1 *Introduction*

1.1 Preliminaries

One of the things which speakers know about their language is that strings of words which make up a sentence may be grouped into significant sub-strings or "phrases." Speakers of English, for example, know that in a sentence such as (1):

(1) A woman with a big hat stepped on the grass

sub-strings like *a woman*, *a big hat*, *the grass*, *with a big hat*, *on the grass*, *a woman with a big hat*, and *stepped on the grass* form phrases, but that sub-strings like *womam with*, *big hat stepped*, and *with a big* do not.

Our knowledge of English also tells us how these phrases are related to each other. We know, for instance, that the phrase *a woman with a big hat* is composed of two smaller phrases, *a woman* and *with a big hat*. In addition, we known that the phrase *a woman* linearly precedes the phrase *with a big hat*.

The collection of statements about the phrase structure of a sentence, a "phrase marker," constitutes the fundamental object in the theory of syntax. In this study I examine the form that these phrase markers take. The general questions which I will be concerned with are the following. What relations must hold among the various phrases in a sentence? To what extent does the linear ordering of phrases reflect the linear ordering of words in the spoken sentence? What predictions does the choice of a particular type of phrase marker make about the kinds of sentences which are possible in natural language?

The answers to these questions are to be taken as claims about the structure of our knowledge of language, that is, about the way in which language is represented in the mind. Since the form of mental representation is presumably determined in large measure by the particular structure of the human brain, these are ultimately claims about the effects of

brain physiology on grammar. Our current understanding of how this works is at a relatively primitive level, so in what follows my attention will necessarily be devoted solely to formal, rather than neurological, representations.

In pursuing this course of inquiry, I will make no systematic attempt to include, or even sample, all of the currently existing varieties of human language; I will, instead, focus on a very small number of them. This will allow me to formulate detailed hypotheses about the organization of the grammars of these languages. Since, as stated above, these are hypotheses about the functioning of the brain, and since this is presumably uniform across the species, it follows that a claim about the fundamental structure of one language will be a claim about the structure of human language in general. Claims that the grammars of two languages are different are only plausible when it can be shown (as it often may) that the child learning the language has access to relevant evidence.

Before proceeding to the core of this study, I present a brief overview of the grammatical framework assumed.

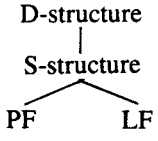
1.2 Background assumptions

1.2.1 The organization of the grammar

I will assume here the version of transformational generative grammar presented in Chomsky (1981, 1982), known as government–binding theory. In this framework, sentences are assigned four distinct levels of representation, listed in (2):

(2) D-structure
S-structure
Phonetic Form (PF)
Logical Form (LF)

These levels are related by mapping operations, organized as in (3):

(3)


As indicated, the relations between D-structure, PF, and LF are mediated by the level S-structure. The general mapping operation is characterized as

"Move α", where α is any category. D-structure is determined by properties of lexical items; PF and LF are the phonetic and logical representations, respectively, of the sentence.

These levels are constrained by several subsystems of the grammar, listed in (4):

(4) a. $\overline{\text{X}}$-theory
b. θ-theory
c. Government theory
d. Case theory
e. Binding theory
f. Bounding theory
g. Control theory

(4) a–e will be directly relevant to much of our subsequent discussion. In the rest of this section, I give a brief introduction to these five subsystems.

1.2.2 $\overline{\text{X}}$-theory

The central idea of $\overline{\text{X}}$-theory is that phrases are projections of lexical categories. That is to say, given a lexical category X, X is dominated by a phrasal node XP (i.e. a phrasal node of the same category type). X is referred to as the "head" of XP, and XP is the "maximal projection" of X. Complements of X are always maximal projections. The linear position of the head relative to its complements must be specified for each language.

1.2.3 θ-theory

θ-theory deals with the arguments taken by predicates. Predicates assign thematic roles (henceforth θ-roles) to their arguments. The number and type of θ-roles assigned by the predicate is determined lexically. θ-roles may be assigned only to a subject or complement of the predicate. These are called "A-positions." Other positions are "$\overline{\text{A}}$-positions." Those A-positions which receive a θ-role are "θ-positions"; those which do not are "$\overline{\theta}$-positions."

The basic principle of θ-theory is the θ-criterion, given in (5):

(5) *θ-criterion*
Each argument receives one and only one θ-role, and each θ-role is assigned to one and only one argument.

