in (King, 1993) for integrating discourse information with syntax is very similar to my approach that will be presented in Chapter 6. The surface structure of a sentence in both of our approaches directly reflects the information structure of the sentence. This will be discussed further in Chapter 6, page 166.

4.2.1.4 Free-Order Tree-Adjoining Grammars

Tree Adjoining Grammars (TAGs) (Joshi, Levy, and Takahashi, 1975) separate the recursion found in a grammar from the local co-occurrence relations and dependencies. A TAG is a mildly context-sensitive grammar (Joshi, Vijay-Shanker, and Weir, 1991) which consists of a set of elementary trees and an adjunction operation which together provide an extended domain of locality not found in context-free grammars.

Elementary trees within lexicalized TAGs can be divided into two sets: initial trees representing minimal linguistic structures associated with at least one lexical item (e.g. a verb and its subcategorization frame) and auxiliary trees representing constituents that are adjuncts to the basic structures. Adjunction is a recursive operation that builds new trees by inserting an auxiliary tree into the middle of an elementary tree at nodes which can be unified. Long distance dependencies can be expressed in a TAG since adjoining a tree in the middle of an elementary tree expands that tree but does not change the dependencies encoded in the tree.

Free-Order TAG (FO-TAG) (Becker, Joshi, and Rambow, 1991) is an ID/LP formalism where the elementary trees only indicate the dominance relations, but not the linear order among the head and its arguments. The leaves of the derived tree can occur in any order that is acceptable by the LP rules. Since the LP rules order the leaves, not just the sisters, in the tree, FO-TAGs can handle long distance scrambling. To capture islands in long distance scrambling, (Becker, Joshi, and Rambow, 1991) define an integrity constraint over subtrees which disallow elements in a marked subtrees to occur outside of the tree. The lexical representations for verbs in FO-TAGs is very similar to Multiset-CCGs, since in Multiset-CCGs, the verbs subcategorize for a set of arguments whose linear order is usually not specified. Both formalisms can capture long distance scrambling and island restrictions upon scrambling in the syntactic component of the grammar. However, Multiset-CCG also handles syntactic restrictions such as the fact that NPs precede verbs in German in the syntactic categories whereas FO-TAG can only capture this in the LP rules. Multiset-CCG lexicalizes syntactic restrictions in word order; the lexical categories can specify the directionality of their arguments without fixing their relative order, or even order some of their arguments with respect to others through the use of prioritized multisets.

4.2.2 Other Lexicalist Formalisms

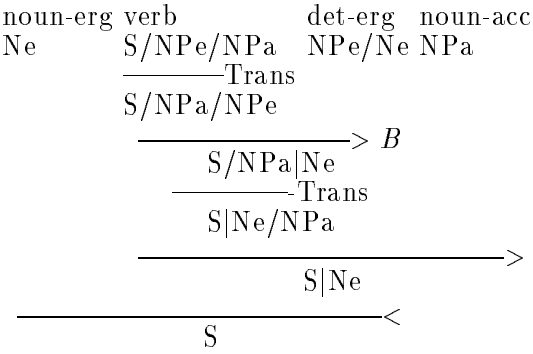
4.2.2.1 Bouma's Categorial Grammar

Multiset-CCG closely resembles Bouma's categorial grammar in (Bouma, 1985) for Warlpiri. His categorial grammar is very similar to CCGs in that it has function application, composition, and type-raising, although Bouma alters the combinatory rules slightly to allow the result of composition to be non-directional. Bouma is able to capture free word order in Warlpiri by adding a type-changing rule to the grammar. The rule below, which he calls *transitivity*, allows arguments to change positions within a category:

(5) $(A/B)/C \Rightarrow (A/C)/B$

Thus, the function category for a verb can change into the category that is appropriate for every possible word order permutation of its arguments. Bouma can handle local scrambling as well as long distance scrambling by allowing verbs to compose together and then applying the transitivity rule to the resulting constituent.

This category-changing rule can apply more than once to a category during the course of a derivation. For example to capture discontinuous NPs, a Warlpiri verb may have to change its category more than once in order to combine with parts of an NP in a mixed order.

(6)


Discontinuous NPs are ungrammatical in Turkish; thus, an unrestricted transitivity rule would overgenerate unwanted word orders for Turkish. In Multiset-CCGs, we restrict the composition rules in order to avoid the overgeneration of discontinuous NPs. The same restriction on composition to disallow the composition of adjectives and verbs would work in Bouma's grammar as well.

The set notation in Multiset-CCGs could be seen as just a notational convenience which captures all the possible category changes for a verb. This is equivalent to a transitivity rule which applies only in the lexicon. In fact, Pure Multiset-CCGs and Bouma's CG are weakly equivalent. The use of any rule in a Pure Multiset-CCG derivation could be simulated by using the equivalent rule in Bouma's CG followed by the use of the transitivity rule. Bouma's curried function categories do not really specify linear order restrictions because the transitivity rule can

apply at any time during a derivation and change the verb's linear order specification. Thus, Bouma's lexical categories for verbs is equivalent to the Multiset-CCG representation which does not specify the ordering of the arguments.

Although both approaches capture the same word order variation, the set notation is convenient in expressing this variation in the lexical representation itself rather than resorting to an extra process of type-changing through the application of a special rule. This is useful in expressing lexical restrictions in word order. For example, in Multiset-CCGs, we can capture the fact that adjunct clauses with certain lexical heads are islands in Turkish by assigning those heads a curried category restricting the order in which they combine with their arguments and the matrix clause. In Bouma's grammar, we could not specify this restriction in the lexicon; we would have to restrict the transitivity rule from applying to each of those lexical heads.

In fact, Bouma's CG is not equivalent to Prioritized or Curried Multiset-CCGs because it cannot preserve the ordering among some or all of the arguments. We could restrict the transitivity rule not to apply to some functions, but there is no way to specify that some of a function's arguments can scramble while other arguments cannot since the transitivity rule can permute any of the arguments in a function. Thus, Bouma's CG cannot handle the islands and object-incorporation data in Turkish that Multiset-CCG captures by using prioritized multisets.

4.2.2.2 Categorial Unification Grammar

(Karttunen, 1989) proposes a categorial analysis of Finnish, using Categorial Unification Grammar (CUG) (Uszkoreit, 1986), which handles free word order by treating noun phrases as functors that apply to the verbal basic elements. This is much like the use of type-raised noun phrases in CCGs. For instance, the noun Mary with the nominative morpheme might be defined as the following set of features in the CUG. In Finnish (as in Turkish), NPs can occur to the left or to the right of the verb, as indicated by the features left and right below. Rightward combination with a verb is constrained in that the verb must be declarative.

$$(7)\ \text{Mary} := \left[\begin{array}{lcl} argument & : & \#1\left[\begin{array}{lcl} cat & : & v \\ syntax & : & \left[subj : \left[\begin{array}{lcl} cat & : & N \\ case & : & nom \\ sem & : & Mary' \end{array}\right]\right] \end{array}\right] \\ left & : & [] \\ right & : & [sem : [type : declarative]] \\ result & : & \#1 \end{array}\right]$$

In Karttunen's analysis, the verb is not a functor at all, but a basic element V (except subordinate verbs are active functors like the NPs). The category V is a set of features. Each verb specifies its arguments in a subcat feature, but the linear order of the arguments is not specified. Through the application rules, the arguments combine with the verb in any order. The following example is a feature set for the transitive verb "eat":

$$(8)\ \text{eat} := \left[\begin{array}{lcl} cat & : & v \\ syntax & : & \left[\begin{array}{lcl} subj & : & \left[\begin{array}{lcl} cat & : & N \\ case & : & nom \\ sem & : & \#1 \end{array}\right] \\ obj & : & \left[\begin{array}{lcl} cat & : & N \\ case & : & acc \\ sem & : & \#2 \end{array}\right] \\ vcomp & : & NONE \end{array}\right] \\ sem & : & \left[\begin{array}{lcl} scene & : & eating \\ agt & : & \#1 \\ pat & : & \#2 \end{array}\right] \end{array}\right]$$

In CUGs, the application rule allows the noun functor to combine with the basic element verb by unifying the argument feature in the noun feature-set with the verb feature-set. If these are unifiable, the resulting constituent has the features of the noun functor's result feature. The result feature now includes the features in the verb due to the coindexation (#1) of the result feature

with the argument feature in the noun's category.

Karttunen proposes the use of *functional uncertainty* in the category definition for the NPs to handle long distance scrambling in Finnish. Noun functors are given a floating type. For example, the NP in (7) will look for a verbal argument which has a feature matching the regular expression ((syntax vcomp)* syntax subj); this means that the NP may be the subject of a verb which is embedded indefinitely many times in the complex verb category. Thus,Kartunnen's approach can handle long distance scrambling with an unbounded number of embedded clauses.

CUG and Multiset-CCG are very similar in that the verb's subcategorization does not indicate the linear order of its arguments. In Multiset-CCGs, we choose to directly represent the verb as a function with an order-free argument set, whereas in CUG the arguments of a verb are represented as a feature in its basic element category. An advantage of my approach over Karttunen's is that at the end of a parse, we do not need an extra process to check if all the arguments of a verb have been found; this falls directly out of the combinatory rules.

CUG represents nouns as functors (V/V or V\V) looking for a verb which subcategorizes for it. This is very similar to CCG's type-raised categories, however the CCG type-raised category is $v/(v\backslash Nnom)$ and must use composition rules because the verb is represented as a functor as well. Karttunen argues that only function application, not composition, is necessary to capture the linguistic facts. This makes the grammar much simpler and avoids the multiple derivations that composition rules may allow. However, without composition rules, we cannot handle coordination of NPs since this requires the composition of two NP functors in a categorial system. In addition, the lack of composition would result in an incorrect analysis for the following Turkish word order. The first bracketing below involves local scrambling to the right of the embedded verb in Turkish, while the second involves long distance scrambling and is ungrammatical; the two derivations are distinguished in Turkish by stress and intonation.

(9) a. NP1 [V2 NP2] V1.

b. *NP1 [V2] NP2 V1.

In fact, CUG would allow the ungrammatical derivation since NP2 and V1 can combine together through the use of functional uncertainty. Moreover, CUG would not allow the grammatical derivation because NP2 and V2 are both assigned functor categories which cannot combine together. The two functor categories could only combine together through the use of function composition rules.

4.2.2.3 Vector Tree-Adjoining Grammars

Within the TAG framework, different approaches have been developed to handle scrambling. (Becker, Joshi, and Rambow, 1991) have found that ordinary TAGs cannot handle scrambling while instantiating co-occurrence constraints (i.e. enforcing that a predicate and all of its arguments occur in the same elementary tree). They have proposed a way to use (non-local) Multi-Component-TAGs to handle long-distance scrambling as well as an ID/LP TAG called Free-Order TAGs, discussed in the last section. (Rambow, 1994a) proposes a formalism called Vector-TAG that is similar to Multi-Component TAGs.

Multi-Component TAG, MC-TAG, (Joshi, Vijay-Shanker, and Weir, 1991; Weir, 1988) is an extension of the TAG formalism (described on page 97) with a greater weak generative capacity. MC-TAGs consist of *sets* of interdependent auxiliary trees and an adjoining operation which is extended so that the auxiliary trees in a set are all adjoined simultaneously into different nodes in another tree or tree set. Three variants of MC-TAGs have been defined.

- In Tree-local MC-TAGs, trees in an auxiliary tree set are simultaneously adjoined to different nodes in the *same initial* tree; this formalism is equivalent to ordinary TAGs in generative power, but increases the descriptive power by increasing the domain of locality over which dependencies can be stated.
- Set-local MC-TAGs slightly increase the generative power by allowing set to set adjunction; trees in an auxiliary tree set are simultaneously adjoined to distinct nodes of *any* member tree of a *single tree set*.
- Non-local MC-TAGs are the most powerful; they allow the trees within an auxiliary tree to simultaneously adjoin into different trees all together (i.e. this is the same as adjoining into derived trees).

Set-local MC-TAGs have been used in handling a variety of linguistic phenomena,(Kroch and Joshi, 1986; Heycock, 1987). However, it is still an open question whether natural languages require the extra generative power of MC-TAGs.

(Rambow, 1994a) proposes the V-TAG (Vector-TAG) formalism for capturing scrambling in German that relaxes immediate dominance constraints rather than relaxing the linear precedence constraints of TAGs. V-TAG is similar to a non-local MC-TAG, but the restriction on simultaneous adjunction has been lifted. (Rambow and Satta, 1992) have shown that non-local MC-TAGs generate NP-Complete languages, but Rambow has shown that V-TAG can be polynomially parsable.

In V-TAG, each tree set consists of an auxiliary or elementary tree anchored by the verb, and auxiliary trees for each argument of that verb. In addition, dominance links, which cannot

be broken during adjunction, are defined between the trees in the set. Thus, the directionality of arguments can be specified without specifying their relative order. Clausal subcategorization in this approach is handled by adjunction of the auxiliary tree anchored by the matrix verb into the elementary tree for an embedded verb; nominal subcategorization is handled by simple substitution of lexical items into the auxillary trees for the arguments. In order to simulate local scrambling, the auxiliary trees for the arguments adjoin into the tree for the verb in their set. If there is long distance scrambling, the auxiliary trees adjoin into the trees for verbs in other tree sets.

A similarity in lexical representations is seen in the auxiliary trees for nominal arguments in V-TAGs and the type-raised arguments in CCGs. A type-raised NP indicates that it is looking for some type of verbal function to its right to result in a verbal function, just like the auxiliary tree for NP arguments in V-TAGs.

Rambow conjectures that V-TAGs are weakly equivalent to []{}-LIGs, an LIG in which non-terminals are associated with a stack and a set. The set is used for vertical dependencies between scrambled arguments and their verbs, while the stack is used to record dependencies between parts of trees where each pair of tree parts is separated by an adjunction; the latter dependencies cannot be represented in a set that can be distributed among daughters because the strict order of the dependencies must be preserved in a sequence of adjunctions. It may turn out that Multiset-CCG with curried functions is also equivalent to these formalisms. Further research is necessary to determine the relationship between V-TAGs and Multiset-CCG.

4.3 Summary

I have shown that the three versions of Multiset-CCGs are within the class of context-sensitive grammars in weak generative capacity and are polynomially parsable. They have a slightly different weak generative capacity than the mildly context-sensitive TAGs, CCGs, and LIGs, but they do not have the full power of context-sensitive grammars. Figure 4.2 shows the conjectured relationships between these grammars based on their weak generative capacity.

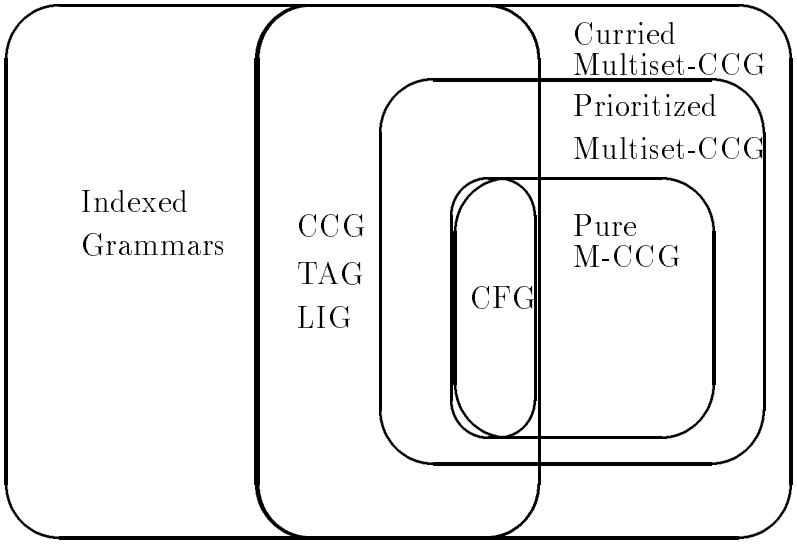


Figure 4.2: The Power of Multiset-CCGs

In this chapter, I have given a polynomial time parsing algorithm for Pure and Prioritized Multiset-CCGs. Thus, I have shown that the Prioritized Multiset-CCG developed for Turkish in Chapter 3 is a computationally attractive formalism. In addition, I have compared Multiset-CCG with some previous syntactic formalisms proposed for "free" word order languages. We have seen that Multiset-CCG has a wider coverage than most of these formalisms because it can handle unbounded scrambling with long distance dependencies and syntactic restrictions on scrambling such as island phenomena. In the next chapters, I extend Multiset-CCG to capture the pragmatic interpretations associated with "free" word order. The formal generative capacity of this extended formalism will be discussed in Chapter 6, page 168.

Chapter 5

The Discourse Functions of Turkish Word Order

The goal of this chapter is to investigate naturally occurring data in order to determine the discourse functions of word order in Turkish. In the previous chapter, I presented a categorial grammar, Multiset-CCG, that can capture the syntax of "free" word order in Turkish. In this chapter, I discuss *why* speakers choose a certain word order in a certain context and what additional meaning these different word orders provide to the hearer.

Word order variation in relatively free word order languages is used to convey distinctions in meaning that go beyond truth-conditional semantics. For example, the different word orders in Turkish sentences below are translated to English using different stylistic constructions and prosodic focus indicated by capital letters in order to approximately capture their different meanings.

(1) a. Esra kitab-ı oku-yor. (SOV)
Esra book-Acc read-PresProg.
"Esra is reading the BOOK."

b. Kitabı Esra okuyor. (OSV)
Book-Acc Esra read-Prog.
"As for the book, it is ESRA who is reading it."

c. Okuyor Esra kitabı. (VSO)
Read-Prog Esra book-Acc.
"She is READING it, the book, Esra."

Word order serves to structure the information being conveyed to the hearer, e.g. by indicating what is the *topic* in the sentence-initial position, and the *focus* in the immediately preverbal

position. In this chapter, I investigate naturally occurring data as well as question-answer pairs in order to define these terms and to develop a representation for the information structure of Turkish sentences as reflected by word order. The examples and statistical results in this chapter are based on a corpus of naturally occurring discourses that contains transitive sentences with differing word orders, collected from the CHILDES corpus (MacWhinney and Snow, 1985), colloquial speech that I have transcribed, and contemporary novels.

In the next section, I review the previous approaches to representing the information structure of sentences. Then in section 2, I present the information structure representation that I will be using in order to capture the interpretation of word order in Turkish question-answer pairs. In section 3, I present naturally occurring data and investigate the interaction between word order and referential form, familiarity, and saliency, in order to support my representation of information structure in Turkish. In section 4, I discuss the information structure of complex sentences that contain embedded clauses and the interpretation of long distance scrambling. The theory that I develop in this chapter about the information structure in Turkish will be integrated with the Multiset-CCG formalism in the next chapter. The theory will be tested by determining whether Turkish sentences with word orders appropriate to the context can be automatically generated within a database query task.

5.1 Previous Representations of IS

There is little agreement on how to best represent the information structure (IS) of a sentence. Among the competing theories for an information structure representation, many different primitives have been proposed. All of the following approaches divide a sentence into separate segments in slightly different ways according to the information content of the segments. (Vallduví, 1990) presents a very good comparison of the competing approaches.

- Theme/Rheme (Halliday, 1967; Steedman, 1991).
- Topic/Comment (Kuno, 1976; Reinhart, 1981; Gundel, 1985; Erkü, 1983).
- Topic/Focus (Sgall, Hajicova, and Benesova, 1973).
- Focus/Presupposition (Chomsky, 1971; Jackendoff, 1972), Focus/Open-proposition (Prince, 1981a; Prince, 1986).
- Link/Focus/Tail (Vallduví, 1990), Topic/Focus/Background (Erguvanlı, 1984).

While some of these accounts are compatible, there are certain sentences which bring to mind one account or another as the most natural one to describe that sentence. For instance, the topic/comment type of structure seems the natural way to describe the sentence in (2)b where (2)a is given as the context, but the focus/open-proposition type of structure seems the natural way to describe (2)c. In the former sentence, Joan, the discourse entity that "she" refers to, is the sentence *topic* because it is the main element that the sentence is about. In the latter, Carol is the focus of the sentence because it is new or important information that is stressed.

(2) a. Joan spent the day at home today.

b. She$_{topic}$ (wanted to lie in bed and read books all day)$_{comment}$.

c. It was CAROL$_{focus}$ who gave her that book.

These sentences point out the shortcomings of theories which only separate a sentence into two parts (e.g. topic/comment, or focus/open-proposition); these theories can only handle the pragmatic distinctions in one or the other of the sentences but not both. Recognizing this, many theories further split a sentence into three or four IS components.

For example, in Steedman's approach, based on (Halliday, 1967), a sentence is divided into a theme and rheme (which is similar to the topic/comment distinction) based on intonational phrasing, and then an element is marked as focus in each theme or rheme, based on the pitch-accents. The example below (from (Prevost and Steedman, 1993)) generates an answer with the appropriate prosody reflecting this information structure in the context of a question.

(3) Q: I know that the OLD widget had a SLOW processor.
But what processor does the NEW widget include?

A:	(The	NEW	widget includes)		(a	FAST	processor).
		L+H*	LH%			H*	LL%
	Ground	*Focus*	*Ground*		*Ground*	*Focus*	*Ground*
		Theme				*Rheme*	

The pragmatics of word order in Turkish has been studied by (Erguvanlı, 1984) and (Erkü, 1983). Erguvanlı presents a functional approach to word order variation in Turkish in which each position in a Turkish sentence is strongly associated with a specific pragmatic function. She identifies the sentence-initial position as the *topic*, the immediately preverbal position as the *focus*, and the postverbal positions as *backgrounded information*. (Erkü, 1983) adopts a Topic-Comment information structure where the *topic* of the sentence can occur either sentence initially or post-verbally, and must refer to a discourse entity that is uniquely identifiable or a member of a uniquely identifiable set. There is also a focused entity within the *comment* component of the information structure, where *focus* is loosely defined as prominent information.

(Vallduví, 1990) provides a theory of information structure, for the "free" word order language Catalan, that has three components as well: *link*, *focus*, and *tail* The *link* and *tail* together form the *ground*, the open proposition, for the *focus* of the sentence. In Turkish and Catalan, in general we first place the information that connects the sentence to the previous context, then the important and new information immediately before the verb, and the information that is not really needed but may help the hearer understand the sentence better, behind the verb. Vallduví describes the information structure as a way speakers package the information to be presented to the hearer. The information structure of a sentence provides instructions to the hearer about entering information into his/her knowledge store.

If we assume that the hearer's knowledge store or discourse model is organized by topics, then the sentence topic, or link in Vallduví's terms, can be seen as specifying an address in the hearer's knowledge store (Reinhart, 1981). The rest of the sentence tells the hearer what to store at this address. Using the file card analogy for discourse entities, Vallduví says that the *link* component of the IS tells the hearer to go to a particular file card; the *focus* is the information the hearer must record on that file card, while the rest of the sentence, the *tail* provides additional information about exactly where on the card to record the information. In Vallduví theory, not all components of the IS need be present in one sentence; only the focus is obligatory. If the speaker uses an IS which does not contain a link, the speaker assumes that hearer has already activated the salient file card or is using an all-purpose situation address to record the information.

The representation that I will use for the IS of Turkish sentences as reflected by word order is described in the next section. I am in debt to various insights of Erguvanlı, Erku, Steedman, and Vallduví in developing this representation.

5.2 My Proposal for an IS Representation

In my representation for the information structure reflected by Turkish word order, I divide each clause into a topic and a comment, (Erkü, 1983), and further divide the comment into a focus and a ground. This representation is notated as below:

(4) $\begin{bmatrix} \text{Topic} & \\ \text{Comment} & \begin{bmatrix} \text{Focus} \\ \text{Ground} \end{bmatrix} \end{bmatrix}$

I associate certain sentence positions with discourse functions as in (Erguvanlı, 1984); the sentence initial position tends to be the topic, the immediately preverbal position tends to be focus, and postverbal elements are in the ground. However, my representation is more like (Vallduví, 1990) in that there is more than one information structure available in the grammar; for example, in some sentences, the verb itself is in focus instead of the immediately preverbal element or there is no sentence-initial element and so the topic must be recovered from the context rather than the word order.

My definitions for topic and focus follow Vallduví's information-packaging theory. The information structure of a sentence provides commands to the hearer about entering information into his/her knowledge store. If we assume that the hearer's knowledge store and the discourse model are organized by topics, then the sentence topic can be seen as specifying an "address" in the hearer's knowledge store (Reinhart, 1981; Vallduví, 1990). The rest of the sentence, the comment, tells the hearer what to store at this address. The informational focus is the most information-bearing constituent in the sentence, (Vallduví, 1990); it is the new or important information in the sentence and often receives prosodic prominence in speech. Everything else in the sentence forms the ground of the sentence.

It is sometimes very hard to identify the topic or the focus of a sentence in natural discourse. This is why I have limited the domain of my theory to describing the contextually appropriate answer to wh-questions and yes/no questions within a database query task. The topic and focus of question-answer pairs are easily identified. As we will see in the next chapter, these notions of topic and focus are useful and in fact necessary in a natural language computer-interface in order to generate appropriate sentences in response to database queries.

In the question-answering task, the topic is the main entity that the question and answer are both about, e.g. "Ahmet" in (5). In Turkish, the topic of the question is most often found in the sentence initial position. If there is no sentence-initial topic realized in the sentence, the topic is inferred from the context. The focus in Turkish is usually placed on the item in the immediately preverbal position. In Turkish questions, the questioned item, which indicates what

information the hearer needs to look up or verify in his/her knowledge store, is focused by stress and word order as seen in (5)a. In the answer to a wh-question, the new information that fills in the questioned wh-element is focused by placing it in the immediately preverbal position, as seen in (5)b. The information structure of the response is shown in (5)c. The canonical SOV word order would not be felicitous in this context, because it would not place the topic and focus of the sentence in the appropriate sentence positions.

(5) a. Ahmet'i kim arıyor?
Ahmet-Acc who seek-Pres.
"As for Ahmet, who is looking for him?"

b. Ahmet'i Fatma arıyor. OSV
Ahmet-Acc Fatma seek-Pres.
"As for Ahmet, it is FATMA who is looking for him."

c. $\begin{bmatrix} \text{Topic :} & \text{Ahmet} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{Fatma} \\ \text{Ground :} & [\text{seek}] \end{bmatrix} \end{bmatrix}$

Identifying the topic of the question is important because it provides a search strategy in the knowledge store or database. If the knowledge store is a set of file-cards organized by topics, the first place to look for the answer to the question is the file-card associated with the question's topic. For example, the direct object "Ahmet" is the main entity that is being talked about in the question-answer pair above and thereby is placed in the sentence-initial position. Upon hearing the question, the hearer looks up the file-card associated with "Ahmet" in his/her knowledge store and reads the information recorded on that file card to determine who called him. "Ahmet" is also the topic of the response and so is generated in the sentence-initial position to indicate to the hearer to record the information in the response under the topic "Ahmet".

