THREE MODELS FOR THE DESCRIPTION OF LANGUAGE*

Noam Chomsky
Department of Modern Languages and Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts

Abstract

We investigate several conceptions of linguistic structure to determine whether or not they can provide simple and "revealing" grammars that generate all of the sentences of English and only these. We find that no finite-state Markov process that produces symbols with transition from state to state can serve as an English grammar. Furthermore, the particular subclass of such processes that produce n-order statistical approximations to English do not come closer, with increasing n, to matching the output of an English grammar. We formalize the notions of "phrase structure" and show that this gives us a method for describing language which is essentially more powerful, though still representable as a rather elementary type of finite-state process. Nevertheless, it is successful only when limited to a small subset of simple sentences. We study the formal properties of a set of grammatical transformations that carry sentences with phrase structure into new sentences with derived phrase structure, showing that transformational grammars are processes of the same elementary type as phrase-structure grammars; that the grammar of English is materially simplified if phrase structure description is limited to a kernel of simple sentences from which all other sentences are constructed by repeated transformations; and that this view of linguistic structure gives a certain insight into the use and understanding of language.

1. Introduction

There are two central problems in the descriptive study of language. One primary concern of the linguist is to discover simple and "revealing" grammars for natural languages. At the same time, by studying the properties of such successful grammars and clarifying the basic conceptions that underlie them, he hopes to arrive at a general theory of linguistic structure. We shall examine certain features of these related inquiries.

The grammar of a language can be viewed as a theory of the structure of this language. Any scientific theory is based on a certain finite set of observations and, by establishing general laws stated in terms of certain hypothetical constructs, it attempts to account for these observations, to show how they are interrelated, and to predict an indefinite number of new phenomena. A mathematical theory has the additional property that predictions follow rigorously from the body of theory. Similarly, a grammar is based on a finite number of observed sentences (the linguist's corpus) and it "projects" this set to an infinite set of grammatical sentences by establishing general "laws" (grammatical rules) framed in terms of such hypothetical constructs as the particular phonemes, words, phrases, and so on, of the language under analysis. A properly formulated grammar should determine unambiguously the set of grammatical sentences.

General linguistic theory can be viewed as a metatheory which is concerned with the problem of how to choose such a grammar in the case of each particular language on the basis of a finite corpus of sentences. In particular, it will consider and attempt to explicate the relation between the set of grammatical sentences and the set of observed sentences. In other words, linguistic theory attempts to explain the ability of a speaker to produce and understand new sentences, and to reject as ungrammatical other new sequences, on the basis of his limited linguistic experience.

Suppose that for many languages there are certain clear cases of grammatical sentences and certain clear cases of ungrammatical sequences, e.g., (1) and (2), respectively, in English.

(1) John ate a sandwich
(2) Sandwich a ate John.

In this case, we can test the adequacy of a proposed linguistic theory by determining, for each language, whether or not the clear cases are handled properly by the grammars constructed in accordance with this theory. For example, if a large corpus of English does not happen to contain either (1) or (2), we ask whether the grammar that is determined for this corpus will project the corpus to include (1) and exclude (2). Even though such clear cases may provide only a weak test of adequacy for the grammar of a given language taken in isolation, they provide a very strong test for any general linguistic theory and for the set of grammars to which it leads, since we insist that in the case of each language the clear cases be handled properly in a fixed and predetermined manner. We can take certain steps towards the construction of an operational characterization of "grammatical sentence" that will provide us with the clear cases required to set the task of linguistics significantly.

*This work was supported in part by the Army (Signal Corps), the Air Force (Office of Scientific Research, Air Research and Development Command), and the Navy (Office of Naval Research), and in part by a grant from Eastman Kodak Company.

Observe, for example, that (1) will be read by an English speaker with the normal intonation of a sentence of the corpus, while (2) will be read with a falling intonation on each word, as will any sequence of unrelated words. Other distinguishing criteria of the same sort can be described.