The θ-criterion is assumed to hold at LF, but the "Projection Principle," given here in (6), extends it into S-structure and D-structure:

(6) *Projection Principle*
The θ-marking properties of each lexical item must be represented categorically at each syntactic level (i.e. D-structure, S-structure and LF).

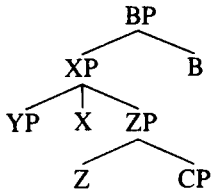
Thus if a verb assigns a θ-role to a complement, that complement will be present and θ-marked at every syntactic level. Moreover, by the θ-criterion, the complement position must contain an argument.

We may now give a more precise definition of D-structure: it is the level where all θ-positions are filled by arguments. It follows from this that in the mapping from D-structure to S-structure (by Move α), arguments may move only to $\bar{\theta}$-positions. If they moved to a θ-position, the θ-criterion would be violated, since a single argument would receive two θ-roles. By the Projection Principle, a (empty) category ("trace") is present in the θ-position even when the argument has moved to a $\bar{\theta}$-position. In order for the θ-criterion to be satisfied, this moved argument must be coindexed with the trace in its D-structure θ-position.

1.2.4 Government theory

The relation "government" plays an important role in many of the subsystems of grammar. In essence, government is the relation between a head and its complements. Consider then a structure such as (7):

(7)



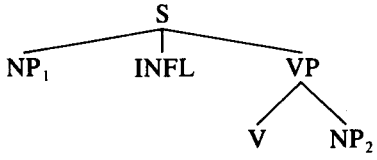
In accordance with $\bar{\bar{X}}$-theory, X is the head of XP, and its complements are maximal projections (i.e. YP and ZP). X "governs" YP and ZP in this configuration. X does not govern BP or CP.

More precisely, we can say that x "governs" y when x c-commands y and when there is no maximal projection which dominates y and does not dominate x. Node x "c-commands" y when the first maximal projection above x also dominates y.

1.2.5 Case theory

Case theory regulates the distribution of phonetically realized NP's via the assignment of abstract case. In a structure like (8), V assigns abstract Case to its NP complement, and INFL, when the clause is tensed, assigns abstract Case to the NP subject:

(8)



Case is assigned under government; potential Case-assigners are V and INFL, as in (8), and P. Notice that in (8), V governs NP_2 and INFL governs NP_1. When the clause is non-finite, INFL does not assign Case. In much of the subsequent discussion, we will assume the presence of INFL without representing it overtly in tree diagrams.

The main regulating mechanism of Case theory is the Case Filter, given in (9):

(9) *Case Filter*
* NP, where NP has a phonetic matrix and is not Case-marked.

This requires that overt NP's appear in positions which are assigned Case. We may assume that the Case Filter applies at S-structure. The relation between S-structure and D-structure now becomes clearer. At D-structure, arguments must appear in θ-marked positions. At S-structure, those arguments which are phonetically realized NP's must appear in Case-marked positions. Move α assures that both of these conditions are satisfied.

The lexical entry of the verb specifies whether or not the verb assigns Case and, if so, which Case(s). In general, the NP must be adjacent to its Case-assigner, thus determining the order of complements in the phrase.

We have seen, then, that a verb may have a set of θ-roles to assign and a set of Cases to assign. These do not always match up, however. An NP position may be θ-marked but not Case-marked, or Case-marked but not θ-marked. For example, verbs such as *arrive* (called "unaccusative" or "ergative" verbs) appear to assign a θ-role to the object, but no Case. The D-structure representation is thus as in (10):

(10) $[_{NP}e]$ $[_{VP}$ has arrived $[_{NP}$ John$]]$

As an S-structure, (10) would violate the Case Filter, so *John* is moved into

subject position, which receives Case from INFL. The S-structure representation is thus as in (11):

(11) $[_{NP}$ John$_i]$ $[_{VP}$ has arrived $[_{NP}t_i]]$

Per the Projection Principle, the θ-marked NP object of *has arrived* is present here. *John* and its trace are coindexed, thus satisfying the θ-criterion.

1.2.6 Binding theory

Binding theory imposes further constraints on the distribution of NP's. Depending on inherent features of the NP, it must be either free or bound within a certain domain. We say that *x* "binds" *y* when *x* is coindexed with *y* and *x* c-commands *y*. "Free" means "not bound." We shall also use terms such as "A-bound," which means "bound by an NP in an A-position."