In yes/no questions, alternative information can be focused in the answer. In Turkish, the question particle "mi" can be placed next to any element in the sentence to question just that element. It is used to mark the focus of the question much like high pitch and stress is used in English. If the statement in the question is not found in the database, the implemented system provides a more natural and helpful answer by replacing the focus of the question with a variable and searching the database for an alternate entity that satisfies the rest of the question. This alternative is focused in the answer by placing it in the immediately preverbal position:

(6) a. Ahmet'i Fatma dün mü gördü?
Ahmet-Dat Fatma yesterday Quest saw-Past.
"As for Ahmet, was it YESTERDAY that Fatma saw him?"

b. Hayır, Ahmet'i Fatma bugün gördü.
No, Ahmet-Acc Fatma today saw-Past.
"No, as for Ahmet, Fatma saw him TODAY."

The verb itself can also be focused by high pitch and stress or by lexical cues like the placement of the question morpheme. We must have more than one IS available, where verbs can be in the focus as seen below or in the ground component of the IS as in the examples above.

(7) a. Gel-iyor mu Ayşe?
Come-Prog Quest Ayşe?
"Is she, Ayşe, coming?"

b. Hayır, gel-mi-yor Ayşe.
No, come-Neg-Prog Ayşe.
"No, she, Ayşe, is NOT coming."

In the rest of this chapter, I will investigate naturally occurring Turkish data in order to further determine what it means to be the topic, focus, or in the ground component of Turkish sentences.

5.3 The Topic and the Sentence-Initial Position

One of the reasons for word order variation in Turkish is to bring a topical element to the front of the sentence. The sentence-initial element in Turkish sentences, as well as in many other languages, tends to be the "topic". For example, the OSV word order in (8)b is used because the object "it" is the topic of the sentence, since it is what is being talked about in this discourse segment.

(8) a. EXP: bu çok güzel bir şey.
this very pretty one thing.
"This is a very pretty thing."

b. EXP: on-u san-a kim ver-di? (OOSV)
it-Acc you-Dat who give-Past?
"Speaking of this, who gave it to you?" (CHILDES 1ca.cha:374)

Although the terms "topic" and "theme" are widely used, there is no consensus among linguists on what these terms mean. Intuitively, most people say that the topic is what the sentence is about, however, this is a very vague definition. Some of the methods used by linguists to identify the topic of a sentence are listed below, but in fact, none alone is a formal and adequate way to identify the topic of a sentence.

1. the sentence initial constituent (Halliday, 1967; Erguvanlı, 1984).
2. the constituent X for which the sentence can be paraphrased "as for X, ..." (Gundel, 1985; Erkü, 1983).
3. the constituent X for which the sentence answers the question "what about X?"
4. the constituent with a particular intonation (Steedman, 1991).
5. The "address" in the hearer's knowledge store where the information in the rest of the sentence can be stored (Reinhart, 1981; Vallduví, 1990).
6. A uniquely identifiable (i.e. definite), or a member of a uniquely identifiable set, (Gundel, 1985; Erkü, 1983).

Following (Erguvanlı, 1984), I identify the sentence-initial position in Turkish clauses as the sentence topic. I will be adopting the definition of topic given by (Reinhart, 1981). Assuming that the hearer's knowledge store and discourse model are organized by topics, the topic of a sentence provides the "address" in the hearer's knowledge store where the information in the rest of the sentence can be stored. The sentence-initial element in Turkish often contains a salient

discourse-entity that has already been evoked in the discourse or is related to an entity that has already been evoked in the discourse. Thus, sentence topics often are used to keep the discourse coherent by linking the current utterance to the prior context.

For example, the OSV word order in (9)c is used because the object "these people" is what the sentence is about, the topic, and serves to link the current utterance to the previous context, while the subject "much of my goodness" is new, focused information.

(9) a. ∅ Tan-ır-lar ben-i. Hepsi ∅ tanır.
∅ know-Aor-3Pl I-Dat. All-Poss3 ∅ know.
"(They) know me. All of (them) know (me)."

b. **Bun-lar-a** çok iyi-liğ-im dokun-muş-tur.
These-PL-Dat much good-Poss1S touch-ReportedPast-be.
"These people, many of my good deeds have touched them." (Aziz Nesin, *Zubuk*, 1961:305)

The topic does not have to be an NP argument in the sentence; it can be any element that can serve as an "address" in the knowledge store. For example, the scene-setting adverbials in the sentence-initial position in (10)b and (11)b serve as the topic of the sentences. The adjuncts point to certain discourse-referents, while the rest of the sentence provides information about that discourse-referent. However, sentence-initial connectives such as 'and' and 'but' are not analyzed as sentence-topics. They serve a function at a higher level of discourse-processing.

(10) a. Bun-lar hepsi eş ol-du zaten bir-bir-ler-ine.
This-Pl all identical be-Past essentially one-one-Pl-Dat.
"All of these turned out to be identical to each other."

b. On-lar-da bir değiş-me ol-du mu?
This-Pl-Loc one change be-Past Quest?
"And in those, has there been a change in them?" (transcribed 1992)

(11) a. Bir kaç gün sonra Anna gel-di.
One how-many day after Anna come-Past.
"After a few days, Anna came."

b. Sırt-ın-da tilki kürk-ü var-dı.
Back-Poss3S-Loc fox fur-Poss3S there-be-Past.
"On (her) back, there was a fox fur." (Çetin Altan, *Büyük Gözaltı*, 1972:85)

Although I identify the sentence-initial element as the topic, there are sentences whose topics are not overt. Constituents that refer to salient discourse entities are often dropped in Turkish. For example, the child and the dog are the topic in all the sentences in the following story segment, although the pronouns that refer to them are not overt.

(12) a. Çocuk ve köpek uyan-dık-lar-ın-da,
Child and dog wakeup-Past-Pl-3Poss-Loc,
"When the child and the dog wake up,"

b. ∅ Frog'un yer-in-de ol-ma-dığı-nı gör-üyor-lar.
∅ Frog-Gen place-3P-Loc be-Neg-Ger-Acc see-Prog-Pl.
"(they) see that Frog is not in his place."

c. ∅ her taraf-ı ar-ıyor-lar.
∅ every side-Acc seek-Prog-Pl.
"(They) look everywhere." (Childes - 20j.cha)

Although referential form interacts with information structure, I will not be investigating the use null pronouns in Turkish since they do not take part in word order variation. See (Turan, 1995) for an investigation of null pronouns in Turkish. However, as we will see in the next chapter, I do provide an information structure in my formalism that marks the topic as "recoverable" , if it is not found in the sentence-initial position of the sentence. This means that after parsing the sentence, further discourse processing is needed to determine the identity of the sentence topic. I will not investigate the nature of this anaphor-resolution task.

There have been claims that only definite or familiar discourse entities can occur in the sentence-initial position in Turkish. In (Erkü, 1983)'s information structure for Turkish, the topic of a sentence is defined as an entity that is uniquely identifiable (i.e. definite) or belongs to a uniquely identifiable set if in the sentence-initial position or activated (i.e. discourse-old information) if in the post-verbal position. The possibility of topics in the post-verbal positions will be discussed later, page 137.

In the next three sections, I investigate the interaction between topichood and definiteness, familiarity (i.e. given/new), and salience using a measure of salience based on referential form and repeated mention. In most languages, there is a tendency for topics, subjects, the sentence initial position, definite information, given information, and salient information to be associated with each other (Chafe, 1976). However, this tendency does not mean that all subjects are topics, or all topics are definite, etc. In "free" word order languages, it is clear that subjects are not always the sentence topic since one reason for word order variation is to bring objects or adjuncts to the sentence-initial position to be interpreted as the sentence topic, an "address" where the hearer can record the rest of the information in the sentence.[1] This is why we associate the sentence-initial position with the function of topic, (Erguvanlı, 1984; Erkü, 1983; Vallduví, 1990), although there are cases where no realized sentence-initial topic can be found. Although sentence-topic

[1] However, even in Turkish, the SOV word order (48%, (Slobin and Bever, 1982)) is much more common than the OSV word order (8%, (Slobin and Bever, 1982)) because subjects are more likely sentence topics.

often refers to salient, discourse-old entities, and thus are definite in form, I will argue against claims that topics by definition have to be realized as definites or refer to discourse-old or salient information. These tendencies arise because speakers endeavor to form coherent discourses and one way of preserving coherence is to choose sentence topics that are linked to other known entities in the prior discourse.

5.3.1 Definiteness and Specificity

Many Turkish linguists have associated the sentence-initial position in Turkish with definiteness and have stipulated restrictions that bar indefinite elements from occurring in this position (Erguvanlı, 1984; Erkü, 1983; Dede, 1986; Tura, 1986; Erguvanlı, 1987). (Erkü, 1983) defines the topic as a uniquely identifiable (i.e. definite) or something belonging to a uniquely identifiable set. (Erguvanlı, 1987) points out that indefinites can occur in the sentence-initial position in certain constructions but not others, depending on their animacy and specificity. For example, she claims that indefinite and inanimate subjects of intransitive verbs or non-verbal predicates cannot occur in the sentence-initial position, as seen in (13)a compared to an animate subject in (13)b.

(13) a. #Bir kitap masa-nın üst-ün-de dur-uyor.
#One kitap table-Gen top-Poss3S-Loc stay-Prog.
"A book is lying/standing on the table."

b. Bir adam kapı-nın ön-ün-de dur-uyor.
One man door-Gen front-Poss3S-Loc stay-Prog.
"A man is standing in front of the door."

This is true in general, however it is possible to put inanimate indefinites in the sentence-initial position if they refer to specific discourse entities. For example, the indefinite, inanimate subject in (14)a is felicitous because the adjective makes the indefinite more specific, and (14)b is felicitous in the context where the speaker is waiting for a plane in an airport and has just heard an announcement about a specific plane that has landed.

(14) a. Mavı kap-lı bir kitap masa-nın üst-ün-de dur-uyor.
Blue cover-with one book table-Gen top-Poss3S-Loc stay-Prog.
"A blue covered book is lying on the table."

b. Bir uçak hava-alan-ı-na in-miş, ama biz-im ki değil.
One plane air-field-3Poss-Dat land-Past, but us-Gen1S not.
"A plane has landed in the airport, but it's not ours."

Erguvanlı also points out that in transitive sentences, the OSV word order can be preferable to the canonical SOV word order if the subject is indefinite, (15)c and d. Regardless of whether they are the subject or the direct object in the sentence, indefinite NPs are generally attracted to the

immediately preverbal position while definite NPs are attracted to the sentence initial position. This is not surprising if we look at the information structure of the sentence which associates the sentence-initial position with the topic and the immediately preverbal position as the focus.

(15) a. Fatma'yı ev-de bir sürpriz bekliyor. (OSV)
Fatma-Acc house-Loc one surprise wait-Prog.
"A surprise awaits Fatma at home."

b. #Bir sürpriz Fatma'yı evde bekliyor. (SOV)
#One surprise Fatma-Acc house-Loc wait-Prog.

I have collected a corpus of naturally occurring data with examples of transitive sentences with the SOV and the OSV word orders. Table 5.1 below describes the referential form of subjects and objects in the sentence initial (S-init) vs. the immediately preverbal (IPV) sentence positions. Unfortunately, the number of data points is too small to see a significant difference with respect to the word order tendencies of definite versus indefinite NPs. All but one sentence-initial object in OSV sentences and one sentence-initial subject in SOV sentences were definite NPs, but the elements in the immediately preverbal positions also tended to be definite NPs in the corpus. We do see a difference with respect to pronouns. The sentence-initial objects were more likely to be realized as pronouns than immediately preverbal objects, (the difference between the last two columns is marginally significant $\chi^2 = 6.349, \rho < 0.02$). As will be discussed in the next sections, page 125, pronouns tend to occur in the sentence-initial position because they refer to salient discourse entities that make good sentence topics.

	S-init Subject		IPV Subject		S-init Object		IPV Object	
Pronoun:	14	(44%)	17	(53%)	19	(59%)	9	(28%)
Def NP:	17	(53%)	14	(44%)	12	(38%)	20	(63%)
Indef NP:	1	(3%)	1	(3%)	1	(3%)	3	(9%)
TOTAL	32		32		32		32	

Table 5.1: The Referential Form of NPs in SOV and OSV Sentences.

In my corpus of OSV sentences, all but one direct object in the sentence-initial position were definite NPs. The one example of an indefinite direct object in an OSV sentence is shown below. The sentence-initial object "a thing that belongs to me" is indefinite in form, however notice that it alludes to discourse entities that are very familiar in the discourse: the speaker and the bucket (the bucket belongs to the speaker). Thus, although it is indefinite in form, it still refers to a familiar and specific discourse-entity, and it is *anchored*, (Prince, 1981b), by the discourse-old NP "me" within it.

(16) a. "Kova-yı ver-di-m, kova-yı da sonra ala-ma-dı-m, Inci", dedi.
"Bucket-Acc give-Past-1S bucket-Acc too after take-Neg-Pst-1S, Inci", say-Pst.
"(I) gave (them) [your] bucket, but then I couldn't get the bucket back, Inci", she said.

b. Ben de dön-dü21-m de-di-m ki,
I too turn-Pst-1S say-Pst-1S Comp,
"And I turned around and said,"

c. "[Ban-a ait bir şey-i] siz nasıl ver-iyor-sun-uz?" (OSV)
"[I-Dat belong one thing-Acc] you how give-Prog-2P-Pl?"
"How could you give (them) [a thing that belongs to me]?" (transcribed 1990)

Direct objects in Turkish express varying degrees of definiteness, specificity, and referentiality through the use of the indefinite marker and the optionality of the accusative case marker. The presence of "bir" (one) marks indefiniteness in Turkish, ex. (17)c,d,e,f. However, there is no definite article in Turkish. An NP without the indefinite marker can be either a definite NP (17)a or nonreferential (17)f; these two meanings can be distinguished through accusative case-marking, word order, and stress. Accusative case-marking in Turkish marks specificity (Enç, 1991), ex. (17)c,d. Although the canonical SOV word order is possible for all of the referential forms below, the OSV word order is only felicitous for those object that are more definite or specific, as can be seen below. As discussed in Chapter 3, page 50, NPs without case-marking (17)e and f generally cannot scramble, and they are given a nonreferential or nonspecific reading.

(17) a. Gazete-yi Ali oku-yor. (OSV)
Newspaper-Acc Ali read-Prog.
"As for **the** newspaper, Ali is reading it."

b. Gazete-ler-in biri-ni Ali oku-yor.
Newspaper-Pl-Gen one-Acc Ali read-Prog.
"As for **one of the** newspapers, Ali is reading it."

c. ?Hiç sev-me-diğ-im bir gazete-yi Ali oku-yor.
?None like-Neg-Rel-1S one newspaper-Acc Ali read-Prog.
"Ali is reading **a** newspaper **that** I don't like."

d. #Bir gazete-yi Ali oku-yor.
#One newspaper-Acc Ali read-Prog.
"Ali is reading **a certain** newspaper."

e. *Bir gazete Ali oku-yor.
*One newspaper Ali read-Prog.
"Ali is reading **a nonspecific** newspaper."

f. *Gazete Ali oku-yor.
*One newspaper Ali read-Prog.
"Ali is reading some newspaper(s)." (nonreferential)

Note that nonspecific objects are allowed in the sentence-initial position in some contexts, for example the contrastive gapping context. Erguvanli-84 also points out these exceptions and calls the indefinite NPs in this construction "strong topics". The indefinite object NP in 18b does not occur with accusative case. The lack of case-marking usually means that it is nonspecific (and cannot scramble), however in this context it acts more like a specific NP.

(18) a. Bir gazete-yi Ali, bir gazete-yi de kardeş-i oku-yor.
One newspaper-acc Ali, one newspaper-acc too sibling-Poss3S read-Prog.
"Ali is reading a (certain) newspaper and his sibling another."

b. Bir gazete Ali, bir gazete de kardeş-i oku-yor.
One newspaper Ali, one newspaper too sibling-Poss3S read-Prog.
"Ali is reading a newspaper and his sibling another."

Bare subject nouns can also be either definite or nonreferential. Their referentiality is determined by word order and stress. If the subject is in the sentence-initial position, it is interpreted as a definite, referring to a specific discourse entity. However, if it is in the immediately preverbal position, the preferred reading is nonspecific or nonreferential.

(19) a. Arı Ahmet'i sok-tu. (SOV)
Bee Ahmet-Acc sting-Past.
"The bee stung Ahmet."

b. Ahmet'i arı sok-tu. (OSV)
Ahmet-Acc bee sting-Past.
"Some bee stung Ahmet."

It is true that indefinite and especially nonspecific NPs tend to occur in the immediately preverbal position rather than the sentence-initial or other positions in Turkish sentences. However, I argue that this is not a grammatical restriction, but a side effect of the information structure of Turkish sentences. I claim that stipulations based on definiteness, specificity, and animacy are not necessary to capture word order variation in Turkish. In most languages, there is a tendency to place definite, specific, and animate discourse entities at the beginning of the sentence. This tendency is a result of the information structure of sentences. In Turkish, speakers place the sentence topic in the sentence-initial position, and good sentence topics are most often definite, specific, and animate entities.

Referential form indicates the accessibility of discourse referents are in the discourse model. Definite NPs usually refer to discourse entities that are already evoked in the discourse model or

inferrable from other evoked entities, (Heim, 1982), uniquely identifiable in the terms of (Erkü, 1983; Gundel, Hedberg, and Zacharski, 1990), while indefinite NPs tend to be new information. Since speakers often focus new information, and focused information tends to be placed in the immediately preverbal position in Turkish, indefinites tend to occur in this position. Speakers can place an indefinite element in the sentence-initial position of a sentence, but in order to keep the discourse coherent, the indefinite's discourse referent must be easily accommodated by the hearer. It is not the referential form of the NP that matters, but that its discourse referent is known or can be easily inferred by the hearer. This is why the indefinites we do find in sentence-initial positions in naturally-occurring discourses are linked in some way to previously evoked entities in the discourse model. In the next section, I will investigate the familiarity status of discourse entities and its interaction with word order.

5.3.2 Given/New Information

(Kuno, 1980) suggests that "free word order" languages such as Japanese, Russian, and Turkish, observe the From-Old-To-New Word Order Principle in their word order arrangement. Thus, Turkish speakers may choose a word order placing already known information first and then the new information in the immediately preverbal position or in the form of the predicate. The data presented in the last section on the interaction of definiteness and word order can be alternatively explained as an interaction between familiarity and word order.

English speakers also tend to place given information in the subject position and new information at the end of the sentence. Kuno points out that English passive sentences and the English dative-shift construction also follow the From-Old-To-New Word Order Principle in that the indefinite elements which refer to new information are more felicitous towards the end of the sentence:

(20) a. John was hit on the head by a boy.
b. ?A boy was hit on the head by John.
c. John gave the book to a boy.
d. ?John gave a boy the book.

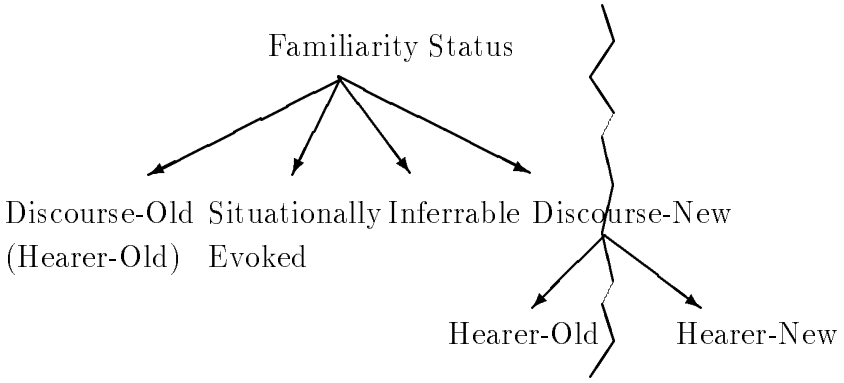
The terms 'given' and 'old' information have been used to mean different things in the linguistic literature. (Prince, 1981b) distinguishes between the varying uses of the term 'given' with respect to predictability/recoverability, saliency, and shared knowledge, as seen below. I will discuss given$_s$ in the next section on saliency, while given$_p$ as used by Kuno in discussing deletable pronouns will be discussed in next sections, page 142.

(21) a. **Given$_p$:** The speaker assumes the hearer can "predict" (i.e. recover from the context) that a particular linguistic item will or would occur in a particular position within a sentence, (Kuno, 1973).
b. **Given$_s$:** The speaker assumes the hearer has some particular entity in his/her consciousness, (Chafe, 1976).
c. **Given$_k$:** The speaker assumes the hearer "knows", assumes, or can infer a particular entity (but is not necessarily thinking about the entity), (Kuno, 1980).

(Prince, 1981b; Prince, 1992) provides a further taxonomy of given$_k$ vs. new information. These distinctions are termed the *familiarity status* of a discourse entity. Discourse-old refers to entities previously mentioned in the discourse, and hearer-old refers to those entities known to the hearer but not necessarily mentioned in the prior discourse. In between the familiarity statuses of discourse-old and discourse-new are the *inferrables* which refer to entities that the hearer can easily accommodate based on entities already in the discourse model. There are also entities that

are evoked by the situation rather than the discourse, for example, those entities that are in the line of sight of the speaker and hearer. The squiggly line below divides the given$_k$ information from brand-new information.

(22)



(Prince, 1981b) shows that subjects in English are more likely to be discourse-old information than nonsubjects. (Birner, 1994) analyzes a large corpus of naturally-occurring English inversion constructions and concludes that the sentence-initial position rather than the subject in inversions is associated with discourse familiarity.

In my corpus of naturally occurring discourses in Turkish, most sentence-initial objects are discourse-old entities, and in fact, most of them (72%) are mentioned in the previous sentence. The rest are situationally evoked, inferrable, or hearer-old discourse entities. As can be seen in Table 5.2, the sentence-initial object in the SOV and the OSV sentences always referred to given$_k$ information and never a brand-new discourse entity, i.e. the last row. The immediately preverbal elements, whether they were subjects or objects, have a higher number of brand-new discourse referents in the last row.

Given/New Status	in SOV Sentences		in OSV Sentences	
	Subject	Object	Object	Subject
Discourse-Old:	29 (91%)	24 (75%)	26 (81%)	19 (59%)
Situationally Evoked	0	0	2 (3%)	3 (9%)
Inferrable	3 (9%)	2 (6%)	4 (13%)	5 (16%)
Discourse-new, Hearer-Old	0	0	1 (3%)	1 (3%)
Discourse-New, Hearer-New	0	6 (19%)	0	4 (13%)
TOTAL	32	32	32	32

Table 5.2: The Given/New Status in SOV and OSV Sentences

In fact, the chi-square test on the 32 OSV sentences, using the statistics for the SOV sentences as the expected values for given/new distribution seen for subjects and objects, demonstrates that the distribution is statistically significant ($\chi^2 = 7.88, \rho < .005$). The expected and the observed

behaviour of subjects and objects are shown in Figure 5.3.

Expected Values (SOV)		
	Subject	Object
Given_k	32	26
Brand-New	0	6

Observed Values (OSV)		
	Subject	Object
Given_k	28	32
Brand-New	4	0

Table 5.3: The Expected and Observed Frequencies for Given/New.

Thus, the data supports the claim that brand-new discourse-entities occur in the immediately preverbal position, and not in the sentence-initial position, regardless of whether they are subjects or objects.

The old-to-new principle does generally describe the natural data, but it can be violated. For example, the sentence (23)a below, which is the first sentence of a story with no prior context, contains a sentence-initial constituent that is new information. However, even when there is no prior context, speakers try to find ways to keep the new information out of the sentence-initial position, as can be seen in (23)b.

(23) a. Bir gün, küçük bir çocuk, bir kurbağa bulmuş.
One day, little, one kid, one frog find-Past.
"One day, a little kid found a frog." (20i.cha)

b. Bu kitapta bir çocuğun, bir köpeğin ve bir kurbağanın hikayesi anlatılıyor.
This book-loc one kid-Gen one dog-Gen and one frog-Gen story-Poss tell-Pass.
"In this book, a child's, a dog's, and a frog's story is being told." (20b.cha)

I believe that the order in which speakers place given vs. new items in a sentence reflects the information structures that are available to the speakers. New information tends to be the focus of assertion for the sentence, and thus, it is placed in the immediately preverbal position which is associated with focus in the Turkish IS. The sentence-initial topic does not have to be discourse-old information. However, in order to form a coherent discourse, speakers try to link each sentence to the prior context. One way of doing this is to place information already known or easily inferred from the known information towards the beginning of the sentence to link the sentence to the prior context as soon as possible.

5.3.3 Salience and Anaphoric Linking

Although we have seen that sentence-initial elements in Turkish tend to be given$_k$ information, that is information that is known or inferrable by the hearer, there is another component of givenness that we must consider. In this section, I investigate whether sentence topics tend to refer to salient discourse entities, i.e. discourse entities that the speaker assumes the hearer already has in his/her consciousness (Chafe, 1976). By definition, salient discourse entities are given$_k$ information. One way to measure whether the speaker thinks something is salient knowledge is by looking at the referential form that the speaker uses. Salient discourse entities are often realized as pronouns. In this section, I use a measure of saliency based on referential form and repeated mention provided by Centering Theory (Grosz, Joshi, and Weinstein, 1983). (Turan, 1995) provides a comprehensive study of null and overt subject in Turkish using Centering Theory. To determine the connection between word order and centering, I collected and analyzed a number of naturally-occurring oral and written discourses in Turkish containing noncanonical word orders in (Hoffman, to appear 1995). Some of this work is also presented here.

Centering Theory is a computational model of local discourse coherence which relates each utterance to the previous and the following utterances by keeping track of the center of attention in the discourse. In the Centering Algorithm, (Grosz, Joshi, and Weinstein, 1983; Kameyama, 1985; Brennan, Friedman, and Pollard, 1987; Walker, Iida, and Cote, 1994), each utterance in a discourse is associated with a ranked list of discourse entities called the forward-looking centers (Cf list) that contains every discourse entity that is realized in that utterance.[2] The backward looking center (Cb) is a special member of the Cf list that links the current utterance to the previous utterance. Using Heim's file metaphor (Heim, 1982), we can think the Cf list as the set of file-cards that the hearer looks up upon hearing an utterance. The Cb of an utterance refers to the file-card that is the center of attention in the hearer's consciousness. As we will see, the Cb has much in common with a sentence-topic.

Centering Theory has a set of constraints, rules, and transition states defined in (Grosz, Joshi, and Weinstein, 1983; Brennan, Friedman, and Pollard, 1987) to model coherency and pronoun resolution within a discourse. I use the following adapted rules to determine the Cb of an utterance in Turkish. These rules capture the observations that salient discourse entities are often mentioned repeatedly within a discourse segment and that they are often realized as pronouns.