Before we can hope to provide a satisfactory account of the general relation between observed sentences and grammatical sentences, we must learn a great deal more about the formal properties of each of these sets. This paper is concerned with the formal structure of the set of grammatical sentences. We shall limit ourselves to English, and shall assume intuitive knowledge of English sentences and nonsentences. We then ask what sort of linguistic theory is required as a basis for an English grammar that will describe the set of English sentences in an interesting and satisfactory manner.

The first step in the linguistic analysis of a language is to provide a finite system of representation for its sentences. We shall assume that this step has been carried out, and we shall deal with languages only in phonemic or alphabetic transcription. By a language, then, we shall mean a set (finite or infinite) of sentences, each of finite length, all constructed from a finite alphabet of symbols. If A is an alphabet, we shall say that anything formed by concatenating the symbols of A is a string in A. By a grammar of the language L we mean a device of some sort that produces all of the strings that are sentences of L and only these.

No matter how we ultimately decide to construct linguistic theory, we shall surely require that the grammar of any language must be finite. It follows that only a countable set of grammars is made available by any linguistic theory; hence that uncountably many languages, in our general sense, are literally not describable in terms of the conception of linguistic structure provided by any particular theory. Given a proposed theory of linguistic structure, then, it is always appropriate to ask the following question:

(3) Are there interesting languages that are simply outside the range of description of the proposed type?

In particular, we shall ask whether English is such a language. If it is, then the proposed conception of linguistic structure must be judged inadequate. If the answer to (3) is negative, we go on to ask such questions as the following:

(4) Can we construct reasonably simple grammars for all interesting languages?

(5) Are such grammars "revealing" in the sense that the syntactic structure that they exhibit can support semantic analysis, can provide insight into the use and understanding of language, etc.?

We shall first examine various conceptions of linguistic structure in terms of the possibility and complexity of description (questions (3), (4)). Then, in § 6, we shall briefly consider the same theories in terms of (5), and shall see that we are independently led to the same conclusions as to relative adequacy for the purposes of linguistics.

2. Finite State Markov Processes.

2.1 The most elementary grammars which, with a finite amount of apparatus, will generate an infinite number of sentences, are those based on a familiar conception of language as a particularly simple type of information source, namely, a finite-state Markov process.[1] Specifically, we define a finite-state grammar G as a system with a finite number of states $S_0, \ldots, S_q$, a set $A = \{a_{ijk} \mid 0 \le i,j \le q;\ 1 \le k \le N_{ij}$ for each $i,j\}$ of transition symbols, and a set $C = \{(S_i, S_j)\}$ of certain pairs of states of G that are said to be connected. As the system moves from state S_i to S_j, it produces a symbol $a_{ijk} \in A$. Suppose that

(6) $S_{\alpha_1}, \ldots, S_{\alpha_m}$

is a sequence of states of G with $\alpha_1 = \alpha_m = 0$, and $(S_{\alpha_i}, S_{\alpha_{i+1}}) \in C$ for each $i < m$. As the system moves from S_{α_i} to $S_{\alpha_{i+1}}$ it produces the symbol

(7) $a_{\alpha_i \alpha_{i+1} k}$

for some $k \le N_{\alpha_i \alpha_{i+1}}$. Using the arch $\frown$ to signify concatenation,[2] we say that the sequence (6) generates all sentences

(8) $a_{\alpha_1 \alpha_2 k_1} \frown a_{\alpha_2 \alpha_3 k_2} \frown \cdots \frown a_{\alpha_{m-1} \alpha_m k_{m-1}}$

for all appropriate choices of k_i (i.e., for $k_i \le N_{\alpha_i \alpha_{i+1}}$). The language L_G containing all and only such sentences is called the language generated by G.

Thus, to produce a sentence of L_G we set the system G in the initial state S_0 and we run through a sequence of connected states, ending again with S_0, and producing one of the associated transition symbols of A with each transition from one state to the next. We say that a language L is a finite-state language if L is the set of sentences generated by some finite-state grammar G.