The relevant domain for binding theory is the "governing category." The governing category for NP y may be defined as the first NP or S which contains *y* and a governor of *y*.

NP's are divided into three classes: anaphors (reflexives, reciprocals, NP-traces), pronouns, and R-expressions (names, *wh*-traces). The conditions imposed on these by binding theory are as follows:

(12) *Binding theory*
A. An anaphor must be A-bound in its governing category.
B. A pronoun must be A-free in its governing category.
C. An R-expression must be A-free.

One further type of NP is PRO, an empty pronominal anaphor. I will assume that PRO must be ungoverned.

1.3 The theory of phrase markers

1.3.1 The definition of Reduced Phrase Markers

The rest of this study will assume the theory of syntax outlined in the previous section. Our fundamental object of inquiry, however, will be not the subsystems of grammar just described, but rather something which underlies all of these subsystems: the nature of phrase structure.

We stated in (2) that sentences are assigned four levels of representation. The grammar specifies what properties these levels must observe, and how they are interrelated. Although the properties required of each syntactic level are different, the general form of sentences at these levels is identical.

Each representation of a sentence is expressed as a phrase marker, on which the various subsystems operate. Our focus here, then, will be on the notion "phrase marker."

I take as my starting point the restrictive theory of phrase markers in Lasnik and Kupin (1977) (henceforth just Lasnik and Kupin), where phrase markers are represented as sets of strings. The vocabulary and conventions used in their system, which I will adopt here, are given in (13):

(13)

N	set of non-terminals
Σ	set of terminals
a, b, c...	single terminals (elements of Σ)
...x, y, z	strings of terminals (elements*)
A, B, C...	single non-terminals (elements of N)
...X, Y, Z	strings of non-terminals (elements of N*)
α, β, γ...	single symbols (elements of $\Sigma \cup$ N)
...χ, ψ, ω	strings of symbols (elements of $(\Sigma \cup \mathrm{N})^*$)
A, B, C...	arbitrary sets

The type of phrase marker allowed by Lasnik and Kupin is called a "Reduced Phrase Marker" (RPM), which we will define below.

RPM's may be thought of as sets consisting of a string of terminals and "monostrings," where these are defined as in (14):

(14) φ is a *monostring* with respect to the sets Σ and N if $\varphi \in \Sigma^* \cdot \mathrm{N} \cdot \Sigma^*$

A monostring thus contains one non-terminal surrounded by strings of terminals. In the definitions to follow, monostrings will be used to identify particular non-terminals. Thus when we say that monostring φ *precedes* monostring ψ, for instance, we mean, in ordinary terms, that the non-terminal in φ precedes the non-terminal in ψ.

Lasnik and Kupin's definition of RPM's makes use of the following predicates (let $\varphi = \mathrm{xAz}$, $\varphi \in P$, $\psi \in P$, where P is an arbitrary set):

(15) y *is a** φ *in* P if $\mathrm{xyz} \in P$

(16) φ *dominates* ψ *in* P if $\psi = \mathrm{x}\chi\mathrm{z}$, $\chi \neq \varnothing$, $\chi \neq \mathrm{A}$

(17) φ *precedes* ψ *in* P if y *is a** φ *in* P and $\psi = \mathrm{xy}\chi$, $\chi \neq \mathrm{z}$

In (15), saying that y *is a** φ, where $\varphi = \mathrm{xAz}$, means that y is an A. Statement (15) ensures that the terminals which surround the non-terminal in a monostring will be exactly those which the non-terminal does not dominate. Thus, comparing a monostring with the string of terminals in a set

tells us which terminals bear the "is a" relationship to the non-terminal of the monostring.

In (16), we see that φ *dominates* ψ if the terminals in φ are a subset of those in ψ. That is to say, if the non-terminal in ψ dominates a subset of the terminals dominated by the non-terminal in φ, then we know that the non-terminal in φ dominates the non-terminal in ψ.

In (17), φ *precedes* ψ if the non-terminal in φ dominates terminals which are to the left of, and distinct from, the terminals dominated by the non-terminal in ψ.