(24) a. The Cb of the current utterance is some entity that is realized both in the current utterance and in the prior utterance.

[2]It is usually ranked according to a hierarchy of grammatical relations, e.g. subjects are assumed to be more salient than objects. The highest ranked element in the utterance, usually the subject, is the preferred center (Cp) for the following discourse, i.e. the element that the speaker will probably continue speaking about.

b. If there is an entity realized as a zero pronoun in the current utterance (that also occurs in the prior utterance), then either this entity is the Cb or the Cb is also realized as a zero pronoun in the current utterance.

c. If there are no zero pronouns in the current utterance, but there is an entity realized as an overt pronoun in the current utterance (that also occurs in the prior utterance), then either this entity is the Cb or the Cb is also realized as an overt pronoun in the current utterance.

In (Grosz, Joshi, and Weinstein, 1983), the Cb of an utterance is actually defined as the highest ranked element of previous utterance's Cf list (i.e. the most salient entity on the list) that is realized in the current utterance instead of my rule a above. I have chosen to only use repeated mention and referential form to identify the Cb, because I do not want to take a stand on how the Cf list is ranked in Turkish.[3] The latter two rules capture the intuition zero pronouns refer to more salient entities than overt pronouns and that overt pronouns refer to more salient entities than full NPs in pro-drop languages such as Turkish.

In canonical SOV sentences in Turkish, the subject and topic in the sentence-initial position is typically the Cb of the sentence as well. The intuitive reason for this may be because speakers want to form a coherent discourse by immediately linking each sentence to the previous ones by placing the Cb in the sentence-initial position. Then, we would expect that the sentence-initial position in Turkish often corresponds with the Cb regardless of whether the element in this position is the subject of the sentence. In fact, in Turkish sentences with the noncanonical order of OSV, the object NP is typically the Cb.

For example, the following discourse is taken from a transcribed conversation between an experimenter and a child talking about stuffed toys (particularly a cat) in the CHILDES corpus. In (25)c, the sentence-initial object 'ona' (referring to the cat) is the Cb since it is mentioned in both (25)b and (25)c and is realized as a pronoun in (25)c. Note that the SOV word order is not felicitous as shown in (25)c' because the discourse-new entity, the mother, should be focused instead of occurring as the topic of the sentence.

(25) a. CHI: Funda'nın top-u-nu ver, $\emptyset_s$ oyna-sın-lar.
Funda-Gen ball-Poss3-Acc give, $\emptyset_s$ play-3P-Pl.
"Give (them)[the cat and the dog] Funda's ball, and (they) will play."

[3](Turan, to appear 1995) argues that the Cf ranking in Turkish is associated with a semantic role hierarchy (which often corresponds with the hierarchy of grammatical relations) rather than word order, and I present some further data on this in (Hoffman, to appear 1995).

b. EXP: yok, $\emptyset_s$ Funda'nın top-u-nu vere-me-m kedi-ye.
no, $\emptyset_s$ Funda-Gen ball-Poss3-Acc give-Neg-1Sg cat-Dat.
"No, (I) cannot give Funda's ball to the cat."

c. EXP: O-na da anne-si al-sın bir tane top. (OSVO)
she-Dat too mother-3Poss buy one piece ball.
"As for her [the cat], her mother should buy her a ball."

c' #anne-si ona da al-sın bir tane top. (#SOVO)
mother-3Poss she-Dat too buy one piece ball.
#"As for her mother, she should buy her [the cat] a ball."

d. CHI: $\emptyset_s$ $\emptyset_o$ al-sın, ver-sin çocuğ-un-a.
$\emptyset_s$ $\emptyset_o$ buy-3Sg, give-3Sg child-Poss3S-Dat.
"(She) should buy (one) and give (it) to (her) child." (1hb.cha - Alev)

To determine the connection between word order and centering, I collected and analyzed a number of naturally-occurring oral and written discourses in Turkish containing noncanonical word orders. I analyze noncanonical sentences with only full NPs and overt pronouns so that I can determine their word order. However, we must keep in mind that centering in pro-drop languages usually predicts which entity is more likely to be dropped in the next utterance rather than realized in a specific position in the sentence.

Tables 5.4 compare the centering analyses of utterances with the canonical word order SOV with the noncanonical OSV word order. In the SOV sentences, the subject is often the Cb, but in the OSV sentences, the object, not the subject, is often the Cb. In some cases, the centering analysis is inconclusive because the subject and the object in the sentence are realized with the same referential form (e.g. both are realized as overt pronouns or as full NPs). I did not use the ranking of the Cf list or the transition relations in finding the Cb of these sentences since these rankings are not yet conclusively determined for Turkish. In my analysis, the Cb can only be determined if there is only one overt pronoun in the current utterance that is also realized in the prior utterance, or if there is only one NP that is realized in both utterances.

Judging from this data, we can see that the linear order of subject and object in the sentence is significantly related to the Cb. The association between sentence-position and Cb is statistically significant, if we compare the 20 discourses in the first two rows of the tables above using the chi-square test ($\chi^2 = 10.10, \rho < 0.001$). If we use the values in the table for SOV sentences, the canonical word order, as the expected frequencies, then the observed frequencies in the table for OSV sentences, a noncanonical word order order, also significantly diverge from the expected frequencies ($\chi^2 = 8.8, \rho < 0.005$). Thus, speakers tend to place the Cb, the most salient entity, in the sentence initial position rather than the immediately-preverbal position in Turkish sentences

The Cb in SOV sentences.		
Cb = Subject	14	(47%)
Cb = Object	6	(20%)
Cb = Subj or Obj ?	6	(20%)
Cb = Subj or Other Obj?	0	(0%)
No Cb	4	(13%)
TOTAL	30	

The Cb in OSV sentences.		
Cb = Subject	4	(13%)
Cb = Object	16	(53%)
Cb = Subj or Obj ?	6	(20%)
Cb = Subj or Other Obj?	2	(7%)
No Cb	2	(7%)
TOTAL	30	

Table 5.4: The Cb in SOV and OSV Sentences.

regardless of whether the Cb is the subject or the object of the sentence.

Although one use of the OSV word order may be to place the Cb in the sentence initial position, we cannot claim that the sentence initial position is reserved for the Cb. The Cb can occur in any position in Turkish sentences. For example, the sentence-initial object "Bu evi" is not the Cb in the OSV sentence in (26)c, because it does not occur in the previus sentence, and there are other entities that are realized as pronouns in the current sentence.

(26) a. Nazire Abla "Kız, güzel-sin" di-yor.
Nazire sister "Girl beautiful-3Sg" say-Prog.
" Nazire sister says, "Girl, you are beautiful"."

b. $\emptyset_s$ Anne-m-in tanı-dığ-ı.
mother-Poss1-Gen know-rel.
"(She) is someone (my) mother knows."

c. Bu ev-i ban-a o bul-du,
This house-Acc I-dat s/he found
"SHE found this house for me,"

d. $\emptyset_s$ yalnız kal-dığ-ım-da.
$\emptyset_s$ alone stay-1sg-When.
"when (I) was staying by myself."

e. Bir o biliyor $\emptyset_s$ yaşadığımı.
One s/he knows $\emptyset_s$ live-1Sg-Acc.
"She is the only person who knows that I live." (Inci Aral, Ağda Zamanı:80)

The sentence-initial element in the OSV sentence is the topic of the sentence because it sets the scene for the rest of the sentence. It is given_k information, although it is not the most salient entity in the sentence, In fact, the Cb in this utterance is probably the focused pronoun in the immediately preverbal position. The OSV word order is used here in order to focus the subject

pronoun rather than to bring the most salient entity to the sentence-initial position. Thus, the Cb can occur in positions other that the sentence initial position in Turkish and can be associated with other information structure components such as *focus*.

There are also situations where no Cb can be found. In fact, speakers try to maintain a coherent discourse using a variety of strategies to link each sentence to the previous discourse; the preference to keep talking about the center of attention is only one of these strategies. For example, the de/da particle in Turkish, often translated as "too" or "also", is used to link a marked discourse entity to a salient set of entities already evoked in the discourse. In (Hoffman, 1994), I show that de/da marking is used as a type of link to the previous context to maintain a coherent discourse. It typically occurs next to the subject in the sentence initial position, and it can be used to switch to a different center of attention in the discourse. For example, even though there is no Cb in (27)b and (27)d in the discourse below, the de/da markings make the discourse coherent by linking the subject of each sentence to the salient set of characters in the story (Hoffman, 1994).

(27) a. Çocuk korku-dan asağı düş-üyor.
Child fear-Abl down fall-Pres.
"The child falls down from fear."

b. Arı-lar da bu ara-da köpeğ-i koval-ıyor-lar.
Bee-Pl too this time-loc dog-Acc chase.
"And the bees, meanwhile, chase the dog."

c. Köpek de korku-yla kaç-ıyor.
Dog too fear-with run-away.
"And the dog, he runs away with fear."

d. Çocuk da yer-de ters bir vaziyette.
Child too ground-Loc wrong a position.
"And the child, he's on the ground in an awkward position." (20b.cha)

In (Hoffman, to appear 1995), I also analyzed the utterances following sentences with SOV and OSV word orders to see if noncanonical word orders affect the Cb of the next utterance (by affecting the ranking of the Cf list in the current utterance). My data supported (Turan, to appear 1995) claim that word order alone of an utterance cannot easily predict what the Cb of the next utterance will be. In the utterance following SOV sentences, speakers often continue to talk about the subject of the SOV sentence. However, there is a high level of indeterminacy in the utterances following the OSV sentences. After an OSV sentence, speakers can either continue to talk about the same center of attention, the Cb which is also sentence-initial object in the OSV sentence, or shift to talking about some other discourse entity, and here, the subject in the OSV

sentence seems to be the first choice for the center of the next utterance. This is discussed further in (Hoffman, to appear 1995).

There is a strong association between subjects, topichood, given information, and the sentence initial position in most languages. As we have seen, this association can be extended to the Cb of an utterance as well. SOV sentences are the most common in Turkish, 48% (Slobin and Bever, 1982) because in this construction, the subject, the sentence-topic, the element in sentence-initial position, and usually the Cb are one and the same. However, in OSV sentences, which have an 8% frequency (Slobin and Bever, 1982), the object rather than the subject occurs in the sentence initial position as the topic and usually as the Cb, in order to keep the discourse coherent by linking the sentence to the prior one. However, just as elements other than the subject can occur in the sentence-initial position, the Cb does not have to occur in the sentence-initial position as the sentence-topic; the Cb can even be the focus of the sentence.

I argue that centering and information structure have different purposes in discourse processing. Centering provides a way to link each utterance to the prior one, while the information structure is a more local phenomenon concerning the information in one utterance. The sentence topic instructs the hearer to go to a certain file-card in order to update it with the information in the sentence (Vallduví, 1990); it does not tell the hearer whether that file-card is in the center of attention and makes no predictions about what will be talked about in the next sentence. When the topic is the Cb, the hearer simply remains at the same file-card in the center of attention of the discourse model. If the topic is not the Cb, the hearer looks up a different file card. The referential form of NPs indicates how accessible their file-cards are. The information structure of a sentence indicates what to do with the file-cards with respect to information-updating, while centering (i.e. the use of pronouns and repeated mention) is used to link each utterance to the prior discourse by keeping track of which file-card is at the center of attention.

5.4 The Focus

5.4.1 The Immediately Preverbal Position

In free word order languages, positional information is often used to identify the focus. (Erguvanlı, 1984; Erkü, 1983) assign the pragmatic function of focus to the immediately preverbal position in Turkish. In most Turkish sentences, this position is the intonational center and receives the primary stress. This position is also identified as the focus position in many other "free" word order languages, e.g. Hungarian, Hindi, Japanese, Urdu.

The following example demonstrates how entities in the immediately preverbal position in Turkish receive a focused interpretation. In (28)b, "father" is marked as the focus by stress, high pitch, the placement of the question morpheme "mi", as well as by its immediately preverbal position. Adjuncts such as "much" can also be focused in the immediately preverbal position as seen in (28)a.

(28) a. EXP: bu defteri de çok sevdim ben. (O-Adj-V-S)
this notebook-Acc too much like-Past-1S I.
"This notebook, I like a LOT."

b. EXP: bunu da baban mı verdi? (OSV)
this-Acc too father-2S Quest give-Past?
"This too, your FATHER gave to you?" (CHILDES 1ba.cha)

Following (Vallduví, 1990), I define the focus as the informative part of the sentence that makes a contribution of new knowledge within some context. This informational definition of focus is very successful in describing the context-appropriate answers to database queries, i.e. wh-questions and yes/no questions. Wh-questions have often been used to identify the focus of a sentence (Selkirk, 1984; Rooth, 1985; Vallduví, 1990). The wh-word in a wh-question can be seen as a variable which must be filled by new information in the answer. The new information in the answer is always focused. In English, stress and high pitch mark this focused part of the answer, whereas in Turkish and other "free" word order languages, the focus appears in a special position in the answer, i.e. the immediately preverbal position. For example, the OSV word order must be used to respond to the following wh-question in order to focus the subject; the SOV word order would be infelicitous in this context. Notice that the wh-word in the question is also focused (as indicated by sentence position and prosody) in Turkish.

(29) a. Ahmet'i kim arıyor?
Ahmet-Dat who seek-Pres.
"Who is looking for Ahmet?"

b. Ahmet'i Fatma arıyor. OSV
Ahmet-Acc Fatma seek-Pres.
"As for Ahmet, it is FATMA who is looking for him."

c. $\begin{bmatrix} \text{Topic :} & \text{Ahmet} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{Fatma} \\ \text{Ground :} & [\text{seek}] \end{bmatrix} \end{bmatrix}$

Focus is often associated with new information (Selkirk, 1984), but it is well-known that old information can be focused as well. For example, in the following sentence, "her" refers to a discourse entity that is discourse-old, but it is focused in order to contrast Joan with the group mentioned by speaker A.

(30) A: Are you coming out to dinner with us tonight?
B: No, Joan's cooking dinner. I'll stay with HER tonight.

In my Turkish corpus, I found that brand-new discourse entities are found in the immediately preverbal position, but never in other positions in the sentence. The distribution of brand-new (the last line of the table) versus given information (the rest of the table) is statistically significant, ($\chi^2 = 10.847, \rho < .001$). However, in all sentence positions, discourse-old entities are much more common than brand new discourse entities.

	S-init S<u>O</u>V,<u>O</u>SV	IPV S<u>O</u>V,O<u>S</u>V	Post-V OV<u>S</u>, SV<u>O</u>
Discourse-Old	55 (85%)	43 (67%)	56 (93%)
Situationally Evoked	1 (2%)	3 (5%)	0
Inferrable	7 (11%)	7 (11%)	4 (7%)
Discourse-new, Hearer-Old	1 (2%)	1 (2%)	0
Discourse-New, Hearer-New	0	10 (15%)	0
TOTAL	64	64	60

Table 5.5: Given/New Status and Different Sentence Positions

In fact, the focused subjects in the OSV sentences in the corpus were often realized as pronouns (53%). As seen in the last section, page 125, the subject pronouns in OSV sentences can even be the most salient entity in the sentence, the Cb, (13% in the corpus). For example, in (31)b, the pronoun "she", referring to a salient character in the discourse, is the focus of the sentence, even though it is salient and discourse-old information. The pronoun is focused in order to contrast the woman with her husband, because the narrator has just suggested in the previous context that the husband wash out the grease spot on his own shirt. If the narrator had used the SOV word

order rather than the OSV word order for this sentence, it would not have the same contrastive focus reading.

(31) a. Yağ lekesi olayı, kızın dişi kaplan gibi
Grease spot-3Poss incident, girl-Gen female tiger like
"The grease spot incident, (was resolved) when the woman, like a female tiger,"

b. ortaya atılışıyla çözülüyor.
middle-Loc jump-Nom-with solve-Pass-Prog.
"jumped into the middle."

c. Bundan böyle kocasının çamasırlarını **o** yıkayacak.
This-Abl thus husband-Gen laundry-Pl-3Poss-Acc she wash-Fut.
"From now on, when it came to her husband's laundry, SHE was going to wash it."

d. Yıkayacak, yıkayacak, ütüleyecek ve mutlu olacak.
Wash-Fut wash-Fut iron-Fut and happy be-Fut.
"(She) was going to wash, wash, iron, and be happy." (D.Asena,Değişen Birşey Yok)

The focus is often associated with a contrastive reading (Chafe, 1976). In the example above, the focus of the sentence contrasts the wife with the husband within the context of the OSV sentence. However, the focus does not always contrast just two different entities. For example, in (32)e, the pronoun "he" is focused by its sentence position and the question morpheme, and it selects one member of a set of relevant discourse entities with three members {Ismail, Rahim, Şaban} all of whom came to visit the child.

(32) a. EXP: Ismail gel-di mi?
Ismail come-Past Quest?
"Did Ismail come over (to your house)?"

b. CHI: gel-di.
come-Past.
"Yes, he came."

c. EXP: Rahim?
Rahim?
"And Rahim?"

d. CHI: geldi. Şaban Efendi.
come-Past. Mr. Şaban
"Yes, he came. and Mr. Şaban."

e. EXP: Şaban Efendi? **O da mı** geldi?
Mr. Şaban? He too Quest came?
"Mr. Şaban? HE came too?
f. CHI: geldi. (CHILDES : 1ab.cha)

According to (Prince, 1981a; Prince, 1986), the focus of a sentence is not necessarily discourse-new information, but it is new within a particular context that is the shared knowledge in the discourse. Prince calls this context the open proposition, and the focused constituent is the one element from a set of relevant discourse entities which instantiates the variable in the open proposition. The open proposition in the example (32)e is "X came to visit", and the set of relevant discourse entities that can fill this open proposition are {Ismail, Rahim, Şaban}. The focus in (32)e picks Şaban out of this set. In (31)b, the pronoun is focused in order to provide new information that is contrary to the narrator's expectations, i.e. that the wife rather than the husband is going to wash his laundry. Contrastive focus can be seen as a special case of focus where the set of alternative discourse entities contains only one other discourse entity.

In the semantic literature, (Rooth, 1985; Krifka, 1992) provide a similar definition of focus. In Rooth's alternative-set theory, a focused item is interpreted by constructing a set of alternatives from which the focused item must be distinguished. (Rooth, 1985; Krifka, 1992; Partee, 1993) show that the focus interacts with the scope of quantifiers and focus-particles such as "only" and "even" in the semantic representation. Rooth's theory provides an in-situ semantic interpretation of focus rather than allowing focus-movement at the surface structure or LF levels (Horvath, 1985; Kiss, 1987; Whitman, 1991). During the semantic interpretation of the sentence, the alternative set is constructed for the focused item, which is marked by prosodic cues, and this alternative set provides the quantificational domain for focus sensitive operators. I will only concentrate on the role of focus in information processing and not on the interaction of focus with quantification and scope in the semantic representation in this thesis.

I define the focus of a sentence as the informative part of the sentence that makes a contribution of new knowledge within some context. This is associated with the primary stress and intonational center of Turkish sentences which usually fall on the immediately preverbal position. The focus usually selects one item from a set of alternative discourse entities that are known by the speaker and hearer and that have some relevance to the current discourse. In a database-query task, this aspect of focusing can be seen in yes-no questions. In Turkish, the question morpheme "mi" can occur next to any element in the sentence, in order to question just that element. Thus, we can easily tell what the focus of the question is. We often incorporate new information into negative answers of yes-no questions. For example, for the question below, we could just answer "No, it is NOT Fatma who is looking for Ahmet"; however, if we know that someone else is looking for

Ahmet, (33)b, which provides this new, alternative information would be a more helpful response. This answer is only felicitous if we replace the focus of the question in the answer. If we replace an unfocused part of the question, the answer is not felicitous in Turkish as well as English as seen in (33)c. In English, the focus can be indicated by prosody and/or it-clefts in the yes-no questions and answers (Chomsky, 1971; Jackendoff, 1972).

(33) a. Ahmet'i Fatma mı arıyor?
Ahmet-Acc Fatma Quest seek-Pres.
"As for Ahmet, is it FATMA who is looking for him?"

b. Hayır, Ahmet'i Ayşe arıyor.
No, Ahmet-Acc Ayşe seek-Pres.
"No, (as for Ahmet) it is AYŞE who is looking for him."

c. #Hayır, Ali'yi Fatma arıyor.
#No, Ahmet-Acc Ayşe seek-Pres.
#"No, it is Fatma who is looking for Ali."

5.4.2 Focusing Verbs and VPs

In most Turkish sentences, the immediately preverbal position is prosodically prominent, and this corresponds with the informational focus. However, verbs can also be focused in Turkish by placing the primary stress of the sentence on the verb instead of immediately preverbal position. In verb initial sentences, the primary stress always falls on the verb and this corresponds with the informational focus. For example, the verb in (34)b, which is the complete sentence, is the informational focus. Thus, we must have more than one IS available for verbs, where verbs can be in the focus or the ground component of the IS.

(34) a. EXP: kuş-lar ne yap-ar?
EXP: bird-Pl what do-Aor?
EXP: "What do birds do?"

b. CHI: ∅ uç-ar-lar.
CHI: ∅ fly-Aor-Pl.
CHI: "(They) FLY." (Childes 1hb.cha)

In yes-no questions in Turkish, the question particle "mi" is placed next to whatever element is being questioned in the sentence. This can serve as a focus marker. For example, if the question particle is placed next to the verb, then the assertion or negation of the verb will be the focus in the answer, represented in the IS in (35)c:

(35) a. Ahmet'i Fatma gördü mü?
Ahmet-Acc Fatma see-Past Quest.
"As for Ahmet, did Fatma SEE him?"

b. Hayır, Ahmet'i$_T$ Fatma [GÖRmedi]$_F$.
No, Ahmet-Acc Fatma see-Neg-Past.
"No, (as for Ahmet) Fatma did NOT see him."

c. $\begin{bmatrix} \text{Topic :} & \text{Ahmet} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{neg(e)} \\ \text{Ground :} & [\text{Fatma, see(e)}] \end{bmatrix} \end{bmatrix}$

In addition, it is possible to focus the whole VP or the whole sentence. These ISs must be determined by the context, which is a question in the following case.

(36) a. Bugün Fatma ne yapacak?
Today Fatma what do-Fut?
"What's Fatma going to do today?"

b. Bugün Fatma [kitap okuyacak]$_F$.
Today Fatma book read-fut.
"Today, Fatma is going to [read a BOOK]$_F$

c. $\begin{bmatrix} \text{Topic :} & \text{today(e)} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & [\text{read(e),book}] \\ \text{Ground :} & [\text{Fatma}] \end{bmatrix} \end{bmatrix}$

In summary, the verb in Turkish sentences can belong to the focus or the ground component in the information structure. Its position is primarily distinguished by prosody. Although it is beyond the scope of this dissertation, in future research I hope to further investigate prosody in Turkish into account to fully capture the IS of Turkish sentences.

5.5 The Ground in Post-Verbal Positions

We have seen that generally the sentence-initial element is associated with the topic of the sentence, while the immediately preverbal position is associated with the focus in Turkish. The rest of the sentence forms the *ground* of the information structure. The ground component of the information structure in Turkish sentences consists of the constituents between the topic and the focus as well as postverbal constituents. In this section I will talk mostly about the discourse function of the noncanonical postverbal positions. Vallduví investigates right dislocation in Catalan which serves a similar purpose to postverbal scrambling in Turkish. He calls postverbal constituents the *tail* of the information structure and argues that its discourse function is to provide further directions to the hearer on how and exactly where the information in the sentence should be entered under a given topic.

The postverbal positions in Turkish are used to background information, (Erguvanlı, 1984). Items in these positions always have gradually falling, low pitch and are never stressed. There is no significant pause between the verb and the post-verbal items in Turkish; they are a part of the same intonational contour.[4] In addition, the post-verbal items are always given$_k$ information, i.e. discourse entities which the speaker assumes the hearer already knows or can infer. For example, the VS word order is used in (37)c to background the discourse-old subject. In English, a right dislocation construction may be used for a similar discourse function.

(37) a. Baba-m-la anne-m yüz-üyor-lar-dı.
Father-Poss1S-with mother-Poss1S swim-Prog-Pl-Past.
"(My) dad and mom were swimming."

b. $\emptyset_s$ Tayyar-a gel de-di-ler.
$\emptyset_s$ Tayyar-Dat come say-Past-Pl.
"(They) said come on to Tayyar"

c. Kork-uyor-du Tayyar. VS
Fear-Prog-Past Tayyar.
"He was afraid, Tayyar." (1hb.cha, Childes Corpus)

(Erkü, 1983) claims that post-verbal items, that refer to activated (discourse-old)) discourse entities, can sometimes be the topic of the sentence and at other times in the ground of the information structure. For example, in the VS sentence above, there is no other constituent in the sentence, other than the post-verbal subject, that could be the topic of the sentence. However, I feel that the topic in VS sentences is recovered from the context as if the topic were a zero pronoun. The word order in this sentence determines the following information structure.

[4] True afterthoughts with a significant pause and a different intonational contour are also possible in Turkish (Erkü, 1983).

The speaker assumes that the hearer already has a salient file-card at the center of his/her attention that can serve as the topic, and thus does not provide a sentence-initial topic. The post-verbal item provides backgrounded information that is not really needed but can help the hearer maintain a mutual discourse model with the speaker. It can be used to confirm that the topic is the salient entity, Tayyar, that the hearer has in mind, but it still serves a role as the ground in the information structure.

(38) $\begin{bmatrix} \text{Topic} & \text{recoverable} \\ \text{Comment} & \begin{bmatrix} \text{Focus} & \text{afraid} \\ \text{Ground} & \text{Tayyar} \end{bmatrix} \end{bmatrix}$

There is a subtle difference in meaning between the SV and the VS word orders, although they can sometimes be used in the same context. First of all, the intonation is very different for sentence-initial and post-verbal elements. In addition, post-verbal items are always backgrounded information, whereas sentence-initial items do not have a backgrounded feel to them. Sentence-initial topics can tell the hearer to look up a new or different file card, but post-verbal items cannot be used in the same contexts. For example, in the discourse below, the SOV word order is used in (39)d to switch to a different topic. The OVS word order is not felicitous in this context (39)d', much like right dislocation would not be felicitous in this context in English.

(39) a. Köpek bu arı kovan-ı-na bak-ar-ken,
Dog this bee hive-3Poss-Dat look-Aor-while,
"While the dog looked at this bee hive,"

b. çocuk da bir tavşan veya sincap yuva-sı-nın içer-sin-e bak-mış... ⋮
kid too a rabbit or squirrel home-3Poss-Gen inside-3Poss-Dat look-Past...
"the kid looked inside the home of a rabbit or a squirrel."...

c. ve kork-muş çocuk.
and fear-Past kid.
"And he was afraid, the kid."

d. Köpek de hala arı kovan-ı-na bak-ıyor-muş. SOV
Dog too still bee hive-3Poss-Dat look-Prog-Past.
"And the dog was still looking at the bee hive."

d'. #($\emptyset_s$) hala arı kovan-ı-na bak-ıyor-muş köpek. SOV but #OV or #OVS
#$\emptyset_s$ still bee hive-3Poss-Dat look-Prog-Past dog.
"He was still looking at the bee hive, the dog."

e. ve $\emptyset_s$ arı kovan-ı-nı düş-ür-müş.
and $\emptyset_s$ bee hive-3Poss-Acc fall-Caus-Past.
"And (he) made the bee hive fall down." (Childes 20a.chi)

Although I have been comparing post-verbal scrambling in Turkish to right dislocation in English, there are some important differences between the two constructions. Unlike the English construction, post-verbal scrambling in Turkish is very common, and there is no pause between the verb and the post-verbal items. In addition, overt pronouns can be postposed in Turkish and more than one item can be postposed as in (40)c,

(40) a. EXP: göz-lük mü tak-ıyor-sun artık?
EXP: eye-for quest wear-Prog-2S now?
EXP: "Are you wearing eye glasses now?"

b. CHI: böyle bak-ınca kay-ıyor.
CHI: like-this look-Ger slide-Prog.
CHI: "When I look like this, it slides off."

c. EXP: ama çok yakış-mış gözlük san-a. VSO
EXP: but much becoming-Past glasses you-Dat
EXP: "but they, the glasses, are very becoming on you." (5ga.cha)

In the next sections, I investigate the interaction between post-verbal positions and definiteness, familiarity status, and salience. I also contrast the function of post-verbal scrambling with the function of zero pronouns.

5.5.1 Definiteness and Familiarity

(Erguvanlı, 1984) claims that indefinites do not occur in post-verbal positions in Turkish. (Erkü, 1983) argues that this is too strong, since specific indefinites can occur in post-verbal positions. In my corpus of SVO and OVS sentences, I found that post-verbal subjects and objects were often definite in form, but there were a couple that were indefinite objects as seen in Table 5.6.

	Postverbal Objects		Postverbal Subjects	
Pronoun	12	(40%)	10	(33%)
Definite NP	16	(53%)	20	(67%)
Indefinite NP	2	(7%)	0	
TOTAL	30		30	

Table 5.6: The Referential Form of Postverbal Elements.

I argue that it is not the referential form but familiarity status that restricts items from post-verbal positions. Since these positions have the pragmatic function of backgrounding, they are only occupied by discourse entities that have already been evoked by the discourse or those that are easily inferrable from already evoked entities, i.e. given$_k$ information (Prince, 1981b), as seen in Table 5.7. And familiar discourse entities are often realized as definite and specific referential forms.

	Post-V Subjs	Post-V Objs
Discourse-old:	28 (93%)	28 (93%)
Inferrable	2 (7%)	2 (7%)
TOTAL	30	30

Table 5.7: Given$_k$ Status of Postverbal Arguments

Nonspecific, indefinite NPs can occur in post-verbal positions, if they refer to discourse-old or inferrable entities. For example, the indefinite NP "a ball" in (41)c refers to a discourse entity that is easily inferrable because the speaker has been talking about another ball in the previous utterance. The postposed "cat" in (41)b is a definite NP, and it refers to a discourse-old entity. However, brand-new discourse entities cannot be placed in post-verbal positions, even if they are definite NPs as seen in (41)c'.

(41) a. CHI: Funda'nın top-u-nu ver, $\emptyset_s$ oyna-sın-lar.
Funda-Gen ball-Poss3-Acc give, $\emptyset_s$ play-3P-Pl.
"Give (them) [the cat and the dog] Funda's ball, and (they) will play."

b. EXP: yok, $\emptyset_s$ Funda'nın top-u-nu vere-me-m kedi-ye. (OVO)
no, $\emptyset_s$ Funda-Gen ball-Poss3-Acc give-Neg-1Sg cat-Dat.
"No, (I) can't give Funda's ball to the cat."

c. EXP: O-na da anne-si al-sın bir tane top. (OSVO)
she-Dat too mother-3Poss buy one piece ball.
"As for her [the cat], her mother should buy her a ball." (Childes 1hb)

c'. EXP: #O-na da top al-sın anne-si. (OSVO)
#she-Dat too ball buy mother-3Poss.
#"She should buy her a ball, her mother."

5.5.2 Salience

Although postverbal elements in Turkish refer to discourse-old entities, they are not necessarily the most salient discourse entity, i.e. the backward looking center (Cb) as defined by Centering Theory (Grosz, Joshi, and Weinstein, 1983). For example, in the centering analysis of the following discourse, the null subject in (42)b continues the center of attention (Anna) from the prior sentences. The postverbal object in (42)b is not the center of attention of the sentence even though it is a discourse-given entity.

(42) a. Bir kaç gün sonra Anna gel-di. Sırt-ın-da tilki kürk-ü var-dı.
One how-many day after Anna came. Back-Poss3S-Loc fox fur-Poss3S be-Past.
"After a few days, Anna came. There was a fox fur on (her) back."

b. $\emptyset_s$ Sar-ıl-dı kucağ-ın-a al-dı ben-i. (VO)
$\emptyset_s$ wrap-Pass-Past lap-Poss3S-Loc take-Past I-Acc.
"(She) hugged me and took me on her lap."

c. Ben de iç-im kop-a kop-a sar-ıl-dı-m o-na. (SVO)
I too inside-Poss1S tear tear wrap-Pass-Past-1S her-Dat.
"And I, with my heart tearing, hugged her back." (Çetin Altan, *Büyük Gözaltı*, p85)

In (Hoffman, to appear 1995), I analyze a corpus of SVO and OVS sentences in which both the subject and the object were overtly realized. Although the distribution of the data is not statistically significant in Table 5.8, there is a slight difference that can be seen in the association between Cb and post-verbal subjects vs. post-verbal objects. The Cb in the OVS sentences tends to be the subject just as in SOV sentences. We could surmise that the post-verbal positions typically contain the Cb just like the sentence-initial position. However, in SVO sentences, both the subject and the post-verbal object are equally likely to be the Cb. Thus, subjects are likely to be the Cb in both the sentence-initial and post-verbal positions, but objects are not.

The Cb OVS Sentences		
Cb = Postverbal Subject	13	(43%)
Cb = Object	9	(30%)
Cb = Subj or Obj?	5	(17%)
No Cb	3	(10%)
TOTAL	30	

The Cb in SVO Sentences		
Cb = Subject	13	(43%)
Cb = Postverbal Object	12	(40%)
Cb = Subj or Obj?	4	(13%)
Cb = Other	1	(3%)
TOTAL	30	

Table 5.8: The Cb in OVS and SVO Sentences

In (Hoffman, to appear 1995), I also did the centering analysis for utterances following the noncanonical word orders. I found that speakers continue to speak about the subject regardless

of whether the prior utterance had the word order SVO or OVS. For example, in the following discourse, the post-verbal object refers to the Cb, the boy, but the speaker does not continue talking about the boy. Instead, the subject, the mouse, of the SVO sentence is preferred as the Cb of the next utterance.

(43) a. Ø [Ø bir tarla fare-si-nin deliğ-i ol-duğ-un-u] fark-ediyor.
Ø [Ø one field mouse-3P-Gen hole-Acc be-Ger-2S-Acc] notice-Prog.
"(He) [the boy] notices that (it) is a field mouse's hole." (Cb = the boy, CONT)

b. Fare kız-mış on-a.
Mouse angry-Past he-Dat.
"The mouse has gotten angry at him." (Cb = the boy, RETAIN)

c. $Ø_s$ rahat-sız ed-il-diğ-i için her-hal-de. (Childes 20e)
$Ø_s$ comfort-without make-Pass-Ger-Acc because all-condition-Loc.
"because (he) was made uncomfortable probably." (Cb = mouse, S-SHIFT)

5.5.3 Deleting vs. Backgrounding Elements

(Kuno, 1973; Kuno, 1980) points out that post-verbal items in Japanese occur in exactly the same contexts as deleted items (zero pronouns). This is generally true in Turkish as well. Discourse-old information can be freely dropped (44)b_1 or placed in post-verbal positions (44)b_2 in Turkish.

(44) a. Fatma Ahmet'i aradı.
Fatma Ahmet-Acc seek-Past.
"Fatma looked for Ahmet."

b_1. Ama Ø Ø bulamadı.
But Ø Ø find-Neg-Past.
"But (she) could not find (him)."

b_2. Ama bulamadı Fatma Ahmet'i.
But find-Neg-Past Fatma Ahmet-Acc.
"But she, Fatma, could not find him, Ahmet."

It seems as though deleting and scrambling to post-verbal positions serve the same pragmatic purpose. However, we can find contexts where post-verbal items cannot be dropped. In fact, the post-verbal item could not be dropped while still maintaining the same meaning for 32 of the 60 sentences in my corpus of naturally occurring Turkish discourses with OVS and SVO sentences. The reasons they could not be dropped are listed below in Table 5.9.

Most of the postverbal subjects that could not be dropped were due to ambiguity in reference (7/13, 54%). For example, in (45)b below, the salient subject "Ali" is realized in the sentence in a postverbal position and cannot be dropped; if it were dropped, we would not know whether the

	Post-V Subj	Post-V Obj	Total
Ambiguous:	7	6	13 (41%)
Not Salient Enough:	4	1	5 (16%)
Not very Inferrable:	1	0	1 (3%)
Provides extra information:	1	3	4 (12%)
Intransitive verb reading:	0	9	9 (28%)
TOTAL	13	19	32

Table 5.9: Why Post-Verbal Items Could Not be Dropped.

dropped entity referred to Ali or Karabaş.

(45) a. Ø çok üz-ül-müş-ler.
Ø very depress-Pass-ReportedPast-Pl.
"[Ali and Karabaş] became very depressed."

b. Ve hemen çizme-ler-i-ni ayağ-ı-na geçir-miş Ali/#Ø,
And at once boot-Pl-3Poss-Acc foot-3Poss-Dat pull-Past Ali
"And at once, he, Ali, pulled his boots on his feet."

c. ve kurbağa-yı ara-ma-ya git-me-ye karar ver-miş.
and frog-Acc seek-Inf-Dat go-Inf-Dat decision give-Past.
"And (he) decided to go to look for the frog." (Childes, 20i.cha)

With postverbal objects, the main problem was that the verbs had an intransitive as well as a transitive reading. For example, if the post-verbal object were dropped as in (46)b', the meaning of the sentence would change; we would infer that the verb had some generic object instead of the specific discourse referent.

(46) a. Ø [Ø bir tarla fare-si-nin deliğ-i ol-duğ-un-u] fark ed-iyor.
Ø [Ø one field mouse-3P-Gen hole-Acc be-Ger-2S-Acc] notice-be-Prog.
"(He) notices that (it) is a field mouse's hole."

b. Fare kız-mış on-a.
Mouse angry-Past he-Dat.
"The mouse has gotten angry at him."

b'. Fare kız-mış.
Mouse angry-Past.
"The mouse has gotten angry (at something)."

c. $\emptyset_s$ rahat-sız ed-il-diğ-i için her-hal-de.
$\emptyset_s$ comfort-without make-Pass-Ger-Acc because all-condition-Loc
"because (he) was made uncomfortable probably." (Childes 20e.cha)

In some of these constructions, speakers may add the post-verbal item because they realize at the last moment that the discourse referent may not be easily identifiable for the speaker, i.e. *reference repair*. Right Dislocation in English can also have a similar purpose.

However, in half of the discourses in my corpus, zero pronouns and post-verbal items were interchangeable. Thus, post-verbal scrambling is not always used to disambiguate a referent. Some of the post-verbal subjects were overt first and second person pronouns, whose meanings could be easily recovered from the agreement markings on the verb. These pronouns were probably overtly realized in the sentence for other purposes, e.g. emphasis, rather than to disambiguate the referent.

(47) a. Ben-im/$\emptyset$ de-diğ-im-e boşver-in siz/$\emptyset$. (OVS)
I-Gen say-Ger-1S-Dat ignore-2Ppolite you.
"You should ignore what I said."

b. $\emptyset$ Bırak-ın yaz-ma-yı.
$\emptyset$ leave-2Ppolite write-Inf-Acc.
"(You) should stop writing." (*Bir kış Günü,* p.18)

(Linson, 1992) points out that right dislocation in English has two purposes as well. It can be used as a reference repair or in contexts where the postverbal item could not possibly help to disambiguate the referent as in (48)b. According to Linson, this second type of right dislocation in English occurs with property-ascribing stative predicates. Turkish post-verbal scrambling is possible with any kind of predicates.

(48) a. They couldn't find him, the cops.

b. He's a bastard, that guy.

Although post-verbal scrambling in Turkish often occurs in a context where zero pronouns could have been used, post-verbal scrambling has has two discourse functions that are different from deletion. Post-verbal elements either provide information that:

- disambiguates a reference,
- or helps the hearer in the task of maintaining a mutual discourse model with the speaker.

5.6 The IS of Complex Sentences in Turkish

5.6.1 Embedded Information Structures

Most analyses of "free" word order languages do not discuss the information structure of complex sentences. However, most "free" word order languages have word order variation within embedded clauses that is pragmatically driven just like in the main clause. As in matrix clauses, arguments and adjuncts in embedded clauses in Turkish can occur in any order. In addition, the embedded clauses can occur in any order in the matrix clause, as long as they are case-marked. The word order variation in a complex sentence affects the meaning of the sentence, as indicated in the translations of the following sentences.

(49) a. Ayşe [dün Fatma'nın gittiğini] biliyor.
Ayşe [yest. Fatma-Gen go-Gerund-Acc] knows.
"Ayşe knows that yesterday, FATMA left."

b. [Dün gittiğini Fatma'nin] Ayşe biliyor.
[Yest. go-Gerund-Acc Fatma-Gen] Ayşe knows.
"As for she, Fatma, leaving YESTERDAY, it's AYŞE who knows that."

The focus of embedded clauses can be determined through question answer pairs, just like in matrix clauses.[5]

(50) a. Fatma [bu kitab-ı kim-in yaz-dığı-nı] san-ıyor?
Fatma [this book-Acc who-Gen write-Ger-Acc] think-Prog?
"Who does Fatma think wrote this book?"

b. Fatma [bu kitabı Atatürk'ün yazdığını] sanıyor.
Fatma [this book-Acc Atatürk-Gen write-Ger-Acc] think-Prog
"Fatma thinks, as for this book, that ATATÜRK wrote it."

In yes/no questions, the question particle "mi" can occur in embedded clauses as well to focus and question a particular element in the embedded clauses. Here, the embedded clause is a relative clause.

(51) a. Fatma [banka-da mı çalış-an] kadın-ı tan-ıyor?
Fatma [bank-Loc Quest work-Rel] woman-Acc know-Prog?
"Is it the woman who works at the BANK that Fatma knows/has met?"

b. Hayır, Fatma [okulda çalış-an] kadın-ı tan-ıyor.
No, Fatma [school-Loc work-Rel] woman-Acc know-Prog?
"No, Fatma knows/has met the woman who works at the SCHOOL."

[5]Unlike Hungarian, focused elements of Turkish embedded clauses can not be long distance scrambled into the immediately preverbal focus position of the matrix clause.

To capture the interpretation of the word order within embedded clauses, we must allow for recursive, embedded information structures. For example, in the sentence below the IS of the embedded clause, as expressed by its word order, is embedded in the matrix clause's IS. The embedded clause is the focus of the matrix clause since it occurs in the immediately preverbal position of the matrix clause. Note that there are other ISs available for this sentence, but the given IS assumes the most natural prosody for the sentences where the immediately preverbal position in the embedded clause receives the primary stress for the clause.

(52) a. Fatma [bu kitabı Atatürk'ün yazdığını] sanıyor.
Fatma [this book-Acc Atatürk-Gen write-Ger-Acc] think-Prog
"Fatma thinks, as for this book, that ATATÜRK wrote it."

$$\begin{bmatrix} \text{Topic :} & \text{Fatma} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \begin{bmatrix} \text{Topic :} & \text{this(book)} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{Atatürk} \\ \text{Ground :} & \text{write} \end{bmatrix} \end{bmatrix} \\ \text{Ground :} & \text{thinks} \end{bmatrix} \end{bmatrix}$$

My approach is similar to (Kiss, 1987)'s approach for handling complex sentences in Hungarian. She captures the pragmatic functions of word order in a structural representation of topic and focus that can be recursively embedded.

5.6.2 Long Distance Scrambling

In Turkish complex sentences with clausal arguments, long distance scrambling allows elements in the embedded clauses to take part in the information structure of the matrix clause. Speakers only place elements of the embedded clauses in matrix clause positions for specific pragmatic functions. Generally, an element from the embedded clause can occur in the sentence initial topic position of the matrix clause (53)b or to the right of the matrix verb as backgrounded information (53)c.[6]

(53) a. Ayşe [Fatma'nin dün gittiğini] biliyor.
Ayşe [Fatma-Gen yesterday go-Ger-Acc] knows.
"Ayşe knows that Fatma left yesterday."

b. Fatma'nın Ayşe [dün gittiğini] biliyor.
Fatma-Gen Ayşe [yest. go-Ger-Acc] knows.
"As for Fatma, Ayşe knows that she left yesterday."

[6] A syntactic restriction ensures that elements of embedded clauses cannot occur in the stressed immediately preverbal position in the matrix clause.

c. Ayşe [dün gittiğini] biliyor Fatma'nın.
Ayşe [yest. go-Ger-Acc] knows Fatma-Gen.
"Ayşe knows that yesterday she, Fatma, left."

Elements that are long distance scrambled from embedded clauses can take part in the IS of the matrix sentence, as its topic or as its ground, although they are not arguments of the matrix verb. For example, in the sentence below, "Fatma" can play the role of topic in the matrix verb's IS, although it is not an argument of the matrix verb "know", but the argument of the embedded verb "go". The topic of the embedded clause is marked as "recoverable" because it is not determined by the word order of the embedded clause. We can infer that Fatma is also the topic of the embedded clause from the context.

(54) a. Fatma'nın Ayşe [dün gittiğini] biliyor.
Fatma-Gen Ayşe [yest. go-Ger-Acc] knows.
"As for Fatma, Ayşe knows that she left yesterday."

$$\begin{bmatrix} \text{Topic} : & \text{Fatma} \\ \text{Comment} : & \begin{bmatrix} \text{Focus} : & \begin{bmatrix} \text{Topic} : & \text{recoverable} \\ \text{Comment} : & \begin{bmatrix} \text{Focus} : & \text{yesterday} \\ \text{Ground} : & \text{go} \end{bmatrix} \end{bmatrix} \\ \text{Ground} : & [\text{Ayşe, know}] \end{bmatrix} \end{bmatrix}$$

5.7 Summary

In this chapter, I have tried to determine the discourse functions of different sentence positions in Turkish by investigating naturally occurring data. Word order serves to structure the information being conveyed to the hearer in Turkish. Turkish sentences can be divided into a topic and a comment. The comment can be further divided into the focus and the ground. These IS components in Turkish sentences have the following characteristics:

- **Topic:** associated with the sentence-initial position. The sentence-topic often refers to given and salient discourse entities because speakers try to form coherent discourses by continuing to talk about a known topic or choosing a new topic that is linked to other known entities in the prior discourse. The purpose of the sentence topic is to instruct the hearer to go to a certain file-card in order to update it with the information in the sentence. In questions, the hearer uses the topic as an address of the file-card where s/he should search for the answer. The referential form of the topic informs the hearer how accessible this file-card is.

- **Focus:** associated with the immediately preverbal position and with the primary stress of the sentence. Brand-new information only occurs in the immediately preverbal position in the corpus. However, verbs can receive the primary stress and be focused instead of the element in the immediately preverbal position. Larger verbal constituents can be the informational focus as well.

- **Ground:** associated with the elements between the topic and focus and to the right of the focus (i.e. everything in the sentence except the topic and focus). Postverbal elements only refer to discourse-old or inferrable entities. Postverbal scrambling serves to background items; postverbal items can disambiguate a reference, or help the hearer in the task of maintaining a mutual discourse model with the speaker.

In addition, we have seen that in Turkish,

- The information structure of a sentence is insensitive to whether the components are arguments in the predicate-argument structure of the sentence.

- Recursive ISs are possible in complex sentences.

Chapter 6

Integrating Syntax and Information Structure in Multiset-CCG

We have seen in the last chapter that word order variation in Turkish, as in other "free" word order languages, is used to express the information structure of a sentence. In chapter 3, I presented a grammar, Multiset-CCG, that can capture the syntax of word order in simple and complex sentences of Turkish. In this chapter, I add another component to Multiset-CCG that captures the information structure of Turkish sentences. This part of the grammar provides the pragmatic word order constraints in Turkish by associating information structure components such as topic and focus with the appropriate sentence positions. As we will see, this extended version of Multiset-CCG allows elements that are not arguments in the predicate-argument structure of a clause to take part in the information structure of the clause, in order to capture the interpretation of long distance scrambling and the scrambling of adjuncts. The interface between the the syntactic and the ordering component of Multiset-CCG is simple since the flexible surface structure derived by Multiset-CCG allows syntactic constituents to easily correspond to informational/pragmatic constituents.

In section 6.1, I present the extension to Multiset-CCG to capture the context-dependent interpretation of "free" word order in Turkish. In 6.1.1, I present the representation for information structures for simple clauses in Multiset-CCG, and in 6.1.2, I describe the syntax/information-structure interface in Multiset-CCG. Then, in 6.1.3, I describe how Multiset-CCG handles complex clauses with embedded ISs and long distance scrambling. In 6.1.4, I compare the extended formalism with other formalisms which combine syntax and IS. And in 6.1.5, I discuss the formal generative capacity of the extended formalism.

In section 6.2, I describe an impletation using Multiset-CCG. I have implemented a database query task which uses Multiset-CCG in order to test and further develop the formalism within

a well-defined domain. The system, implemented in Quintus Prolog, can be used as a Personal Assistant that helps the user schedule meetings and phone calls with a number of individuals. The system interprets Turkish wh- and yes/no questions and generates answers with contextually appropriate word orders. We will see that the information structure is essential in generating context-appropriate word orders in Turkish.

6.1 Multiset-CCG

6.1.1 Information Structures in Multiset-CCG

In question-answer pairs in Turkish, the topic is found in the sentence-initial position and refers to the discourse entity that both the question and answer are primarily about. The question word and the new information that replaces the question word in the answer are found in the immediately preverbal position and interpreted as the focus of the sentences. For example, given the wh-question in (1)a, the SOV word order is the most appropriate in the response in b in order to express the correct IS, in c. However, given the question in (2)a, the OSV word order is preferred in the response in (2)b because a different IS, seen in c, must be expressed. Moreover, the word order in (1)b would not be a felicitous response to the question in (2)a, and the word order in (2)b cannot be used in response to (1)a. Although the responses in (1)b and (2)b convey the same propositional interpretation, *seek(Fatma,Ahmet)*, they have different context-dependent interpretations which we capture in the information structure representations.

(1) a. Fatma kimi arıyor?
Fatma who seek-Pres?
"Who is Fatma looking for?"

b. Fatma Ahmet'i arıyor. SOV
Fatma Ahmet-Acc seek-Pres.
"Fatma is looking for AHMET."

c. $\begin{bmatrix} \text{Topic :} & \text{Fatma} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{Ahmet} \\ \text{Ground :} & \text{seek} \end{bmatrix} \end{bmatrix}$

(2) a. Ahmet'i kim arıyor?
Ahmet-Acc who seek-Pres.
"As for Ahmet, who is looking for him?"

b. Ahmet'i Fatma arıyor. OSV
Ahmet-Acc Fatma seek-Pres.
"As for Ahmet, it is FATMA who is looking for him."

c. $\begin{bmatrix} \text{Topic :} & \text{Ahmet} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{Fatma} \\ \text{Ground :} & \text{seek} \end{bmatrix} \end{bmatrix}$

We could imagine capturing the different information structures directly in the syntactic categories in CCG. The following categories could be assigned to Turkish transitive verbs in order to capture the discourse interpretation of each possible word order of the arguments.

(3) a. S\Nnom:topic(X)\Nacc:focus(Y) (SOV)
b. S\Nacc:topic(Y)\Nnom:focus(X) (OSV)
c. S/Nacc:ground(Y)\Nnom:focus(X) (SVO)
d. S:focus(verb)/Nacc:ground(Y)\Nnom:topic(X) (SVO)
e. S/Nnom:ground(X)\Nacc:focus(Y) (OVS)
f. S:focus(verb)/Nnom:ground(X)\Nacc:topic(Y) (OVS)
g. S:focus(verb)/Nnom:ground(X)/Nacc:ground(Y) (VSO)
h. S:focus(verb)/Nacc:ground(Y)/Nnom:ground(X) (VOS)

However, such a formalism would not be able to capture the interpretation of sentences with adjuncts in different word orders or with long distance scrambling. For example, we must place an adjunct in focus to form a felicitous answer to the following query.

(4) a. Fatma ne zaman git-ti?
Fatma what time go-Past?
"When did Fatma leave?"

b. Fatma beş-te git-ti.
Fatma five-Loc go-Past.
"Fatma left at FIVE."

c. $\begin{bmatrix} \text{Topic :} & \text{Fatma} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{time(e,5)} \\ \text{Ground :} & \text{[go(e)]} \end{bmatrix} \end{bmatrix}$

In "free" word order languages, adjuncts and elements long distance scrambled from embedded clauses can take part in the information structure (IS) of the matrix clause even though they are not arguments in the predicate-argument structure (AS) of the matrix clause. We capture this in Multiset-CCG by separating the grammar into two components: categories/rules that

derive the AS and categories/rules that derive the IS of a sentence. As presented in Chapter 3, the syntactic component of Multiset-CCG captures the AS of sentences while allowing local scrambling of arguments and adjuncts and long distance scrambling. In addition, it can capture the syntactic restrictions on word order by allowing prioritized multisets in the lexical categories or by restricting the composition rules. As we will see in this chapter, Multiset-CCG can be extended so that the AS and the IS of a sentence are built in parallel in a compositional way. Each word in the sentence is assigned a syntactic/semantic category which is then associated with an ordering category. The ordering categories order the topic and focus of the sentence, but do not care whether or not they are arguments of the verb. As the syntactic/semantic categories combine to form larger constituents in the AS, the ordering categories combine to form constituents in the IS. The syntax-pragmatics interface in Multiset-CCG will be described further in the next section. In this section, I will describe the ordering component of the grammar.