RPM's may now be defined as in (18):

(18) P is an RPM if there exist A and z such that $A \in P$ and $z \in P$; and if $\{\psi, \varphi\} \subseteq P$,
either ψ *dominates* φ *in* P
or φ *dominates* ψ *in* P
or ψ *precedes* φ *in* P
or φ *precedes* ψ *in* P.

There are thus two requirements on RPM's. One is that they must minimally contain a single non-terminal and a string of terminals. The second is that every pair of distinct strings in the RPM must satisfy either *dominates* or *precedes*.

Consider, for example, the set in (19):

(19) {A, Bbc, aC, aDc, abE, abc}

This satisfies the first requirement that the set contain a single non-terminal (A) and a string of terminals (abc). We must now see whether each pair of strings satisfies *precedes* or *dominates*. The pairs are listed in (20):

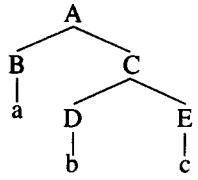
(20)

a.	A	Bbc	i.	Bbc	abc
b.	A	aC	j.	aC	aDc
c.	A	aDc	k.	aC	abE
d.	A	abE	l.	aC	abc
e.	A	abc	m.	aDc	abE
f.	Bbc	aC	n.	aDc	abc
g.	Bbc	aDc	o.	abE	abc
h.	Bbc	abE			

Pairs (a–e), (i–l), (n), and (o) satisfy *dominates*, whereas (f–h) and (m) satisfy *precedes*. This means that A dominates B, C, D, and E, that C domi-

nates D and E, and each non-terminal dominates portions of the terminal string. In addition, B precedes C, D, and E, and D precedes E. As a tree diagram, this may be represented as in (21):

(21)



(21) is "rooted," i.e. there is a single node dominating all others, and it has a terminal string. Moreover, for any pair of nodes in (21), one member of the pair dominates or precedes the other. Intuitively, these descriptions of (21) embody the definition of RPM's in (18).

There are of course many kinds of sets which do not qualify as RPM's under (18). Consider (22), for instance:

(22) {Bbc, aC, aDc, abE, abc}

Set (22) violates the requirement that one of the elements of the set be a single non-terminal. Similarly in (23), there is no terminal string, contrary to the definition in (18):

(23) {A, Bbc, aC}

Sets (22) and (23) thus demonstrate violations of the first requirement on RPM's imposed in (18).

The second requirement, that each pair of elements satisfy *dominates* or *precedes*, is violated in sets such as (24):

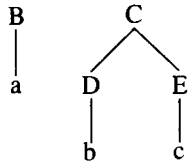
(24) {A, Bbc, aC, aDc, abE, Fc, abc}

The string Fc in (24) does not satisfy *dominates* or *precedes* with the string aC.

RPM's as defined in (18) are only partially equivalent to the syntactic trees customarily employed by linguists.[1] Specifically, the set of RPM's is a proper subset of the set of phrase markers characterizable as trees. Loosely speaking, then, there are trees which are not possible RPM's, but there are no RPM's which are not possible trees.

The sets which we just saw in (22)–(24), for instance, are examples of objects which are neither trees nor RPM's. The diagram associated with (22) is given in (25):

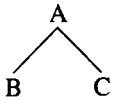
(25)



The structure in (25) is not "rooted" in the sense described above, and hence is not a well-formed tree.

(23) is equivalent to the diagram in (26):

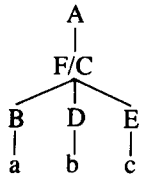
(26)



The fact that (26) contains no terminal nodes disqualifies it as a well-formed tree structure.

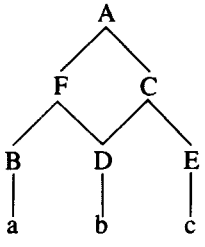
Although (25) and (26) are not trees, they are at least representable as something like trees, i.e. as diagrams showing the dominance and precedence relations among nodes. This is not so in (24), since there the dominance–precedence relations are not fully specified. Node C neither dominates nor precedes F, and *vice versa*. I indicate this as F/C in the following diagram:

(27)



Notice that (27) is not equivalent to (28):

(28)



In (28), F precedes C, contrary to the information given in the set (24). In any event, neither (27) nor (28) are well-formed trees: (27) is out because, as just mentioned, neither dominance nor precedence holds between F and C; (28) is out because the nodes F and C "overlap" at node D.