The ordering category associated with verbs in Multiset-CCG serves as a template for the IS. For example, the ordering category in (5) is a function that specifies where the focus, topic, and ground components must be found to complete a possible IS. The forward and backward slashes in the category indicate the direction in which the IS components must be found, and the parentheses around IS components indicate optionality. This function is looking for a focused constituent on its left, then an optional ground constituent on its left, then a topic constituent on its left, and then an optional ground constituent on its right. The variables $Top, Foc, Grnd1, Grnd2$ will be unified with the semantic interpretations of the proper constituents in the sentence during the derivation to complete the features in the information structure.[1]

(5) $IS/\ (Grnd2)\backslash\ Top\backslash\ (Grnd1)\backslash\ Foc,$

$$\text{where } IS = \begin{bmatrix} \text{Topic :} & Top \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & Foc \\ \text{Ground :} & [\text{verb } \&\ Grnd2\ \&\ Grnd1\ ] \end{bmatrix} \end{bmatrix}$$

Nonverbal elements are associated with simpler ordering categories, often just a variable associated with a semantic interpretation which can unify with the topic, focus, or any other component in the IS template during the derivation. The function associated with verbs can use the simple forward and backward application rules below to combine with other elements in the sentence.

(6) a. **Simple Forward Application (>):** $X/Y \quad Y \Rightarrow X.$

b. **Simple Backward Application (<):** $Y \quad X\backslash Y \Rightarrow X.$

Optional IS components (i.e. the ground) can be skipped during the derivation through a category rewriting rule, that can apply after the application rules.

[1] Many of the examples in this section contain just the words in the IS instead of the full semantic representation, for the ease of readability.

(7) **Skip Optional:** $X|(Y) \Rightarrow X$

A sample derivation for the sentence "Fatma left at FIVE" involving just the ordering categories is shown below. The derivation is complete when all of the obligatory components of the information structure have been found, and all of the optional components have been skipped.

(8) Fatma beş-te git-ti.
Fatma five-Loc go-Past.

X:Fatma Y:time(e,5) $\left[\begin{array}{l}\text{Topic:} \quad T \\ \text{Comment:} \quad \left[\begin{array}{ll}\text{Focus:} & F \\ \text{Ground:} & [\text{go}(e), G2, G1]\end{array}\right]\end{array}\right] /(G2)\backslash T\backslash(G1)\backslash F$

$$\frac{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}{\left[\begin{array}{l}\text{Topic:} \quad T \\ \text{Comment:} \quad \left[\begin{array}{ll}\text{Focus:} & \text{time}(e,5) \\ \text{Ground:} & [\text{go}(e), G2]\end{array}\right]\end{array}\right] /(G2)\backslash T} <,\text{skip}$$

$$\frac{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}{\left[\begin{array}{l}\text{Topic:} \quad \text{Fatma} \\ \text{Comment:} \quad \left[\begin{array}{ll}\text{Focus:} & \text{time}(e,5) \\ \text{Ground:} & [\text{go}(e)]\end{array}\right]\end{array}\right]} <,\text{skip}$$

There is also an identity rule in the ordering component of Multiset-CCG. Although the ordering function associated with verb only subcategorizes for four components in the information structure, these components can be unbounded in length. The identity rule allows two constituents with the same discourse function (or variables) to combine. Note that their syntactic counterparts in Multiset-CCG must also be able to combine.

(9) **Identity (=):** $(X \;\; X \;\Rightarrow X)$

There are also other ordering categories for verbs that represent different information structures. In most Turkish sentences, the immediately preverbal position is prosodically prominent, and this corresponds with the informational focus. However, verbs can be focused in Turkish by placing the primary stress of the sentence on the verb instead of immediately preverbal position and by lexical cues such as the placement of the question morpheme. Thus, we must have more than one IS available for verbs, where verbs can be in the focus or the ground component of the IS. For example, in the following question-answer pair, the verb is in focus in the question, and the negated verb is in focus in the answer.

(10) a. Ahmet'i Fatma GÖRdü mü?
Ahmet-Acc Fatma see-Past Quest.
"As for Ahmet, did Fatma SEE him?"

b. Hayır, Ahmet'i Fatma GÖRmedi.
No, Ahmet-Acc Fatma see-Neg-Past.
"No, (as for Ahmet) Fatma did NOT see him."

c. $\begin{bmatrix} \text{Topic :} & \text{Ahmet} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & [\text{neg(e) \& see(e)}] \\ \text{Ground :} & [\text{Fatma}] \end{bmatrix} \end{bmatrix}$

In addition, it is possible to focus the whole VP or the whole sentence. The primary stress of the sentence falls on the immediately preverbal element in these cases, and the scope of the focus must be determined by the context, in this case the database query.

(11) a. Bugün Fatma ne yapacak?
Today Fatma what do-Fut?
"What's Fatma going to do today?"

b. Bugün Fatma [kitap okuyacak]$_F$.
Today Fatma book read-fut.
"Today, Fatma is going to [read a BOOK]$_F$

In addition, in some Turkish sentences there is no overtly realized topic in the sentence-initial position. Another IS is available where the topic component is marked as "recoverable", for those cases where the topic is a zero pronoun instead of an element which is realized in the sentence, as in (12). After the derivation is complete, further discourse processing is necessary to infer the identity of the unrealized topic from among the salient entities in the discourse model.

(12) a. Fatma kimi arıyor?
Fatma who seek-Pres?
"Who is Fatma looking for?"

b. Ø Ahmet'i arıyor. OV
Ø Ahmet-Acc seek-Pres.
"(She) is looking for AHMET."

c. $\begin{bmatrix} \text{Topic :} & \text{recoverable} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{Ahmet} \\ \text{Ground :} & [\text{seek(e)}] \end{bmatrix} \end{bmatrix}$

In verb-initial sentences, the verb always receives the primary stress of the sentence and is focused. Any post-verbal items are backgrounded information, as in 13. The topic of the sentence below is marked recoverable because there is no sentence-initial topic. The actual topic "Fatma" can be recovered from the context. In order to make the recovering process easier for the hearer, the speaker has provided the referent "Fatma" as backgrounded information in a post-verbal position, but not all post-verbal elements turn out to be the topic of the sentence.

(13) a. Git-ti mi Fatma?
Go-Past Quest Fatma?
"Did she, Fatma, LEAVE?"

b. Hayır, GIT-me-di Fatma.
No, go-Neg-Past Fatma.
"No, she, Fatma, did NOT leave."

c. $\begin{bmatrix} \text{Topic :} & \text{recoverable} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & [\text{neg(e)\& leave(e)}] \\ \text{Ground :} & [\text{Fatma}] \end{bmatrix} \end{bmatrix}$

In summary, the ordering categories that can be associated with verbs in the Multiset-CCG for Turkish are shown below. Each represents a different information structure. The first (14)a has an obligatory sentence-initial topic and an obligatory immediately preverbal focus; ground constituents between the topic and the focus or to the right of the verb are optional. In (14)b, the verb rather than the immediately preverbal constituent is in focus; this category is only chosen by verbs that are stressed or lexically marked as focused by the question morpheme. (14)c captures sentences where the topic is not realized in the sentence, but there is an obligatory preverbal focus. In (14)d, there is no preverbal topic or focus, and thus, the verb is in focus. We could also add ordering categories that allow both the verb and a preverbal constituent to be in focus, with or without an overtly realized topic. These have not been implemented in the system in order to avoid dealing with the ambiguity in the information structure of questions.

(14) a. $IS/\,(Grnd1)\backslash\, Top\backslash\,(Grnd2)\backslash\, Foc,$

where $IS =$ $\begin{bmatrix} \text{Topic :} & Top \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & Foc \\ \text{Ground :} & [\text{verb}, Grnd2, Grnd1\,] \end{bmatrix} \end{bmatrix}$

b. (only assigned if verb is stressed or occurs with a question morpheme.)
$IS/\,(Grnd1)\backslash\, Top\backslash\,(Grnd2),$

where $IS =$ $\begin{bmatrix} \text{Topic :} & Top \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{verb} \\ \text{Ground :} & [Grnd2, Grnd1\,] \end{bmatrix} \end{bmatrix}$

c. $IS/\,(Grnd)\backslash\, Foc$,

where $IS =$ $\begin{bmatrix} \text{Topic :} & \text{recoverable} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & Foc \\ \text{Ground :} & [\text{verb}, Grnd\,] \end{bmatrix} \end{bmatrix}$

d. (for verb-initial sentences)
$IS/\,(Grnd)$,

where $IS =$ $\begin{bmatrix} \text{Topic :} & \text{recoverable} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{verb} \\ \text{Ground :} & Grnd \end{bmatrix} \end{bmatrix}$

The rules in the ordering component of Multiset-CCG are summarized below:

(15) a. **Simple Forward Application (>):** $X/Y \quad Y \;\Rightarrow X$.

b. **Simple Backward Application (<):** $Y \quad X\backslash Y \;\Rightarrow X$.

c. **Skip Optional:** $X|(Y) \Rightarrow X$

d. **Identity (=):** $(X \;\; X \;\Rightarrow X)$

6.1.2 The Syntax/IS Interface in Multiset-CCG

Multiset-CCG can capture both the syntax and context-dependent interpretation of word order in Turkish by deriving the predicate-argument structure and the information structure of a sentence in parallel. I adopt the simple compositional interface described below to integrate the syntactic and ordering components of Multiset-CCG.

A. Each Multiset-CCG category encoding syntactic and semantic properties in the AS is associated with an Ordering Category which encodes the ordering of IS components.

B. Two constituents can combine if and only if

   i. their syntactic/semantic categories can combine using the Multiset-CCG application and composition rules,

   ii. and their Ordering Categories can combine using simple application and identity rules.

This interface is very similar to Steedman's approach in integrating prosody and syntax in CCGs for English (Steedman, 1991; Prevost and Steedman, 1993). Their theory of prosody, closely related to a theory of information structure, is integrated with CCGs by associating every CCG category encoding syntactic and semantic properties with a prosodic category. Taking advantage of the nontraditional constituents that CCGs can produce, two CCG constituents are allowed to combine only if their prosodic counterparts can also combine. The interface I have presented ties each syntactic constituent to a component of the information structure as determined by word order, rather than prosody. In future research, I would like to expand Multiset-CCG to use both prosodic and word order information to determine the information structure of Turkish sentences.

Multiset-CCG provides a compositional and parallel derivation of the predicate-argument structure and the information structure of a sentence. For example, given the following question, a felicitous answer uses a word order that indicates that "Fatma" is the topic of the sentence, and that "a student" is the focus. The derivation for this answer is seen in Figure 6.2.

(16) a. Fatma'yı kim ara-dı bu-gün?
Fatma-acc who seek-Past this-day?
"As for Fatma, who called her today?"

b. Fatma'yı bir öğrenci aradı bugün.
Fatma-Acc one student seek-Past this-day.
"As for Fatma, it was a STUDENT who called her today."

Every word in the sentence is associated with a lexical category which is then associated with an ordering category. In the implementation, both categories are placed into one DAG. For example, Figure 6.1 is the DAG associated with the verb "aradı" (seek). This verb is assigned the

lexical category seen in the *category* feature of the DAG which contains the argument structure in the features *syn* and *sem* and an empty information structure in the feature *info*. The ordering category is unified in as the feature *order* of the DAG, and it is linked to the *category* feature via the co-index *IS*. The Multiset-CCG application and composition rules apply to the syntactic/semantic category contained in the *category* feature, and simple application rules apply to the ordering category contained in the *order* feature. The syntactic/semantic and ordering features remain together in one DAG as we build larger constituents through the application of the rules. At the end of the parse, the category:result feature will contain the syntactic, semantic, and information structure features associated with the completed sentence.

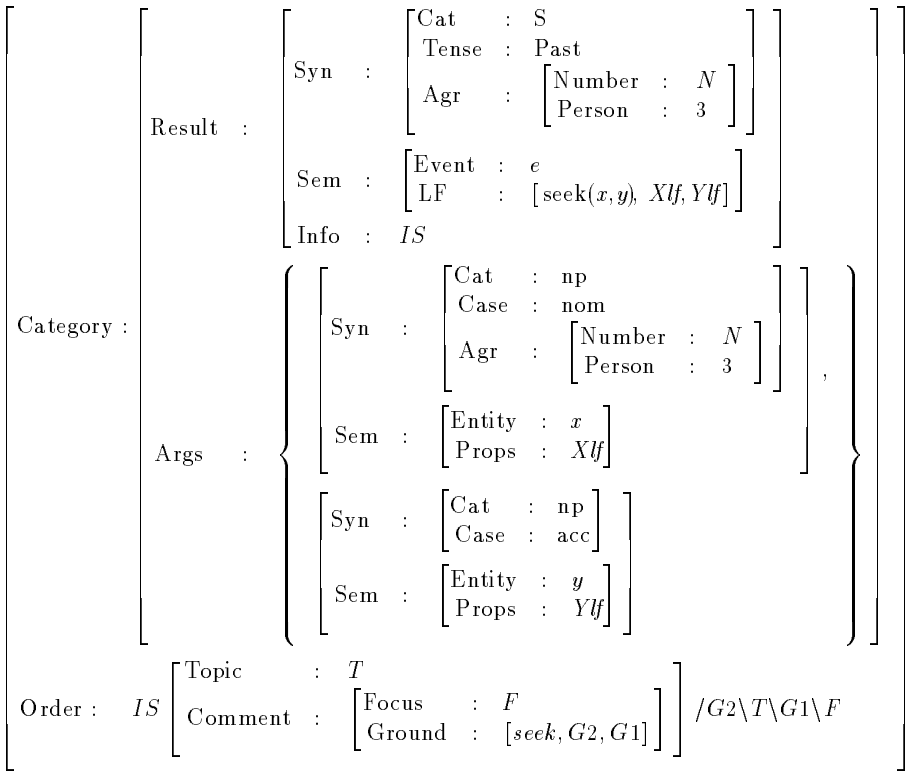


Figure 6.1: The DAG for the transitive verb "aradı" (seek).

Figure 6.2 shows the bottom-up derivation for the sentence in (16)b. First, each word is assigned a lexical category, shown on the line below it, and then this is associated with an ordering category shown on the next line. Then, we apply the rules of each component of the grammar in parallel to form larger constituents; this process is notated as parallel horizontal lines in Figure 6.2. The first line is the application of a rule for combining the syntactic/semantic categories to derive the AS, and the second line is for combining the ordering categories of the two constituents to derive the IS. The syntactic constituents are allowed to combine to form a larger constituent, only if their pragmatic counterparts (the ordering categories) can also combine.

In Multiset-CCG derivations, the surface structure directly reflects both syntactic constituency and informational/pragmatic constituency. The advantage of using a CCG formalism is that its flexible surface structure can produce syntactic constituents that correspond to information structure components. For example, in Figure 6.2, the subject "a student" and the verb "seek" can combine together to form a syntactic and informational constituent; this would not be possible in a traditional grammar that only allows VPs to combine with subjects. In addition, type-raising and composition can be used to produce non-traditional syntactic constituents that correspond to the ground components in the IS. For example, two NPs can form a constituent in the IS

using the identity rule in the ordering grammar, and a non-traditional syntactic constituent in the syntactic part of Multiset-CCG by using type-raising and composition.

The ordering component of Multiset-CCG also influences the syntactic derivation of sentences. For example, the syntactic rules could combine the verb "seek" and the post-verbal adjunct "today" together before the verb has combined with the rest of the sentence. However, this combination is not possible for the ordering categories associated with "seek" and "today". The ordering category captures the intuition that the sentence segment up to and including the verb feels like one informational (and prosodic) unit to Turkish speakers; the post-verbal items contribute backgrounded information to the unit formed by the rest of the sentence. Thus, the ordering categories prohibit this extra derivation. The syntactic grammar also influences the IS components. For example, "one" and "student" must combine together in order to satisfy the syntactic component of the grammar; this forces their ordering categories to also combine using the identity rule. Thus, syntactic and pragmatic constraints work together to determine the surface structure and word order of the sentence.

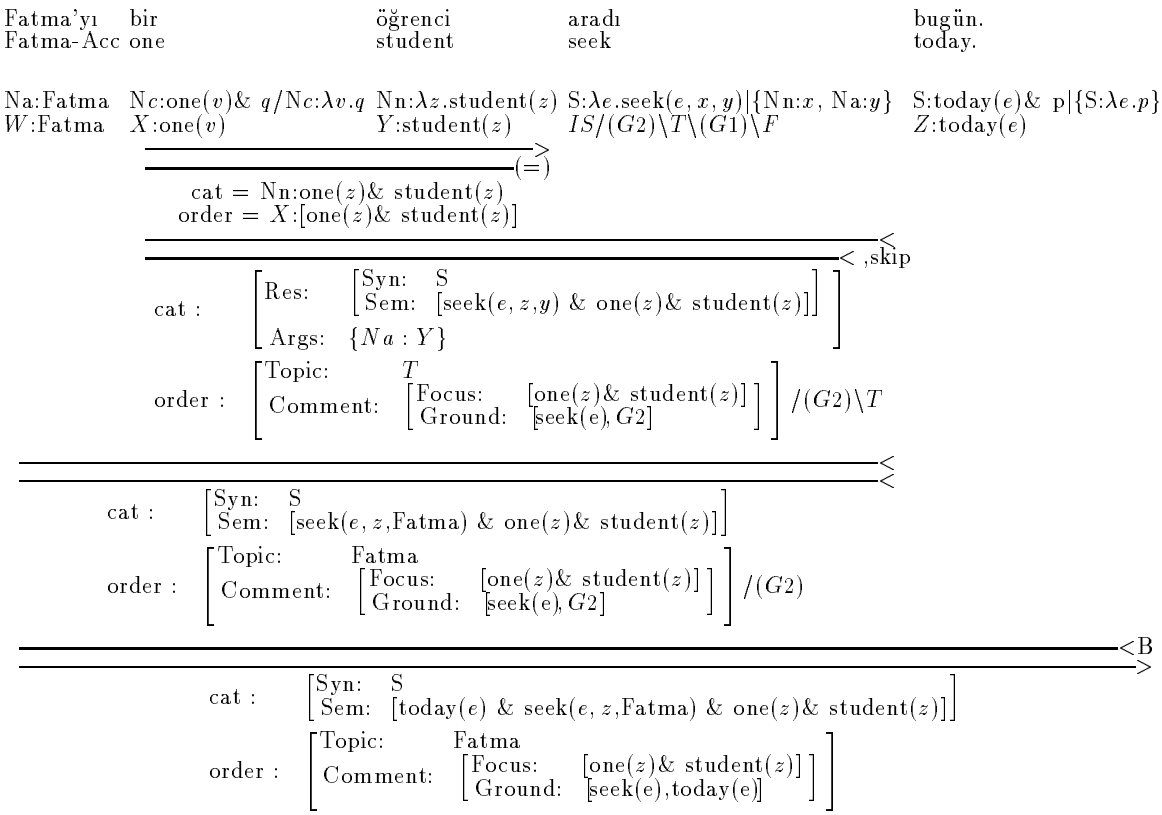


Figure 6.2: Deriving the Predicate-Argument and Information Structure for a Simple Sentence.

6.1.3 Complex Clauses

6.1.3.1 Embedded Information Structures

Arguments and adjuncts within the embedded clauses also exhibit "free" word order. The word order variation in embedded clauses corresponds with subtle differences in meaning. In addition, case-marked embedded clauses can occur in different positions in Turkish sentences and change the discourse-meaning of the sentence. Thus, we must allow embedded information structures in order to capture the interpretation of the word order in complex sentences. For example, in (17), the subject "Ayşe" is the topic of the matrix clause while the embedded clause in the sentence acts as the informational focus of the matrix clause, and the word order within the embedded clause marks "yesterday" as the embedded topic and "Fatma" as the embedded focus. Sentence (18) has the same truth-conditional meaning as (17) but a different discourse-meaning. In (18), the embedded clause acts as the topic of the matrix clause, while "Ayşe" acts as the focus, and the word order within the embedded clause marks "Fatma" as the embedded topic and "yesterday" as the embedded focus.

(17) a. Ayşe [dün Fatma'nın git-tiğ-i-ni] bil-iyor.
Ayşe [yest. Fatma-Gen go-Ger-3S-Acc] know-Pres.
"As for Ayşe, she knows that yesterday, FATMA left."

$$\begin{bmatrix} \text{Topic :} & \text{person(Ayşe)} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \begin{bmatrix} \text{Topic :} & \text{yesterday(e1)} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{person(Fatma)} \\ \text{Ground :} & \text{go(e1,Fatma)} \end{bmatrix} \end{bmatrix} \\ \text{Ground :} & \text{know(e2,Ayşe,e1)} \end{bmatrix} \end{bmatrix}$$

(18) a. [Fatma'nın dün git-tiğ-i-ni] Ayşe bil-iyor.
[Fatma-Gen yest. go-Ger-3S-Acc] Ayşe know-Pres.
"As for Fatma's leaving YESterday, it's AYŞE who knows that."

$$\begin{bmatrix} \text{Topic :} & \begin{bmatrix} \text{Topic :} & \text{person(Fatma)} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{yesterday(e1)} \\ \text{Ground :} & \text{[go(e1,Fatma)]} \end{bmatrix} \end{bmatrix} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{person(Ayşe)} \\ \text{Ground :} & \text{[know(e2,Ayşe,e1)]} \end{bmatrix} \end{bmatrix}$$

In Multiset-CCG, subordinate verbs, just like matrix verbs, are associated with ordering categories that determine the available information structures for the clause. When the subordinate

clause syntactically combines with the matrix clause, the IS of the subordinate clause is embedded into the IS of the matrix clause.

To ensure that the embedded IS is complete before it is placed into the matrix clause's IS, we must restrict the application rules in the ordering component of Multiset-CCG; we stipulate that the argument Y must not be a function (with arguments left to find). The restriction ensures that the ordering category for the embedded verb has found all of its obligatory components and skipped all the optional ones before combining with the matrix verb's ordering category.

(19) a. **Simple Forward Application' (>):** $X/Y \quad Y \Rightarrow X \quad (Y \neq Res|Args)$

b. **Simple Backward Application' (<):** $Y \quad X\backslash Y \Rightarrow X \quad (Y \neq Res|Args)$

6.1.3.2 Long Distance Scrambling

Multiset-CCG can recover the appropriate predicate-argument relations of the embedded clause and the matrix clause even when the arguments occur out of the domain of the subordinate verb. For example, in (20), "Fatma" is the subject of the embedded verb, but occurs as the topic of the matrix sentence in the sentence-initial position.

(20) a. Fatma'nın Ayşe [dün gittiğini] biliyor.
Fatma-Gen Ayşe [yest. go-Ger-Acc] knows.
"As for Fatma, Ayşe knows that she left YESTERDAY."

b.
$$\begin{bmatrix} \text{Topic :} & \text{person(Fatma)} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \begin{bmatrix} \text{Topic :} & \text{recoverable} \\ \text{Comment :} & \begin{bmatrix} \text{Focus :} & \text{yesterday(e1)} \\ \text{Ground :} & \text{go(e1,Fatma)} \end{bmatrix} \end{bmatrix} \\ \text{Ground :} & [\text{know(e2,Ayşe,e1)} \ \& \ \text{person(Ayşe)}] \end{bmatrix} \end{bmatrix}$$

The derivation for the sentence above is shown in Figure 6.3 on the next page. First, the embedded verb completes its IS (IS_2); then, the two verbs compose together, and the subordinate IS is embedded into the matrix IS (IS_1). The syntactic composition rules allow two verb categories with multisets of arguments to combine together. As the two verbs combine, their arguments collapse into one argument set in the syntactic representation. The complex verbal constituent can then combine with the rest of the arguments of both verbs in any order. The linear order of the arguments will determine which components of the matrix IS each fill. In this sentence, "Fatma" is an argument in the AS of the embedded verb "go", not the matrix verb "know", however it plays the role of topic in the matrix verb's IS. The ordering component of Multiset CCG allows individual elements from subordinate clauses to be components in the IS of the matrix clause, even though they are not a part of the matrix argument structure. This is because the ordering

category for a matrix verb does not specify that its components be arguments in its AS.

In summary, Multiset CCG captures the context-appropriate use of word order by compositionally deriving the predicate-argument structure and the information structure of a sentence in parallel. It allows adjuncts and elements from embedded clauses to take part in the information structure of the matrix clause, even though they do not take part in its predicate-argument structure. Thus, this formalism provides a uniform approach in capturing the syntactic and pragmatic aspects of word order variation among arguments and adjuncts, and across clause boundaries.

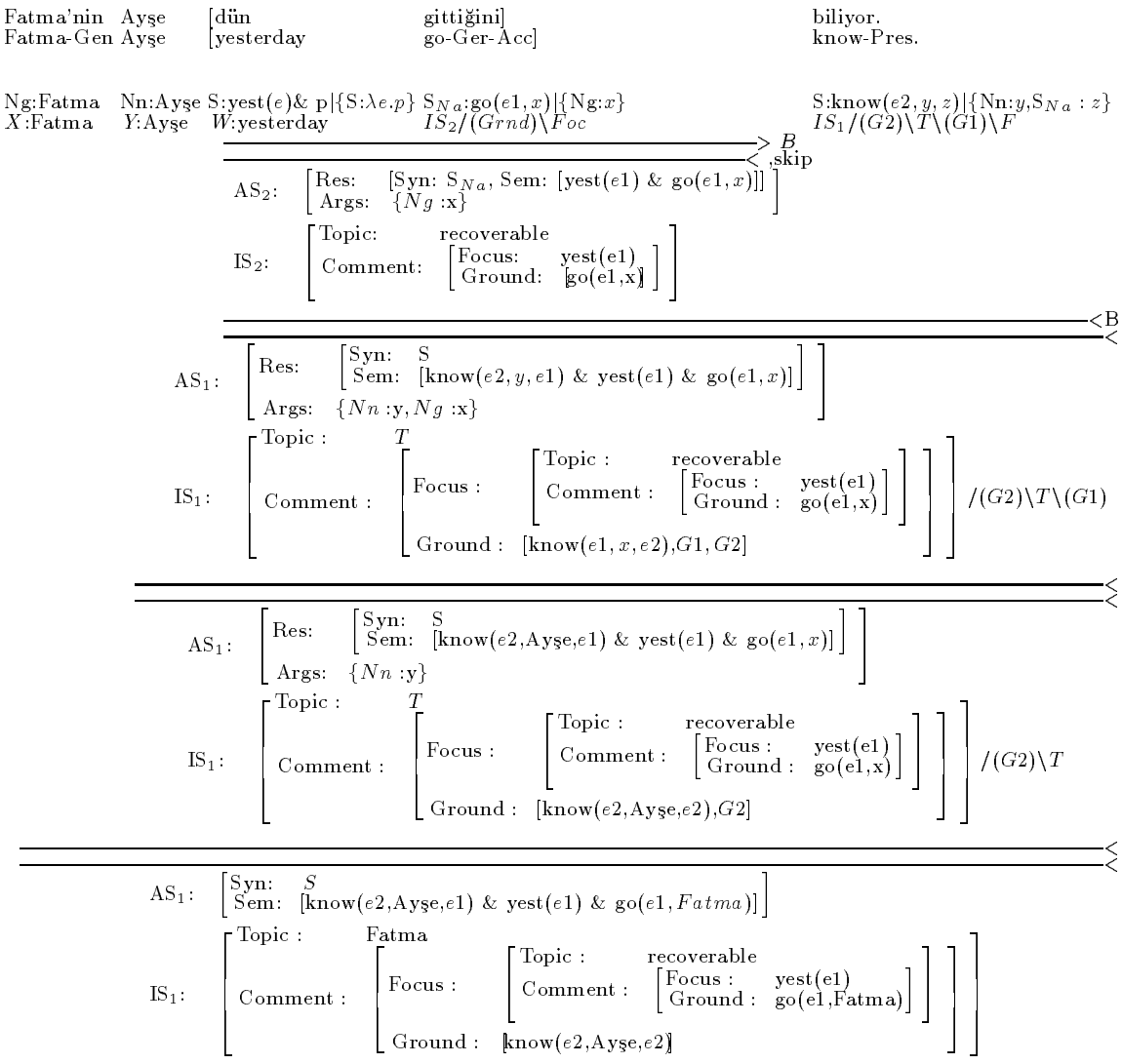


Figure 6.3: Derivation for the AS and IS of a Complex Sentence.

6.1.4 Comparison to Other Approaches

In this section I compare Multiset-CCG to two common previous approaches to integrating syntax and information structure for "free" word order languages: syntactic movement theories and ID/LP formalisms. These approaches were introduced in Chapter 2.

In the linguistic literature, there has been a tradition of associating "free" word order with syntactic movement (Horvath, 1985; Kiss, 1987; King, 1993). In this approach, arguments are generated in a structure that represents the argument structure of the sentence, and then elements in the sentence can move into specific positions in the surface structure and receive an interpretation as the topic or the focus of the sentence. Multiset CCG is similar to these linguistic approaches in that it identifies certain sentence positions with pragmatic functions. Thus, in both approaches, the surface structure of a sentence directly reflects its information structure. However, Multiset-CCG does not need movement rules and traces to account for "free" word order, because the notion of constituency in Multiset CCG is much more flexible than in the more traditional grammars. The argument structure in Multiset-CCG is determined in the lexical categories, and the information structure is determined by the ordering categories. The surface structure of a sentence in Multiset-CCG is derived directly from the lexical and ordering categories by using composition rules. In addition, unlike many of the linguistic approaches, Multiset CCG is computationally attractive because it is a lexicalist and compositional grammar that is polynomially parsable.

Another approach to "free" word order languages has been to separate the grammar into two components. ID/LP grammars that divide the grammar rules into immediate dominance (ID) and linear precedence (LP) rules are very common among linguistic and computational approaches (Uszkoreit, 1987; Gunji, 1987; Mohanan, 1982; King, 1993; Engdahl and Vallduvi, 1994; Steinberger, 1994). The syntactic component of Multiset-CCG was compared to these approaches in Chapter 4. Here, I concentrate on the way these formalisms can integrate syntax and information struccture. Multiset CCG is similar to ID/LP formalisms in that the grammar is divided into two components for descriptive purposes. However, the line of division is not exactly identical. The LP rules in ID/LP grammars can order syntactic, semantic, and pragmatic features. For example, the syntactic restriction that NPs precede verbs (NP $<$ V) in strictly verb-final languages as well as the pragmatic restriction that topics precede focused items (Topic $<$ Focus) are placed in the LP rules[2] In contrast, in Multiset CCG, syntactic restrictions on word order are handled in the syntactic part of the grammar by directionality restrictions in lexical

[2]It is not clear in the ID/LP literature whether a pragmatic LP rule such as (Topic $<$ Focus) acts on sisters in ID rules like the syntactic LP rules do. Such a LP rule would have to act on leaves in the tree, not just sisters, in order to capture complex sentences. This would probably increase the power of the formalism.

categories or restrictions on the syntactic composition rules. The ordering part of Multiset CCG contains only information about the ordering of information structure components such as topic and focus. Thus, the distinction in Multiset CCG is not between dominance and word order, but between the predicate-argument structure and the information structure. In addition, unlike many of the ID/LP formalisms, Multiset CCG provides a treatment of complex sentences with embedded information structures, unbounded long distance scrambling, and island behaviour.

6.1.5 The Generative Capacity of Multiset-CCG

In Chapter 4, I discussed the weak generative capacity of the syntactic component of Multiset-CCG and showed that it was within the class of context-sensitive grammars. I also presented a polynomial-time parsing algorithm for Multiset-CCG . In this section, I conjecture that adding the IS component to Multiset-CCG does not change its weak or strong generative capacity because the rules of the two components always apply in parallel to produce one surface structure.

The parallel parsing procedure introduced for Multiset-CCG could be seen as deriving the intersection of two languages, the languages that can be generated by each component of the grammar separately. As seen in Chapter 4, the syntactic component of Multiset-CCG, i.e. Prioritized Multiset-CCG, generates some but not all context-sensitive languages. The ordering component of Multiset-CCG is a categorial grammar that uses only application rules, and thus is a CFG (Hillel, Gaifman, and Shamir, 1960) (although regular expressions may be adequate to capture the ordering as well). Prioritized Multiset-CCG are not closed under intersection with CFLs since $\{a^m b^n c^n d^n | n, m \geq 0\} \cap \{a^n b^n c^* d^* | n \geq 0\} = \{a^n b^n c^n d^n e^n | n \geq 0\}$, (although Prioritized Multiset-CCG can generate the COUNT-3 language, it cannot generate COUNT-4).[3] Although the weak generative capacity of a formalism that intersects Prioritized Multiset-CCLs with CFLs would slightly increases, the formalism does remain context-sensitive, because CSLs are closed under intersection with CFLs (Hopcroft and Ullman, 1979). Thus, the resulting formalism can only generate languages that are context-sensitive, and moreover it still cannot generate CSLs such as COUNT-K or WWW.

However, the extended Multiset-CCG appears to be further restricted in its generative capacity, because the two components of the grammar have isomorphic derivation structures. The derivation structures for strings in each of the two languages that the extended Multiset-CCG intersects must be isomorphic. This prevents the intersection of languages like COUNT-3 and $\{a^n b^n c^* d^* | n \geq 0\}$ that increase the weak generative capacity as above. (Rambow and Satta, personal communication) conjecture that synchronization with isomorphic derivation structures does not increase the weak (or strong) generative capacity of the synchronized formalisms. This has been proven for a class of synchronized CF grammars; simple STDS (syntax-directed translation schemas) are closed under homomorphism and the projection of the languages in the translation are in the CFL class (Aho and Ullman, 1969b; Aho and Ullman, 1969a). In addition, (Shieber, 1994) claims that the weak generative capacity of Synchronous-TAGs remains in the mildly context-sensitive class when we add the requirement that the derivation structures of the

[3]Similarly, the mildly context-sensitive languages such as CCLs, TALs, LILs as well as Curried Multiset-CCGs are not closed under intersection with CFLs since $\{a^m b^n c^n d^n e^n | n, m \geq 0\} \cap \{a^n b^n c^* d^* e^* | n \geq 0\} = \{a^n b^n c^n d^n e^n | n \geq 0\}$ which is not a CCL or a Curried Multiset-CCL.

synchronized TAGs be isomophic. Similarly, we claim that the generative capacity of Multiset-CCG is unchanged, because of the parallel derivation process that yields one surface structure.

In Chapter 4, page 91, I present a CKY-parsing algorithm that allows parsing in polynomial time and space for a Multiset-CCG with finite sets of terminals and nonterminals. This parsing algorithm can be extended for the Multiset-CCG extended formalism by storing the ordering category of each constituent in the chart along with the constituent's syntactic category. Two constituents in the chart will only be allowed to combine if their syntactic categories and their ordering categories can combine with their respective rules. The ordering categories are always finite in size because the simple application and identity rules used by the ordering component do not increase the size of the categories during the derivation. Thus, the chart is still bounded by the size of the syntactic categories, in polynomial space. In addition, the rule applications on the ordering categories take constant time because the ordering categories are finite in size. Thus, the algorithm remains polynomial in space and time. We argue that the extended Multiset-CCG remains within a subclass of context-sensitive grammars and that it remains polynomially parsable.

6.2 The Question Answering System

I have implemented a simple data-base query task, diagrammed in Figure 6.4, in Quintus Prolog to demonstrate that Multiset-CCG can generate Turkish sentences with word orders appropriate to the context. The system simulates a Personal Assistant who schedules meetings and phone calls with a number of individuals. The user issues queries that the system answers using the contextually appropriate word order, by consulting the data-base and maintaining a model of the changing context.

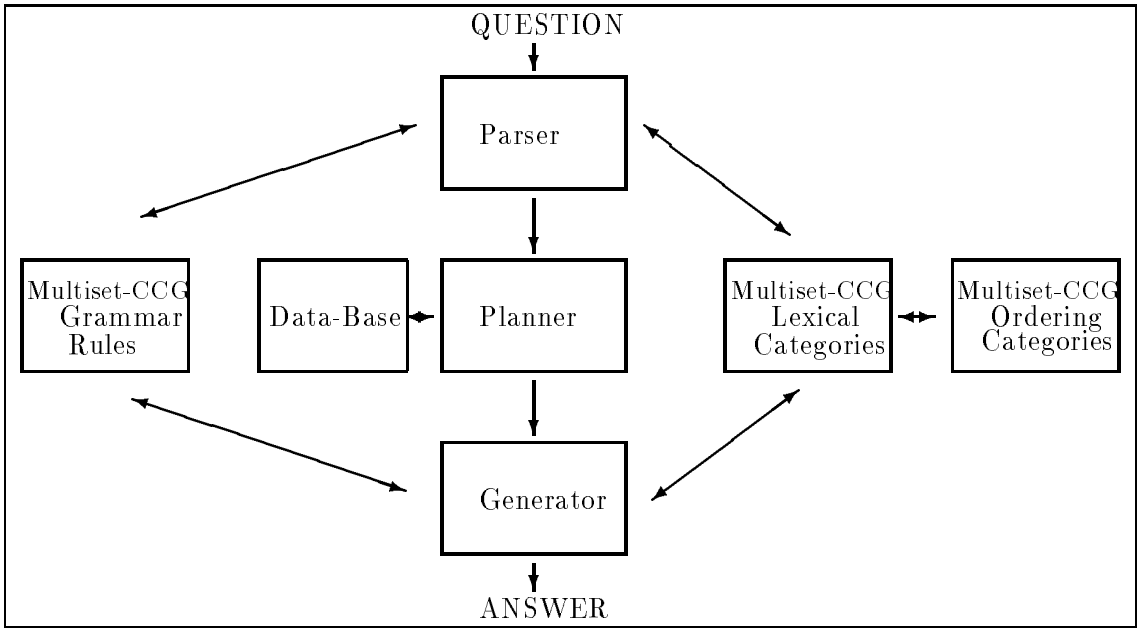


Figure 6.4: The Personal Assistant Generation System

Multiset-CCG is used by the parser and the generator to interpret the question and produce an answer. After the question is parsed, the parser's output is sent to the planning component of the generator. The sentence planner consists of simple plans for constructing answers to certain wh-questions and yes/no questions. Certain predicates in the queries trigger the planner to look up schedules and make appointments for the agents mentioned in the query. Then, the semantic representation and the information structure for the answer is sent to the generation component of the system which uses the input, the grammar, and the lexicon to construct an answer with the appropriate word order.

6.2.1 The Lexicon and Grammar

The lexicon in this system is not extensive; it contains about 150 Turkish words, which is adequate to demonstrate this working model of the theory. Since Turkish has agglutinative morphology,

there are hundreds of different morphological forms for each word. (Olfazer, 1993) provides a two-level morphological analyzer for Turkish based on the KIMMO system which could be added onto my system. However, I assume that we can do morphological analysis off-line to build a large-scale lexicon of Turkish words.

In the lexicon, each word is assigned a set of syntactic/semantic categories. The categories are stored in an abbreviated template form, for example the category for "gel-di-m" (come-Past-1Sing) is shown below in the form (Result ! { Arguments}):

```
[*syn(s,active,past,sing,1), *sem(event,E,[come(E,X),XProps],decl)] !
    { [*syn(n,sing,1,nom),*sem(kind,X,XProps)] }
```

The templates are expanded into feature-structures, DAGs, as needed during parsing or generation. The expanded form is saved so that we do not have to spend time doing the expansion the next time the same word is used during the same session. For example, the template above is expanded into the following DAG:

(21)
$$\begin{bmatrix} \text{Result} : \begin{bmatrix} \text{Syn} : \begin{bmatrix} \text{Cat} : \text{S} \\ \text{Tense} : \text{Past} \\ \text{Agr} : \begin{bmatrix} \text{Number} : \text{Sing} \\ \text{Person} : 1 \end{bmatrix} \end{bmatrix} \\ \text{Sem} : \begin{bmatrix} \text{Event} : E \\ \text{LF} : [\text{come}(E,X), XProps] \end{bmatrix} \end{bmatrix} \\ \text{Args} : \left\{ \begin{bmatrix} \text{Syn} : \begin{bmatrix} \text{Cat} : \text{NP} \\ \text{Case} : \text{Nom} \\ \text{Agr} : \begin{bmatrix} \text{Number} : \text{Sing} \\ \text{Person} : 1 \end{bmatrix} \end{bmatrix} \\ \text{Sem} : \begin{bmatrix} \text{Entity} : X \\ \text{Props} : XProps \end{bmatrix} \end{bmatrix} \right\} \end{bmatrix}$$

I have extended the traditional DAG unification algorithm to unify sets of DAGs to handle the multiset of arguments[4], and sets of functions such as *come(E,X)* in the semantic representation.

Each lexical category which contains syntactic/semantic features is associated with an ordering category during the parsing or generation process. For example, the lexical category for the verb

[4]We could instead associate each argument in the multiset with an extra feature label, e.g. N(nom), that indicates its category and case, and change the lexical representation of NPs and adjectives accordingly. Then, the multiset of arguments could just be represented as a traditional attribute-value matrix. This would also improve the efficiency of unification in parsing and generation with categories that have more than one argument, since then we could avoid full-scale unification with each argument in the multiset.

"came" which contains the features *syn* and *sem* and an empty information structure in the feature *info* is placed in the *category* feature in the DAG in Figure 6.5. This is associated with an ordering category that is unified in as the value of the feature *order* in Figure 6.5. The two categories are linked together by the co-index IS, which contains the template for the information structure of the sentence. This co-index ensures that at the end of the parsing or generation process, the category:result feature will contain the syntactic, semantic, and information structure features for the complete sentence.

$$\left[\begin{array}{ll}
\text{Category}: & \left[\begin{array}{ll}
\text{Result} : & \left[\begin{array}{ll}
\text{Syn} : & \left[\begin{array}{ll}\text{Cat} : \text{S} \\ \text{Tense} : \text{Past} \\ \text{Agr} : \left[\begin{array}{ll}\text{Number} : \text{Sing} \\ \text{Person} : 1\end{array}\right]\end{array}\right] \\
\text{Sem} : & \left[\begin{array}{ll}\text{Event} : E \\ \text{LF} : [\text{come}(E,X), XProps]\end{array}\right] \\
\text{Info} : & IS
\end{array}\right] \\
\text{Args} : & \left\{\left[\begin{array}{ll}
\text{Syn} : & \left[\begin{array}{ll}\text{Cat} : \text{NP} \\ \text{Case} : \text{Nom} \\ \text{Agr} : \left[\begin{array}{ll}\text{Number} : \text{Sing} \\ \text{Person} : 1\end{array}\right]\end{array}\right] \\
\text{Sem} : & \left[\begin{array}{ll}\text{Entity} : X \\ \text{Props} : XProps\end{array}\right]
\end{array}\right]\right\}
\end{array}\right] \\
\text{Order} : & IS\left[\begin{array}{ll}\text{Topic} : T \\ \text{Comment} : \left[\begin{array}{ll}\text{Focus} : F \\ \text{Ground} : [come(e), G2, G1]\end{array}\right]\end{array}\right] / G2 \backslash T \backslash G1 \backslash F
\end{array}\right]$$

Figure 6.5: The DAG associated with the Intransitive Verb "geldi" (came).

We can also specify pragmatic information directly in lexical categories. For example, the question particle "mi" used in yes/no questions in Turkish serves to focus as well as to question elements in a sentence. The lexical category for "mi" is a function which combines with some element to its left, $X\backslash X$. The lexical category marks this element X with the features *type: quest(yes/no,X)* in the *sem* feature and *focus: X* in the *info* feature.

The rules of the syntactic component of Multiset-CCG apply to the category contained in the *category* feature, and the rules of the ordering component apply to the ordering category contained in the *order* feature. Dag unification instead of term unification is used in the grammar rules in this implementation. For example, in the syntactic component of Multiset-CCG, the forward

application rule, written below in Quintus Prolog, tries to combine two categories by recursively selecting an argument from the argument multiset of the first category and unifying it with the second category. The resulting constituent inherits the result of the primary function category and the rest of the arguments in its multiset. The *result/2* function constructs this resulting category and rewrites a function with an empty set of arguments to be a basic element category, i.e. $(X|\emptyset \Rightarrow X)$.

```
%%% Forward Application:    Res|{Arg...}  Arg  => Res|{...}
reduce(Cat1, Cat2, ResCat, forward) :-
        path_value(Cat1,result,Res),
        path_value(Cat1,args,MSet),
        select(Arg,MSet,RestSet),
        unify_feature(Arg,dir,right),
        unify(Arg,Cat2),
        result(Res,RestSet,ResCat).
```

The composition rules are used to combine two functions, when one function subcategorizes for the other, for example a subordinate verb followed by a matrix verb. The resulting function inherits all the arguments in the multisets of both functions. As seen below in the backward composition rule, there is a restriction that ensures that verbs and adjectives cannot compose together as this would overgenerate ungrammatical word orders in Turkish as discussed in Chapter 3.

```
%%% Backward  Composition:   Arg|{...}   Res1|{Arg...} => Res1|{......}
reduce(Cat2, Cat1, ResCat, backward) :-
     path_value(Cat1,result,Res1), path_value(Cat1,args,Set1),
     path_value(Cat2,result,Res2), path_value(Cat2,args,Set2),
     restrict(Res1,Res2),      % if Res1 = S, Res2 <> NP
     select(Arg,Set1,RestSet),
     unify_feature(Arg,dir,right),
     unify(Arg,Res2),
     union(RestSet,Set2,Union),
     result(Res1,Union,ResCat).
```

6.2.2 The Parser

In my implementation, I use a shift-reduce parser with backtracking which prefers reducing to shifting in order to simulate left-to-right incremental processing.

```
%%% Shift-reduce parser with backtracking (Steedman)
%%% parse(Sentence,Stack,Result)
parse([],[Result],Result) :-
     complete(Result).
parse(Sentence, [Cat2, Cat1|Stack], Result) :-
     apply_rules(Cat1, Cat2, Cat3),
     parse(Sentence, [Cat3|Stack], Result).
parse([Word|Rest], Stack, Result) :-
      lexical_lookup(Word,Cat),
      parse(Rest, [Cat|Stack], Result).
```

There is a check at end of the parse to make sure that all the obligatory syntactic arguments and the obligatory information structure components have been found. The predicate *complete* simply checks whether the syntactic category and ordering category are basic elements rather functions that are looking for more arguments. The planner in my system can only handle questions that are complete sentences. However, in other applications, we may want to check just the completeness of the information structure and not the predicate-argument structure. For example, to handle zero pronouns in Turkish, we could allow the derivation of incomplete sentences and then allow further processing to infer the discourse referents of the missing arguments.

The parser above is not very efficient, but it can be made more efficient by adding a chart to keep track of already derived constituents. In Chapter 4, page 91, I present a CKY-parsing algorithm that allows parsing in polynomial time and space for a Multiset-CCG with finite sets of terminals and nonterminals. Note that this parsing algorithm can be extended for the Multiset-CCG with ordering presented in this chapter as discussed on page 168. The ordering category of each constituent can be stored in the chart along with the constituent's syntactic category because the ordering categories are always finite in size (the application rules only serve to decrease the size of the ordering categories during a derivation). The chart is bounded by the size of the syntactic categories, in polynomial space. Two constituents in the chart will be allowed to combine only if their syntactic categories and their ordering categories can combine. The application rules that combine the ordering categories take constant time because the ordering categories are finite in size. Thus, we conjecture that the algorithm remains polynomial in space and time.

However, once we allow the categories in Multiset-CCG to be feature-structures, the set of nonterminals grows much larger. This affects the running time, $O(n^{3+3|args(V_N)|})$, of the CKY

algorithm drastically, where $|args(V_N)|$ in the running time is the upperbound for the number of different feature-structures that can be subcategorized arguments in the multisets. However, the multisets in derived categories decrease or remain constant in size unless there is rampant long distance scrambling where we must keep track of many arguments at one time. For inputs without long distance scrambling, the maximum size of multisets in derived categories will be a constant k, the maximum number of arguments in lexical categories, giving a worst-case runtime of $O(n^{3+3k})$. Thus, the average runtime for the algorithm will be considerably faster than the worst-case time.

6.2.3 The Planner

6.2.3.1 Analyzing the Question

The Planner uses simple plans to handle the following types of questions:

1. **Wh-Questions:** "Who called Fatma?"
2. **Yes/No Questions:**
   (a) **Proposition Question:** "Did Ahmet call Fatma?"
   (question morpheme attached to verb in Turkish).
   (b) **Focused Question:** "Was it Ahmet who called Fatma?"
   (question morpheme attached to non-verbal elements in Turkish).
   (c) **Scheduling Question:** "Can I meet with Fatma today?"

For example, given the wh-question below, the parser returns the following DAG representation of the sentence which is then passed to the planner. The planner recognizes that this is the representation of a wh-question by looking at the *sem:type* feature which indicates that the Prolog variable _7350, which is the subject of the verb, is being questioned.[5] Each question type has a different value for this semantic type feature. The other Prolog variable _7349 refers to the Davidsonian event variable for the calling event.

(22) Fatma'yı kim aradı?
Fatma-Acc who seek-Past?
"As for Fatma, who called her?"

```
| ?- parse.
|: fatmayi kim aradi?

syn : cat : s
      voice : active
      tense : past
      agr : number : sing
            person : 3
sem : event : _7349
      type : quest(lambda(_7350))
      lf : [call(_7349,_7350,fatma),
            [person(_7350)],
            [one(fatma),specific(fatma,+),person(fatma)]]
```

[5]The question word is a type-raised NP which looks for a declarative verb and results in the appropriately marked interrogative sentence.

```
info :
       topic :  person(fatma)
       comment :
                 focus :  person(_7350)
                 ground : call(_7349,_7350,fatma)
```

The Prolog variables in the representation above must be unified with constants in the database (or get accommodated as new constants in the discourse model). Actually there needs to be a level of discourse processing after parsing in order to resolve the anaphora in sentences, perhaps using the Centering Algorithm (Grosz, Joshi, and Weinstein, 1983). This level of processing would find the antecedents for zero and overt pronouns and definite full NPs and accommodate new discourse entities for indefinite NPs, before the database is queried to fill in the questioned variable. This is not implemented in this system. For the sake of simplicity, I use only the database queries to bind the variables in the semantic representation.

The semantic representation of the question, in the *lf* feature, is used to query the database. The data base contains a set of predicates that are partitioned by topics. Using the file-card metaphor of (Heim, 1982), each topic is assigned a file in which predicates about that topic are recorded. For example, we may have the following information about Fatma in the database:

```
db_file(fatma, person(fatma)).
db_file(fatma, one(fatma)).
db_file(fatma, specific(fatma,+)).
db_file(fatma, call(e3,ayse,fatma)).
db_file(fatma, see(e4,fatma,ahmet)).
```

The information structure of the question is used by the planner to guide it in its search through the database. From the information structure, we know that the topic of the question is Fatma and therefore the information that we want to find is probably stored in Fatma's file in the database. We can pull up all the information in Fatma's file and try to match the predicates in the semantic representation of the question with this information. If this is successful, we will now have a set of predicates where the Prolog variables have unified with constants in the database, e.g.
[call(e3,ayse,fatma),person(ayse),one(fatma),specific(fatma,+),person(fatma)] for the question above. Predicates that are not recorded in Fatma's file, such as *person(ayse)* are then verified against the database without specifying a topic. If the match against the database fails to bind the questioned variable to a constant, we must generate a negative answer to the query, such as "Noone called Fatma, as far as I know".

The planner can also handle yes-no questions. These are marked as the semantic type *quest(yes/no,X)* where X can refer to an event variable (if the question particle is found next to the verb) or a different entity in the focus of the question, if the question particle is found next to a nonverbal element as in (23). The procedure above is followed to see whether the predicates in the *lf* feature of the question are satisfiable in the database. If they are satisfiable, we answer yes and repeat the statement in the question, otherwise the answer is no and the process below is followed.

(23) Fatma'yı Ahmet mi aradı?
Fatma-Acc Ahmet Quest seek-Past?
"Was it Ahmet who called Fatma?"

```
| ?- parse.
|: fatmayi ahmet mi aradi?

syn : cat : s
      voice : active
      tense : past
      agr : number : sing
            person : 3
sem : event : _9041
      type : quest(yes/no,ahmet)
      lf : [call(_9041,ahmet,fatma),
            [one(ahmet),specific(ahmet,+),person(ahmet)]
            [one(fatma),specific(fatma,+),person(fatma)]]
info :
       topic :  person(fatma)
       comment :
                 focus :  [one(ahmet),specific(ahmet,+),person(ahmet)]
                 ground : call(_9041,ahmet,fatma)
```

With focused yes/no questions such as (23), if the question is not validated in the data-base, the planner replaces the focus of the question with a variable and requests another search of the data-base to find a new focus which satisfies the rest of the question. For example, if (24)a is not satisfiable, then we query the database again to see if (24)b is satisfiable in the database. Thus, the focus of the question is also important in guiding the database search.

(24) a. [call(_9041,ahmet,fatma), [one(ahmet),specific(ahmet,+),person(ahmet)]
[one(fatma),specific(fatma,+),person(fatma)]]
b. [call(_9041,X,fatma), [one(fatma),specific(fatma,+),person(fatma)]]

If the second query is successful, the following cooperative answer can be generated. Otherwise, the negated version of question is generated by negating the verb and instantiating all the variables in the sentence with new constants, using the *numbervars* in Prolog.

(25) a. Fatma'yı Ahmet mi aradı?
Fatma-Acc Ahmet Quest seek-Past?
"Was it Ahmet who called Fatma?"

b. Hayır, Fatma'yı Ayşe aradı,
No, Fatma-Acc Ayşe seek-Past.
"No, it was Ayşe who called Fatma."

The planner can also handle yes-no scheduling questions. These questions are marked in the semantic type feature as requests. They contain a verb with the abilitative morpheme "bil" which corresponds to "can" in English as well as a question particle next to the verb. The planner uses the semantic representation in a different way during the database query in order to determine whether it can schedule the meeting requested and provide a cooperative answer such as (26)b.

(26) a. Fatma'yı göre-bil-ir-mi-yim ben?
Fatma-Acc see-Abil-Aor-Quest-1S I?
"Can I see FATMA?"

b. Evet, Fatma'yı siz üç-te göre-bil-ir-sin-iz.
Yes, Fatma-Acc you(polite) three-Loc see-Abil-Aor-2nd-Pl.
"Yes, you can see Fatma at THREE."

```
| ?- parse.
|: fatmayi gorebilirmiyim ben?

syn : cat : s
      voice : active
      tense : aorist
      agr : number : sing
            person : 1
      compound : abilitive
sem : event : _7371
      type : request(_7371,see)
      lf : [see(_7371,user,fatma)
            [one(user),specific(user,+),person(user)]
            [one(fatma),specific(fatma,+),person(fatma)]]
info :
       topic : recoverable
       comment :
                 focus :  person(fatma)
```

```
ground : [see(_7371,user,fatma), person(user)]
```

Given a scheduling request like the one above, the planner determines whether the participants in the requested event are busy during time and day that are either specified in the question or are assumed to be the current day and hour (using the *time_stamp* command in Prolog). Whenever a meeting is scheduled, the participants are all assigned the predicate *busy(Participant,Event)* in the database. For example, if the current day and hour is Monday and 3 o'clock, the planner queries the database for each participant to see whether that participant is busy Monday at 3, e.g.

(27) a. for some event E [day(E,monday),time(E,3),busy(E,fatma)]

b. or for some event F [day(F,monday),time(F,3),busy(F,user)].

If neither of the participants are busy during the proposed day and time, we can generate the appropriate response seen in (26). If the participants are busy during a proposed date, the planner tries to reschedule by proposing the next hour and querying the database again to see if the participants are busy during that hour.

6.2.3.2 Planning the Answer

The planner creates a representation for the answer by copying much of the question representation and by adding the appropriate new information found in the database. The parsed representation of the question contains syntactic, semantic, and pragmatic information in the features *syn, sem,* and *info*. These features are constructed for the answer as well.

The syntactic information, i.e. the category, tense, voice, and agreement features, is directly copied from the question to the answer. The only change is that the first person and second person agreement features are switched in the answer.

The semantic information is also copied, however the semantic type is changed from a question type to a *declarative*, and extra predicates are added in order to describe the new focus in the answer. The new focus may be a new discourse entity that must be described, for example in answer to a wh-question or a focused yes/no question, or it can be the negated verb in answer to a yes/no question, or it can be a proposed time or day for a meeting in scheduling questions.

The information structure constructed for the answer specifies its topic and its focus; this information will be used to determine the answer's word order during the generation process. The information found in the database lookup that is triggered by the question is specified to be the focus of the answer. In order to maintain topic continuity in the question-answer pair, the topic of the question is copied directly to the answer. In a different domain, we would also need an algorithm that allows for shifts in topic. Further research is needed to model the

complex inferencing processes that determine the relationships between the topics and foci from one sentence to another in a longer discourse.

The extra information added to the semantics and the focus of the information structure in the answer are determined by more queries to the database and discourse model. For example, given the wh-question in (28)a , the planner may match its semantic representation with information in the database to retrieve the following set of predicates:
[call(e3,s1,fatma),person(s1),one(fatma),specific(fatma,+),person(fatma)].
We will now have to find out more about the constant $s1$ in order to generate the answer in (28)b.

(28) a. Fatma'yı kim aradı?
Fatma-Acc who seek-Past?
"As for Fatma, who called her?"

b. Fatma'yı bir Türk öğrenci aradı.
Fatma-Acc one Turkish student seek-Past.
"As for Fatma, it was a Turkish student who called her."

The semantic properties of the focused entity, s1, are found by the Entity-Describer module of the planner which works as follows. If the discourse entity is a constant that refers to a proper name, e.g. fatma, then the constant is sufficient to describe that individual. However, if the constant is not a proper name, e.g. s1, then we need to construct a full NP description of the entity by either consulting the database or discourse model. We describe discourse-old entities which are already in the discourse-model by just copying the semantic properties expressed for that entity in the discourse model. Discourse-new entities are described by consulting the database. For example, for the constant s1, the database search produces the following information:

```
db_file(s1, student(s1)).
db_file(s1, turkish(s1)).
db_file(s1, one(s1)).
db_file(s1, person(s1)).
db_file(s1, go(e7,s1)).
```

The database query may return many predicates associated with s1. Choosing the relevant properties is not an easy job. I simply filter out all but the one-place predicates; this selects the adjectival properties but not the verbal ones (which unfortunately rules out the generation of relative clauses). What we actually need is a filtering mechanism that keeps only enough properties to distinguish it from other discourse entities in the discourse model as in (Dale, 1992).

We also need a separate algorithm, perhaps Centering Theory (Grosz, Joshi, and Weinstein, 1983; Turan, 1995), to use zero and overt pronouns in Turkish. In the implemented system, I only

handle first and second person overt pronouns, and all arguments are realized in the generated response so that we can determine whether the system is generating contextually-appropriate word orders. The system uses dynamic predicates *speaker(S)* and *hearer(H)* to keep track of whether the system or the user is speaking. When the system is interpreting a question from the user, the variable S is set to the constant *user* and H is set to *system*. However, when the system is generating the answer to the question, S is set to *system* and H to *user*. The lexical categories for "I" and "you" use the dynamic predicates to refer to the right discourse entity, system or user, depending on who is the speaker and hearer at that moment.[6]

Figure 6.6 shows the representation containing syntactic, semantic, and pragmatic features constructed by the planner for the answer to the wh-question below. The planner then passes this representation to the generator described in the next section in order to produce a string of words with the appropriate word order.

(29) a. Fatma'yı kim aradı?
Fatma-Acc who seek-Past?
"As for Fatma, who called her?"

b. Fatma'yı bir Türk öğrenci aradı.
Fatma-Acc one Turkish student seek-Past.
"As for Fatma, it was a Turkish student who called her."

$$
\left[\begin{array}{ll}
\text{Cat:Res:} & \left[\begin{array}{ll}
\text{Syn} : & \left[\begin{array}{ll} \text{Cat} : \text{S} \\ \text{Tense} : \text{Past} \end{array}\right] \\
\text{Sem} : & \left[\begin{array}{ll} \text{Event} : \text{e3} \\ \text{LF} : & [\text{call(e3,s1,fatma)}, \\ & \ [\text{one(fatma), specific(fatma,+), person(fatma)}], \\ & \ [\text{one(s1), student(s1), turkish(s1),person(s1)}]] \end{array}\right] \\
\text{Info} : & \left[\begin{array}{ll} \text{Topic} : & [\text{person(fatma)}] \\ \text{Comment} : & \left[\begin{array}{ll} \text{Focus} : & [\text{one(s1), student(s1), turk(s1),person(s1)}] \\ \text{Ground} : & [\text{seek(e3,fatma,s1)}] \end{array}\right] \end{array}\right]
\end{array}\right]
\end{array}\right]
$$

Figure 6.6: Input to the Generation Algorithm.

[6]This approach to deictics simplifies the situation somewhat because it cannot handled reported speech, for example, "I" can refer to the speaker in a reported situation rather than the current speaker, e.g. "Mary said, "I'm here"".

6.2.4 The Generator

I have adapted a head-driven bottom up generation algorithm (Calder, Reape, and Zeevat, 1989; Shieber et al., 1989; van Noord, 1990) for CCGs. A pure bottom-up generation algorithm finds all lexical items that have an interpretation that takes part in the semantics of the input and tries to combine them into a grammatical sentence. This approach is not feasible, because if we try combining the lexical items in every way possible, there are $n!$ different combinations to try while generating a sentence of length n. On the other hand, a pure top-down generation algorithm also has problems. Such an algorithm operates by running the grammar backwards; the input is taken as the result of applying a rule, and then the daughter constituents are guessed, filling out the tree until reaching the leaves of the tree that contain the lexical items. However, if there are recursive rules in the grammar such as composition, the generator may not terminate.[7] The generation algorithm presented here combines aspects of both top-down and bottom-up generation. It takes advantage of the bottom-up lexical information as well the top-down input provided by the planner. The algorithm is well-suited for lexicalist formalisms like CCG, where most of the information is stored in the lexicon rather than in grammar rules.

The algorithm for the head-driven bottom up generator for CCGs is presented below in Prolog code:

```
generate(Input) :-
      find_lex_functor(Input,LexDag),
      bup_generate(Input,LexDag).

bup_generate(Input,LexDag):- unify(Input,LexDag).

bup_generate(Input, LexDag) :-
        combine(Arg, LexDag, ResDag, backward),
        generate(Arg),
        order(Arg, LexDag, ResDag),
        concat_phons(Arg, LexDag, ResDag),
        bup_generate(Input, ResDag).

bup_generate(Input, LexDag) :-
        combine(LexDag, Arg, ResDag, forward),
        generate(Arg),
        order(LexDag, Arg, ResDag),
        concat_phons(LexDag, Arg, ResDag),
        bup_generate(Input, ResDag).
```

[7] (Prevost and Steedman, 1993) presents a top-down generation algorithm for an English CCG with prosodic information which solves this problem by restricting the depth in following recursive rules.

The algorithm works as follows. First, the function **generate** finds a category in the lexicon which is the head of the sentence. Then in **bup-generate**, we try to apply the combinatory grammar rules (i.e. the forward and backward Multiset-CCG rules) to this lexical functor to generate its arguments in a bottom-up fashion. The order function applies the ordering rules to the functor and argument to make sure that they form a constituent in the information structure.[8] The **bup-generate** function is called recursively on the result of applying the rules until it has found all of the head functor's (*LexDag*) arguments, eventually resulting in something which unifies with the *Input*. The derivation in the generation process looks much like the derivations produced by bottom-up parsing except that the feature-structures are incomplete until all the lexical information is found.

According to (Wedekind, 1988), a generation algorithm is complete if the input to the generator to generate a string subsumes (i.e. is more general than) the description produced by the grammar for that string (LexDag) at the end of generation, and coherent if the grammatical description (LexDag) subsumes the Input. The generation algorithm above does not check whether the LexDag and the Input are subsumed by one another, but it does check whether they are compatible by unification, i.e. some Dag that subsumes both exists and this Dag is the most general one possible. We could easily replace this unification check with subsumption checks. However, I think it is useful to allow the Input specification to be more general than the grammatical description; certain lexical features of individual words may not need to be specified in the input. The algorithm would be incoherent if it generated incorrect strings such as "Mary met a smart graduate student" for an input that only specifies the information in the sentence "Mary met a student". However, this is not allowed in my algorithm because of the way the semantic information is structured. First of all, variables in the logical form of the input are not true Prolog variables, but Prolog constants. When the planner looks up the question in the database, all true Prolog variables in the logical form are unified with individual constants in the database or accommodated with new constants (using the numbervars predicate in Quintus Prolog). For example, if the question is "Did Fatma meet with a student?" with the logical form *meet(E,Fatma,Y) & student(Y)*, the database query may bind the variable Y to the student *Ayşe* or some individual *s1* whose name we do not know and the event E to some individual event *e1*. The resulting logical form for the answer "Yes, Fatma did meet a student" is *meet(e1,Fatma,s1) & student(s1)*. During generation, these constants unify with the variable arguments in the logical form of the lexical entries. In addition, I represent the logical form as a set instead of a Dag in Prolog, i.e. it does not have a variable at the end of the list [...|_] that would allow additional features to be unified in. This

[8] Note that order and concat-phons must be called after we have lexically instantiated both Arg and LexDag to avoid infinite loops. The UCG algorithm also freezes such features until the argument is instantiated.

ensures that the algorithm is coherent with respect to preserving the logical form in the input.

The main difference between this CCG algorithm and previous head-driven bottom-up generation algorithms is that this algorithm uses all of the information (syntactic, semantic, and information structure features) given in the input, instead of using only the semantic information, to find the head functor in the lexicon. This is possible because of the formulation of the CCG rules. CCG derivations are monotonic in that the head daughter in each rule (shown in bold in the following Multiset-CCG rules) shares its function result (X) with the final result after applying the rule. Thus, if a lexical function category is to take part as a head functor in a CCG derivation that produces a set of features (the input to the generator), the function's result must unify with that set of features.

(30) a. $\mathbf{X}|(Args \cup \{\vec{Y}\}) \quad Y \Rightarrow \mathbf{X}|Args$

b. $Y \quad \mathbf{X}|(Args \cup \{\vec{Y}\}) \Rightarrow \mathbf{X}|Args$

c. $\mathbf{X}|(Args1 \cup \{\vec{Y}\}) \quad Y|Args2 \Rightarrow \mathbf{X}|(Args1 \cup Args2)$

d. $Y|Args2 \quad \mathbf{X}|(Args1 \cup \{\vec{Y}\}) \Rightarrow \mathbf{X}|(Args1 \cup Args2)$

A chart can be added to this algorithm as in (Calder, Reape, and Zeevat, 1989) to improve the efficiency of generation by saving the generated bottom-up constituents in case they are needed again. The lexical search for the head functor can also slow down the algorithm if full DAG unification is used. To improve the efficiency of the lexical search for the head functor in my implementation, **find-lex-cat** first finds a rough match in the lexicon using term-unification. We associate each item in the lexicon with a semantic key-predicate that is one of the properties in its semantic description.[9] A lexical entry roughly matches the input if its semantic key-predicate is a member of the list of semantic properties given in the input. After a rough match using term-unification, **find-lex-cat** unifies the DAGs containing all of the known syntactic, semantic, and pragmatic information for the most embedded result of the lexical category and the result of the input to find the lexical category which is the head functor. Then, the rules can be applied in a bottom up fashion assuming that the found lexical category is the head daughter in the rules.

For example, given the input to the generator shown in Figure 6.6, we find the head functor "seek" in the lexicon. This functor's result unifies with the input to give us the LexDag in Figure 6.7. Note that the Input to the generation algorithm helps to pick out the correct syntactic category as well as ordering category (e.g. the contraint that the verb must be in the ground limits the ordering category that we can use).

Now we can use the Multiset-CCG rules to generate each of the arguments of this function. We call the rules with LexDag as the primary category and the Input as the resulting category,

[9]This key predicate is also used as the interpretation in the ordering categories as a shorthand for the complete semantic representation.

and the rule returns the secondary category that must have participated in the derivation. The only derivation that satisfies the syntactic, semantic, and informational constraints generates a sentence with the following word order.

(31) Fatma'yı bir Türk öğrenci aradı.
Fatma-Acc one Turkish student seek-Past.
"As for Fatma, it was a Turkish student who called her."

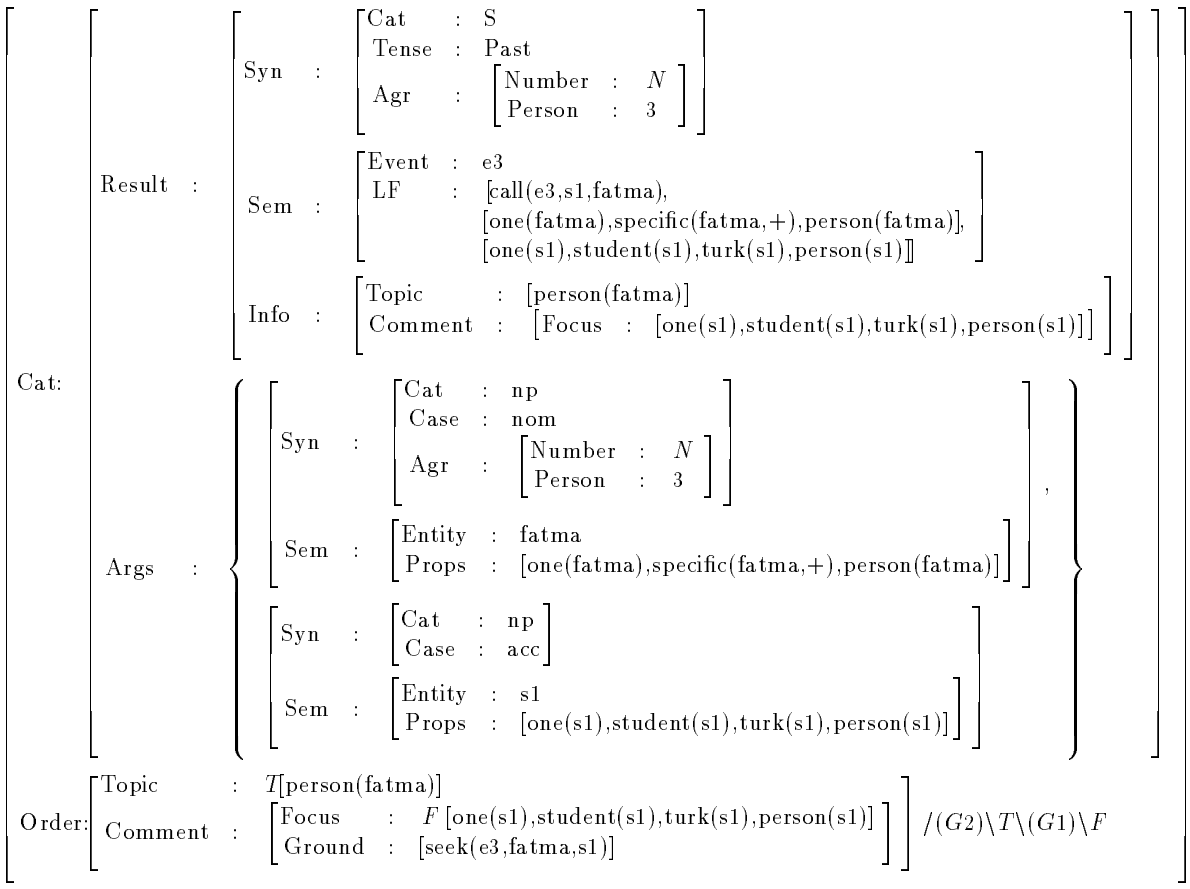


Figure 6.7: LexDag after the lexical entry for "seek" is unified with the Input.

6.2.5 Sample Runs

The following sample runs of the generation system further demonstrate how context-appropriate word orders are generated using Multiset-CCG. The questions are the user's input and the answers are the Personal Assistant system's replies. Each question-answer pair is followed by the DAG representation of the answer.

For example, in the following pairs of wh-questions, the answers have the same syntactic and truth-conditional semantic features, but since they are generated in different contexts, they have different information structures expressed by different word orders.

(32) a. Fatma kimi ara-dı?
Fatma who seek-Past?
"Whom did Fatma call?"

b. Fatma Ayşe'yi aradı. SOV
Fatma Ayşe-Acc seek-Past.
"Fatma called Ayşe."

(33) a. Ayşe'yi kim ara-dı?
Ayşe-Acc who seek-Past.
"As for Ayşe, who called her?"

b. Ayşe'yi Fatma aradı. OSV
Ayşe-Acc Fatma seek-Past.
"As for Ayşe, it is FATMA who called her."

```
| ?- loop.
|: fatma kimi aradi?

Answer:  fatma ayseyi aradi.
ReplyDag:

sem : event : e4
      type : decl
      lf : [call(e4,fatma,ayse),
            [one(fatma),def(fatma,+),person(fatma)],
            [person(ayse),one(ayse),def(ayse,+)]]
syn : cat : s
      voice : active
      tense : past
      agr : number : sing
            person : 3
info : topic : person(fatma)
       comment :
                 focus : one(ayse)
                         def(ayse,+)
                         person(ayse)
                 ground : call(e4,fatma,ayse)


|: ayseyi kim aradi?

Answer:  ayseyi fatma aradi.
ReplyDag:
sem : event : e4
      type : decl
      lf : [call(e4,fatma,ayse),
            [one(fatma),def(fatma,+),person(fatma)],
            [person(ayse),one(ayse),def(ayse,+)]]
syn : cat : s
      voice : active
      tense : past
      agr : number : sing
            person : 3
info : topic :  person(ayse)
       comment :
                 focus : one(fatma)
                         def(fatma,+)
                         person(fatma)
                 ground : call(e4,fatma,ayse)
```

An example of a yes-no question where the verb is focused is seen below.

(34) a. Ben-i Ayşe ara-dı mı?
I-Acc Ayşe seek-Past-Quest?
"Did Ayşe call me?"

b. Hayır, siz-i Ayşe ara-ma-dı. (OSV)
No, you-Acc Ayşe seek-Neg-Past.
"No, Ayşe didn't call you."

```
|: beni ayse aradimi?

Answer: hayir, sizi ayse aramadi.

ReplyDag:

sem : event : W3
      type : decl
      lf : [neg(W3),call(W3,ayse,user),
            [one(ayse),def(ayse,+),person(ayse)],
            [one(user),def(user,+),person(user)]]
syn : cat : s
      voice : active
      tense : past
      agr : number : sing
            person : 3
info : topic : person(user)
       comment :
                 focus : [neg(W3), call(W3)]
                 ground : person(ayse)
```

A yes-no question where the immediately preverbal element is focused in the question by the question particle is shown below. The new, contrastive information is focused.

(35) a. Ahmet'i Ayşe mi aradı?
Ahmet-Acc Ayşe Quest seek-Past?
"As for Ahmet, was it Ayşe who called him?"

b. Hayır, Ahmet'i bir türk oğrenci aradı. (OSV)
No Ahmet-Acc one Turkish student seek-Past.
"No, as for Ahmet, it was a Turkish student who called him."

```
|: ahmeti ayse mi aradi?


Answer:  hayir, ahmeti bir turk ogrenci aradi.
ReplyDag:

sem : event : e8
      type : decl
      lf : [call(e8,s1,ahmet),
            [one(ahmet),def(ahmet,+),person(ahmet)],
            [person(s1),def(s1,-),one(s1),student(s1),turkish(s1)]]
syn : cat : s
      voice : active
      tense : past
      agr : number : sing
            person : 3
info :
       topic :  person(ahmet)
       comment :
                 focus : def(s1,-)
                         one(s1)
                         person(s1)
                         student(s1)
                         turkish(s1)
                 ground : call(e8,s1,ahmet)
```

A scheduling question is demonstrated below. The new information about the time of the meeting is focused in the answer.

(36) a. Fatma Ayşe'yi görebilirmi?
Fatma Ayşe-Acc see-abil-aor-quest?
"Can Fatma see Ayşe?"

b. Evet, Fatma Ayşe'yi ikide görebilir.
Yes, Fatma Ayşe-Acc two-Loc see-abil-aor.
"Yes, Fatma can see Ayşe at TWO."

```
| ?- loop.
|: fatma ayseyi gorebilirmi?

Answer: evet, fatma ayseyi ikide gorebilir.
ReplyDag:

syn : cat : s
      voice : active
      tense : aorist
      agr : number : sing
            person : 3
      compound : abilitive
sem : type : decl
      event : E5
      lf : [time(E5,2), see(E5,fatma,ayse),
            [one(fatma),def(fatma,+),person(fatma)],
            [one(ayse),def(ayse,+),person(ayse)]]
info : topic :  person(fatma)
       comment :
                 focus :  time(E5,2)
                 ground : [see(E5,fatma,ayse), person(ayse)]
```

A complex sentence with an embedded information structure is shown below:

(37) a. Ayşe'nin gel-diğ-i-ni kim bil-iyor?
Ayşe-Gen come-Ger-3S-Acc who know-Prog?
"Who knows that Ayşe has arrived?"

b. Ayşe'nin gel-diğ-i-ni Fatma bil-iyor.
Ayşe-Gen come-Ger-3S-Acc Fatma know-Prog.
"It is FATMA who knows that Ayşe has arrived."

```
| ?- loop.
|: aysenin geldigini kim biliyor?


Answer:  aysenin geldigini fatma biliyor.
ReplyDag:

syn : cat : s
      voice : active
      tense : pres
      agr : number : sing
            person : 3
sem : event : e6
      type : decl
      lf : [know(e6,fatma,e2),
            [person(fatma),one(fatma),def(fatma,+)],
            [come(e2,ayse),[one(ayse),def(ayse,+),person(ayse)]]]

info :
       topic :
               topic : recoverable
               comment :
                focus : person(ayse)
                        ground : come(e2,ayse)
       comment :
                 focus : one(fatma)
                         def(fatma,+)
                         person(fatma)
                 ground :  know(e6,fatma,e2)
```

A sentence with long distance scrambling is shown below:

(38) a. Ahmet'i Fatma kim-in ara-dığ-ı-nı söyle-di?
Ahmet-Acc Fatma who-Gen call-Ger-3S-Acc say-Past.
"As for Ahmet, who did Fatma say called him?"

b. Ahmet'i Fatma bir Türk öğrenci-nin ara-dığ-ı-nı söyle-di?
Ahmet-Acc Fatma one Turkish student-Gen call-Ger-3S-Acc say-Past.
"As for Ahmet, Fatma said it was a Turkish student who called him."

```
| ?- loop.

|: ahmeti fatma kimin aradigini soyledi?

Answer:  ahmeti fatma bir turk ogrencinin aradigini soyledi.

ReplyDag:

syn : cat : s
      voice : active
      tense : past
      agr : number : sing
            person : 3
sem : event : e10
      type : decl
      lf : [say(e10,fatma,e8),
            [one(fatma),def(fatma,+),person(fatma)],
            [call(e8,s1,ahmet),
             [def(s1,-),one(s1),turkish(s1),student(s1),person(s1)],
             [one(ahmet),def(ahmet,+),person(ahmet)]]]
info :
       topic : person(ahmet)
       comment :
                 focus :
                         topic : recoverable
                         comment :
                                   focus : one(s1)
                                           turkish(s1)
                                           student(s1)
                                   ground : call(e8,s1,ahmet)
                 ground : [say(e10,fatma,e8),person(fatma)]
```

Chapter 7

Conclusions

"Free" word order languages reveal that there is an important aspect of meaning that is context-dependent. In these languages, the word order directly reflects the information structure of the sentence, which captures differences in meaning that make a sentence only appropriate in certain contexts. In fact, this aspect of meaning exists in all languages, but it is expressed in different ways. In fixed word order languages like English, prosody is the main method used to express the information structure, whereas in languages like Turkish, word order is the main method to express the information structure of a sentence. Thus, this additional aspect of context-dependent meaning must be incorporated into theories of grammar in order to capture the distinctions in meaning necessary for effective communication and translation in all languages.

The novel contributions of this dissertation are in four areas:

- **Linguistic Analysis** of the syntactic, pragmatic, and formal properties of Turkish word order.
- **Development of Multiset CCG**, a novel formalism that:
  - Integrates Information Structure with a syntactic theory of grammar by deriving the AS and IS of sentences in parallel.
  - Handles the freedom as well as restrictions in word order in complex sentences with long distance dependencies.
- **Formal Analysis** of Multiset-CCG's weak generative capacity and development of a polynomial-time parsing algorithm for Multiset-CCG.
- **A Computational Application** using Multiset-CCG to interpret Turkish questions and generate answers with contextually appropriate word orders.

In Chapter 2, I outlined the formal and descriptive properties that a formalism needs in order to capture "free" word order in simple and complex sentences with long distance dependencies and discontinuous constituents. In Chapter 3, I presented the syntactic component of Multiset CCG which is flexible enough to derive the predicate-argument structure of simple and complex sentences without relying on word order, and yet expressive enough to capture syntactic restrictions on word order in different languages such as languages with NP or clausal islands or languages which allow discontinuous NPs or clauses. In Chapter 4, I investigated the weak generative capacity of different versions of Multiset-CCG. I showed that Multiset-CCG is context-sensitive but argue that it does not have the full power of context-sensitive grammars or Indexed Grammars. In addition, I presented a polynomial-time parsing algorithm for Multiset-CCG in which the processing time increases proportionally to the amount of long distance scrambling in the input sentences. Thus, I have shown that Multiset-CCG is an computationally attractive formalism.

In Chapter 5, I investigated naturally-occurring Turkish discourses in order to determine the information structure in Turkish sentences. In Chapter 6, I added the ordering component to Multiset CCG which specifies the ordering of information structure components in a sentence. This part of the grammar captures the context-dependent interpretation of word order variation among arguments, adjuncts, and across clause boundaries. It allows adjuncts and elements from embedded clauses to take part in the information structure of the matrix clause, even though they are not arguments in its predicate-argument structure. It also allows embedded information structures to capture the interpretation of free word order in embedded clauses.

The dissertation presents an integrated grammar that captures the syntax as well as the context-dependent interpretation of "free" word order in Turkish. A novel characteristic of Multiset CCG is that it compositionally derives the predicate-argument structure and the information structure of a sentence in parallel. Multiset-CCG captures the context-appropriate use of word order because its flexible surface structure allows syntactic constituents to correspond to information structure constituents. Every Multiset CCG category encoding syntactic and semantic properties is associated with an ordering category that encodes the ordering of information structure components such as topic and focus; two syntactic/semantic categories are allowed to combine to form a larger constituent only if their ordering categories can also combine. Thus, the grammar uses both syntactic and pragmatic constraints to determine the surface structure and word order of the sentence.

This integrated grammar is of considerable importance for practical applications in natural language interpretation, generation, and machine-translation in "free" word order languages. The advantages of my formalism can be seen in an implemented system which generates answers

to data-base queries using contextually appropriate word orders. A formalism that integrates information structure and syntax such as Multiset-CCG is essential to the computational task of interpreting and generating simple and complex sentences with contextually appropriate word orders in "free" word order languages.

Bibliography

A.E. Ades and M. Steedman. 1982. On the order of words. *Linguistics and Philosophy*, 4:517–558.

A. V. Aho and J. D. Ullman. 1969a. Properties of syntax directed translations. *Journal of Comput. Syst. Science*, 3(3):319–334.

A. V. Aho and J. D. Ullman. 1969b. Syntax directed translations and the pushdown assembler. *Journal of Comput. Syst. Science*, 3(1):37–56.

A. V. Aho. 1968. Indexed grammars – an extension to context free grammars. *Journal of the ACM*, 15:647–671.

Judith Aissen. 1979. *The Syntax of Causative Constructions.* Garland Publishing Inc., New York.

Kazimierz Ajdukiewicz. 1935. Die syntaktische Konnexität. *Studia Philosophica*, 1:1–27. English translation in Storrs McCall (ed), *Polish Logic 1920-1939*, Oxford University Press, pp. 207-231.

Emmon Bach. 1988. Categorial grammars as theories of language. In Richard Oehrle, Emmon Bach, and Deirdre Wheeler, editors, *Categorial Grammars and Natural Language Structures*, pages 17–34. Reidel, Dordrecht.

Mark Baker. 1988. *Incorporation: A Theory of Grammatical Function Changing.* University of Chicago Press, Chicago.

Yehoshua Bar-Hillel. 1953. A quasi-arithmetical notation for syntactic description. *Language*, 29.

Josef Bayer and Jaklin Kornfilt. 1994. Against scrambling as an instance of move-alpha. In Henk van Riemsdijk and Norbert Corver, editors, *Studies on scrambling. Movement and non-movement approaches to free word-order phenomena*, Studies in generative grammar 41, pages 17–60. Mouton de Gruyter, Berlin.

Tilman Becker, Aravind Joshi, and Owen Rambow. 1991. Long distance scrambling and tree adjoining grammars. In *Proceedings of the 5th conference of the European Chapter of ACL.*

Betty J. Birner. 1994. Information status and english inversion. *Language*, 70(2):1–32.

Gosse Bouma and Gertjan van Noord. 1994. Constraint-based categorial grammar. In *Proceedings of the 32nd Annual Meeting of Association for Computational Linguistics*, pages 147–154.

Gosse Bouma. 1985. Warlpiri wildness: A categorial study of free word order. Masters Thesis, Instituut voor Algemen Taalwetenschap, Rijksuniversiteit Groningen.

Susan E. Brennan, Marilyn W. Friedman, and Carl Pollard. 1987. A centering approach to pronouns. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics*, pages 155–162, Stanford, CA.

Joan Bresnan and Ronald Kaplan. 1982. Lexical-functional grammar: A formal system for grammatical representation. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*. MIT Press.

J. Calder, M. Reape, and H. Zeevat. 1989. An algorithm for generation in unification categorial grammars. In *Proceedings of the 4th Conference of the European ACL*, pages 233–40.

Wallace Chafe. 1976. Givenness, contrastiveness, definiteness, subjects, and point of view. In Charles Li, editor, *Subject and Topics*, pages 25–55. Academic Press, New York.

Noam Chomsky and Howard Lasnik. 1977. Filters and control. *Linguistic Inquiry*, 8:425–504.

Noam Chomsky. 1971. Deep structure, surface structure, and semantic interpretation. In D. Steinberg and L. Jacobovits, editors, *Semantics*. Cambridge University Press.

Noam Chomsky. 1981. *Lectures in government and binding*. Studies in generative grammar 9. Foris Press, Dordrecht.

Noam Chomsky. 1993. A minimalist program for linguistic theory. In Kenneth Hale and Samuel J. Keyser, editors, *The view from Building 20*, pages 1–52. MIT Press, Cambridge, MA.

Bernard Comrie. 1978. Definite direct objects and referent identification. *Pragmatics Microfiche*.

Peter Culicover. 1980. Adverbials and stylistic inversion. *Social Science Working Papers 77*.

Haskell B. Curry and R. Feys. 1958. *Combinatory Logic: Vol I*. Amsterdam: North-Holland.

Robert Dale. 1992. *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes*. MIT Press.

Müşerref Dede. 1986. Definiteness and referentiality in Turkish verbal sentences. In Dan Slobin and Karl Zimmer, editors, *Studies in Turkish Linguistics*, pages 147–164. John Benjamins Publishing Company.

Hans den Besten. 1985. The ergative hypothesis and free word order in Dutch and German. In Jindřich Toman, editor, *Studies in German grammar*, Studies in generative grammar 21, pages 23–64. Foris, Dordrecht.

David Dowty, Robert Wall, and Stanley Peters. 1981. *Introduction to Montague Semantics.* Reidel, Dordrecht.

David Dowty. 1982. Grammatical relations and Montague grammar. In P. Jacobson and G. Pullum, editors, *The Nature of Syntactic Representation.* Reidel, Dordrecht.

David Dowty. 1988. Type raising, functional composition, and non-constituent conjunction. In Richard Oehrle et al, editor, *Categorial Grammars and Natural Language Structures*, pages 153–197. D. Reidel.

David Dowty. 1989, revised 1991. Toward a minimalist theory of syntactic structure. In *Proceedings of the Tilburg Conference on Discontinuous Constituency.*

Mürvet Enç. 1991. The semantics of specificity. *Linguistic Inquiry*, 22:1–25.

Elizabet Engdahl and Enric Vallduvi. 1994. Information structure and grammar architecture. NELS, University of Pennsylvania.

Eser Emine Erguvanlı. 1984. *The Function of Word Order in Turkish Grammar.* University of California Press. UCLA PhD dissertation 1979.

Eser Emine Erguvanlı. 1987. The role of semantic features in Turkish word order. *Folia Linguistica*, 21:215–227.

Feride Erkü. 1983. *Discourse Pragmatics and Word Order in Turkish.* Ph.D. thesis, University of Minnesota.

Nomi Erteschik-Shir and Shalom Lappin. 1979. Dominance and the functional explanation of island phenomena. *Theoretical Linguistics*, 6:41–85.

Gerald Gazdar, Ewan Klein, George Pullum, and Ivan Sag. 1985. *Generalized Phrase Structure Grammar.* Harvard University Press.

G. Gazdar. 1988. Applicability of indexed grammars to natural languages. In U. Reyle and C. Rohrer, editors, *Natural Language Parsing and Linguistic Theories.* D. Reidel, Dordrecht.

S. Greibach. 1965. A new normal form theorem for context-free phrase structure grammars. *Journal of the Association for Computing Machinery*, 12(1):42–52.

Gunther Grewendoorf and Wolfgang Sternefeld. 1990. *Scrambling and Barriers.* John Benjamins Publishing Company, Amsterdam/Philadelphia.

Barbara Grosz, Aravind K. Joshi, and Scott Weinstein. 1983. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*, pages 44–50, Cambridge, MA.

Jeannette K. Gundel, Nancy Hedberg, and Ron Zacharski. 1990. Givenness, implicature, and the form of referring expressions in discourse. In *Proceedings of the 16th Annual Meeting of the Berkeley Linguistic Society.*

Jeanette K. Gundel. 1985. Shared knowledge and topicality. *Journal of Pragmatics*, 9:83–107.

Takao Gunji. 1987. *Japanese Phrase Structure Grammar: A Unification-based Approach*, volume 8 of *Studies in Natural Language and Linguistic Theory.* D. Reidel, Dordrecht.

Michael Halliday. 1967. *Intonation and Grammar in British English.* The Hague: Mouton. PhD dissertation.

Jorge Hankamer. 1979. *Deletion in Coordinate Structures.* Garland Publishing Inc., New York.

Irene Heim. 1982. *The Semantics of Definite and Indefinite Noun Phrases.* Ph.D. thesis, University of Massachusetts.

Mark Hepple. 1990. *Word Order, Binding, and Extraction in Categorial Grammar.* Ph.D. thesis, University of Edinburgh.

Caroline Heycock. 1987. The structure of the Japanese causative. Technical Report MS-CIS-87-55, University of Pennsylvania Department of Computer and Information Sciences.

Y. Bar Hillel, C. Gaifman, and E. Shamir. 1960. On categorial and phrase structure grammars. *Bulletin of the Research Council of Israel*, 9F.

Jack Hoeksema. 1991. Complex predicates and liberation in Dutch and English. *Linguistics and Philosophy*, 14:661–710.

Beryl Hoffman and Ümit Turan. 1991. Scrambling in Turkish. In *Proceedings of the 15th annual Penn Linguistics Colloquium.*

Beryl Hoffman. 1991. A TAG analysis of scrambling in Turkish. ms.,University of Pennsylvania.

Beryl Hoffman. 1992. A CCG approach to free word order languages. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics, Student Session.*

Beryl Hoffman. 1993. The formal consequence of using variables in CCG categories. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, Student Session.*

Beryl Hoffman. 1994. Marking set-membership in Turkish. In *Proceedings of the 18th annual Penn Linguistics Colloquium.*

Beryl Hoffman. 1995. Integrating free word order syntax and information structure. In *Proceedings of the Seventh Conference of the European Chapter of the Association for Computational Linguistics.*

Beryl Hoffman. to appear 1995. Word order, information structure, and centering in Turkish. In Ellen Prince, Aravind Joshi, and Marilyn Walker, editors, *Centering in Discourse.* Oxford University Press.

John E. Hopcroft and Jeffrey D. Ullman. 1979. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, Reading, Massachusetts.

Julia Horvath. 1985. *FOCUS in the Theory of Grammar and the Syntax of Hungarian.* Foris Publications, Dordrecht, The Netherlands.

Ray Jackendoff. 1972. *Semantic Interpretation and Generative Grammar.* MIT press, cambridge.

Pauline Jacobson. 1990. Raising as function composition. *Linguistics and Philosophy*, 13:423–75.

Mark Johnson. 1988. *Attribute-Value Logic and the Theory of Grammar.* Number 16 in CSLI Lecture Notes.

Aravind K. Joshi, L. S. Levy, and M. Takahashi. 1975. Tree adjunct grammars. *J. Comput. Syst. Sci.*, 10(1):136–63.

Aravind K. Joshi, K. Vijay-Shanker, and David J. Weir. 1991. The convergence of mildly context-sensitive grammatical formalisms. In Peter Sells, Stuart Shieber, and Thomas Wasow, editors, *Foundational issues in natural language processing*, pages 31–81. MIT Press, Cambridge, MA.

Aravind Joshi. 1985. How much context-sensitivity is required to provide reasonable structural descriptions: Tree-adjoing grammars. In D. Dowty, L. Karttunen, and A. Zwicky, editors, *Natural Language Parsing: Psycholinguistic, Computational and Theoretical Perspectives*, pages 206–350. Cambridge U Press, New York.

Megumi Kameyama. 1985. *Zero anaphora: the case of Japanese.* Ph.D. thesis, Stanford University, Linguistics Department.

Lauri Karttunen. 1989. Radical lexicalism. In Mark Baltin and Anthony Kroch, editors, *Alternative Conceptions of Phrase Structure*. The University of Chicago Press.

T. Kasami. 1965. An efficient recognition and syntax algorithm for context-free languages. Scientific Report AFCRL-65-758, Air Force Cambridge Research Laboratory, Cambridge, MA.

Tracy Holloway King. 1993. *Configuring Topic and Focus in Russian*. Ph.D. thesis, Dept. of Linguistics, Stanford University.

Katalin E. Kiss. 1987. *Configurationality in Hungarian*. D.Reidel Publishing Company, Dordrecht.

Laura Knecht. 1986. *Subject and Object in Turkish*. Ph.D. thesis, Massachusetts Institue of Technology.

Jaklin Kornfilt, Susumo Kuno, and Engin Sezer. 1980. A note on crisscrossing double dislocation. In *Harvard Syntax and Semantics*, volume 3, pages 185–242.

Jaklin Kornfilt. 1984. *Case Marking, Agreement, and Empty Categories in Turkish*. Ph.D. thesis, Dept. of Linguistics, Harvard University.

Manfred Krifka. 1992. A compositional semantics for multiple focus constructions. ms.,University of Texas, Austin.

Anthony Kroch and Aravind K. Joshi. 1986. Analyzing extraposition in a tree adjoining grammar. In G. Huck and A. Ojeda, editors, *Discontinuous Constituents*, volume 20 of *Syntax and Semantics*, pages 107–49, New York, NY. Academic Press.

Susumo Kuno. 1973. *The Structure of the Japanese Language*. The MIT Press.

Susumu Kuno. 1976. Subject, theme and speaker's empathy: A reexamination of relativization phenomena. In C. Li, editor, *Subject and Topic*, pages 417–444. Academic Press, New York.

Susumo Kuno. 1980. Discourse deletion. *Harvard Studies in Syntax and Semantics*, 3:1–144.

Murat Kural. 1991. Scrambling and mixed positions in Turkish. In *NELS 22*.

Murat Kural. 1993. V-to-(I-to)-C in Turkish. In *UCLA Occasional Papers in Linguistics*, volume 11.

J. Lambek. 1958. The mathematics of sentence structure. *American Mathematical Monthly*, 65:154–169.

Young-Suk Lee and Michael Niv. 1989. Scrambling and coordination in korean: A ccg analysis. University of Pennsylvania, ms.

Young-Suk Lee and Beatrice Santorini. 1994. Towards resolving webelhuth's paradox: evidence from german and korean. In Henk van Riemsdijk and Norbert Corver, editors, *Studies on scrambling. Movement and non-movement approaches to free word-order phenomena*, Studies in generative grammar 41, pages 257–300. Mouton de Gruyter, Berlin.

Harry Lewis and Christos H. Papdimitriou. 1981. *Elements of the Theory of Computation.* Prentice Hall, New Jersey.

Brian Linson. 1992. A functional approach to right-dislocation. In *Penn Linguistics Colloquim.*

B. MacWhinney and C. Snow. 1985. The child language data exchange system. *Journal of Child Language*, 12:271–296.

Anoop Mahajan. 1990. *The A/A-bar distinction and movement theory.* Ph.D. thesis, M.I.T.

K.P. Mohanan. 1982. Grammatical relations and clause structure in malayalam. In Joan Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 504–589, Cambridge, Mass. MIT Press.

Richard Montague. 1974. The proper treatment of quantification in ordinary english. In *Formal Philosophy: Selected Papers of Richard Montague.* Yale University Press.

Michael Moortgat. 1988. *Categorial Investigations: logical and linguistic aspects of the Lambek Calculus.* Ph.D. thesis, Rijksuniversiteit, Gröningen, Dordrecht.

Kemal Olfazer. 1993. Two level description of Turkish morphology. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics.*

R.J. Parikh. 1966. On context-free languages. *Journal of the Association for Computing Machinery*, 4:570–581.

Barbara Partee and Mats Rooth. 1983. Generalised conjunction and type ambiguity. In Rainer Bauerle, Christoph Schwarze, and Arnin von Stechow, editors, *Meaning, Ues, and Interpretation of Language*, pages 361–83. de Gruyter, New York.

Barbara Partee. 1991. Topic, focus, and quantification. In S. Moore and A. Wyner, editors, *Proceedings of SALT I*, pages 159–197, Cornell.

Barbara Partee. 1993. Quantificational domains and recursive contexts, invited talk at acl.

Stanley Peters and Robert Ritchie. 1973. On the generative power of transformational grammars. *Information Sciences*, 6:49–83.

Carl Pollard and Ivan Sag. 1987. *Information-Based Syntax and Semantics, Volume 1: Fundamentals*. CSLI, Stanford, CA.

Scott Prevost and Mark Steedman. 1993. Generating contextually appropriate intonation. *Proceedings of the Sixth Conference of the European Chapter of the Association for Computational Linguistics*.

Ellen F. Prince. 1981a. Topicalization, focus movement and yiddish movement:a pragmatic differentiation. In D. Alford etal., editor, *Proceedings of the Seventh Annual Meeting of the Berkely Linguistics Society*, pages 249–264. BLS.

Ellen F. Prince. 1981b. Toward a taxonomy of given-new information. In *Radical Pragmatics*, pages 223–255. Academic Press.

Ellen F. Prince. 1986. On the syntactic marking of the presupposed open proposition. *Chicago Linguistic Society*.

Ellen F. Prince. 1992. The ZPG letter: Subjects, definiteness and information status. In S. Thompson and W. Mann, editors, *Discourse description: diverse analyses of a fund raising text*, pages 295–325. John Benjamins B.V.

Geoffrey Pullum. 1982. Free word order and phrase structure rules. In *NELS 12*, pages 209–220.

Owen Rambow and Giorgio Satta. 1992. Formal aspects of non-locality. Presented at the TAG+ Workshop, University of Pennsylvania.

Owen Rambow. 1994a. *Formal and Computational Aspects of Natural Language Syntax*. Ph.D. thesis, Dept. of Computer and Information Sciences, University of Pennsylvania.

Owen Rambow. 1994b. Multiset-valued linear index grammars. In *Proceedings of the 32nd Annual Meeting of Association for Computational Linguistics*.

Michael Reape. 1991. Clause union and word order variation in germanic. In Nerbonne et al, editor, *Proceedings of the Symposium on Discontinuous Constituency*, IT, Tilburg University, Netherlands. Mouron de Gruyter.

Tanya Reinhart. 1981. Pragmatics and linguistics, an analysis of sentence topics. *Philosophica*, 27:53–94.

Michael Rochemont and Peter Culicover. 1990. *English Focus Constructions and the Theory of Grammar*. Cambridge University Press.

Michael Rochemont. 1978. *A Theory of Stylistic Rules in English*. Garland Press, New York.

Mats Rooth. 1985. *Association with Focus*. Ph.D. thesis, University of Massachusetts, Amherst.

John Robert Ross. 1967. *Constraints on variables in syntax*. Ph.D. thesis, MIT.

M. Saito. 1989. Scrambling as semantically vacuous A'-movement. In Mark Baltin and Anthony Kroch, editors, *Alternative Conceptions of Phrase Structure*. The University of Chicago Press.

Elisabeth O. Selkirk. 1984. *Phonology and syntax : the relation between sound and structure*. MIT Press.

Engin Sezer. 1986. The unmarked sentential subject constraint in Turkish. In Dan Slobin and Karl Zimmer, editors, *Studies in Turkish Linguistics*. John Benjamins Publishing Company.

P. Sgall, E. Hajicova, and E. Benesova. 1973. *Topic, focus and generative semantics*. Scriptor Verlag, Kronberg, Germany.

P. Sgall, E. Hajicova, and J. Panevova. 1986. *The meaning of the sentence and its semantic and pragmatic aspects*. Reidel, Dordrecht.

S. Shieber, G. van Noord, R. Moore, and F. Pereira. 1989. A semantic-head-driven generation algorithm for unification based formalisms. In *Proceedings of the 27th Conference of ACL*.

S. M. Shieber. 1985a. *An Introduction to Unification-Based Approaches to Grammar*. Center for the Study of Language and Information, Stanford, CA.

Stuart Shieber. 1985b. Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343.

Stuart B. Shieber. 1994. Restricting the weak generative capacity of Synchronous Tree Adjoining Grammar. *Computational Intelligence*, 10(4):371–385.

Dan I. Slobin and Thomas G. Bever. 1982. Children use canonical sentence schemas: A crosslinguistic study of word order and inflections. *Cognition*, 12:229–265.

Mark Steedman. 1985. Dependencies and coordination in the grammar of Dutch and English. *Language*, 61:523–568.

Mark Steedman. 1987. Combinatory grammars and parasitic gaps. *Natural Language and Linguistic Theory*, 5:403–439.

Mark Steedman. 1989. Constituency and coordination in a combinatory grammar. In Mark Baltin and Anthony Kroch, editors, *Alternative Conceptions of Phrase Structure*. The University of Chicago Press.

Mark Steedman. 1991. Structure and intonation. *Language*, 67:260–296.

Mark Steedman. 1993. 'Verb Raising' without Reanalysis. ms., University of Pennsylvania.

Ralf Steinberger. 1994. Treating free word order in machine translation. In *Proceedings of Coling 1994*, Kyoto, Japan.

Anna Szabolcsi. 1987. Bound variables in syntax: Are there any? In *Proceedings of the 6th Amsterdam Colloquium*.

Sabahat Tura. 1986. Definiteness and referentiality in Turkish non-verbal sentences. In Dan Slobin and Karl Zimmer, editors, *Studies in Turkish Linguistics*, pages 165–94. John Benjamins Publishing Company.

Ümit Turan. 1995. *Null vs. Overt Subjects in Turkish Discourse: A Centering Analysis*. Ph.D. thesis, University of Pennsylvania, Linguistics. Ph.D. dissertation.

Ümit Turan. to appear 1995. The prominence of entities and the forward looking center hierarchy. In Ellen Prince, Aravind Joshi, and Marilyn Walker, editors, *Centering in Discourse*. Oxford University Press.

Hans Uszkoreit. 1986. Categorial unification grammars. In *Proceedings of the 11th International Conference on Computational Linguistics*, pages 187–194, Bonn.

Hans Uszkoreit. 1987. *Word Order and Constituent Structure in German*. CSLI, Stanford, CA.

Enric Vallduví. 1990. *The Informational Component*. Ph.D. thesis, University of Pennsylvania. Published in 1992 in the *Outstanding dissertations in linguistics* series. New York: Garland.

Johan van Benthem. 1988. The lambek calculus. In Richard Oehrle, Emmon Bach, and Deirdre Wheeler, editors, *Categorial Grammars and Natural Language Structures*, pages 35–68. Reidel, Dordrecht.

Gertjan van Noord and Gosse Bouma. 1994. Adjuncts and the processing of lexical rules. In *Proceedings of Coling*.

Gertjan van Noord. 1990. An overview of head-driiven bottom-up generation. In Robert Dale, Chris Mellish, and Michael Zock, editors, *Current Research in Natural Language Generation*, Cognitive Science Series, pages 141–166. Academic Press, New York.

K. Vijay-Shanker and D. J. Weir. 1990. Polynomial parsing of combinatory categorial grammars. In *28th Annual Meeting of Association for Computational Linguistics*, Pittsburgh.

K. Vijay-Shanker and D. J. Weir. 1993. Parsing some constrained grammar formalisms. *Computational Linguistics*, 19(4).

Marilyn Walker, Masayo Iida, and Sharon Cote. 1994. Japanese discourse and the process of centering. *Computational Linguistics*, 20(2).

G. Webelhuth. 1989. *Syntactic Saturation Phenomena and the Modern Germanic Languages.* Ph.D. thesis, University of Massachusetts.

Jürgen Wedekind. 1988. Generation as structure driven derivation. In *Proceedings of the 12th International Conference on Computational Linguistics.*

D. Weir and A.K. Joshi. 1988. Combinatory categorial grammars: Generative power and relationship to linear context-free rewriting systems. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics (ACL)*, Buffalo, NY.

David Weir. 1988. *Characterizing Mildly Context-Sensitive Grammar Formalisms.* Ph.D. thesis, University of Pennsylvania.

John Whitman. 1991. Rightward movement in verb final languages. In *LSA Annual Meeting.*

Kent. Wittenburg. 1986. *Natural language parsing with combinatory categorial grammar in a graph-unification-based formalism.* Ph.D. thesis, University of Texas at Austin, Austin, TX.

Mary McGee Wood. 1993. *Categorial Grammars.* Linguistic Theory Guides. Routledge, London.

D. H. Younger. 1967. Recognition and parsing of context-free languages in time $O(n^3)$. *Information and Control*, 10(2):189–208.

Henk Zeevat, Ewan Klein, and Jo Calder. 1987. An introduction to unification categorial grammar. In N. Haddock et al, editor, *Edinburgh Working Papers in Cognitive Science, 1, Categorial Grammar, Unification Grammar, and Parsing.*

Arnold Zwicky. 1986. Concatenation and liberation. In *Papers from the 22nd Regional Meeting of the Chicago Linguistic Society*, pages 65–74.