2.2. Suppose that we take the set A of transition symbols to be the set of English phonemes. We can attempt to construct a finite state grammar G which will generate every string of English phonemes which is a grammatical sentence of English, and only such strings. It is immediately evident that the task of constructing a finite-state grammar for English can be considerably simplified if we take A as the set of English morphemes[3] or words, and construct G so that it will generate exactly the grammatical strings of these units. We can then complete the grammar by giving a finite set of rules that give the phonemic spelling of each word or morpheme in each context in which it occurs. We shall consider briefly the status of such rules in § 4.1 and § 5.3.

Before inquiring directly into the problem of constructing a finite-state grammar for English morpheme or word sequences, let us investigate the absolute limits of the set of finite-state languages. Suppose that A is the alphabet of a language L, that $a_1, \ldots, a_n$ are symbols of this alphabet, and that $S = a_1 \frown \ldots \frown a_n$ is a sentence of L. We say that S has an (i,j)-<u>dependency</u> with respect to L if and only if the following conditions are met:

(9)(i) $1 \leq i < j \leq n$

(ii) there are symbols $b_i, b_j \epsilon A$ with the property that S_1 is not a sentence of L, and S_2 is a sentence of L, where S_1 is formed from S by replacing the i^{th} symbol of S (namely, a_i) by b_i, and S_2 is formed from S_1 by replacing the j^{th} symbol of S_1 (namely, a_j) by b_j.

In other words, S has an (i,j)-dependency with respect to L if replacement of the i^{th} symbol a_i of S by b_i $(b_i \neq a_i)$ requires a corresponding replacement of the j^{th} symbol a_j of S by b_j $(b_j \neq a_j)$ for the resulting string to belong to L.

We say that $D = \left\{ (\alpha_1, \beta_1), \ldots, (\alpha_m, \beta_m) \right\}$ is a <u>dependency set</u> for S in L if and only if the following conditions are met:

(10)(i) For $1 \leq i \leq m$, S has an (α_i, β_i)-dependency with respect to L

(ii) for each i,j, $\alpha_i < \beta_j$

(iii) for each i,j such that $i \neq j$, $\alpha_i \neq \alpha_j$ and $\beta_i \neq \beta_j$.

Thus, in a dependency set for S in L every two dependencies are distinct in both terms and each "determining" element in S precedes all "determined" elements, where we picture a_{α_i} as determining the choice of a_{β_i}.

Evidently, if S has an m-termed dependency set in L, at least 2^m states are necessary in the finite-state grammar that generates the language L.

This observation enables us to state a necessary condition for finite-state languages.

(11) Suppose that L is a finite-state language. Then there is an m such that no sentence S of L has a dependency set of more than m terms in L.

With this condition in mind, we can easily construct many nonfinite-state languages. For example, the languages L_1, L_2, L_3 described in (12) are not describable by any finite-state grammar.

(12)(i) L_1 contains $a \frown b$, $a \frown a \frown b \frown b$, $a \frown a \frown a \frown b \frown b \frown b, \ldots$, and in general, all sentences consisting of n occurrences of a followed by exactly n occurrences of b, and only these;

(ii) L_2 contains $a \frown a$, $b \frown b$, $a \frown b \frown b \frown a$, $b \frown a \frown a \frown b$, $a \frown a \frown b \frown b \frown a \frown a, \ldots$, and in general, all "mirror-image" sentences consisting of a string X followed by X in reverse, and only these;

(iii) L_3 contains $a \frown a$, $b \frown b$, $a \frown b \frown a \frown b$, $b \frown a \frown b \frown a$, $a \frown a \frown b \frown a \frown a \frown b, \ldots$, and in general, all sentences consisting of a string X followed by the identical string X, and only these.

In L_2, for example, for any m we can find a sentence with a dependency set $D_m = \left\{ (1,2m), (2,2m-1), \ldots, (m,m+1) \right\}$.[4]

2.3. Turning now to English, we find that there are infinite sets of sentences that have dependency sets with more than any fixed number of terms. For example, let $S_1, S_2, \ldots$ be declarative sentences. Then the following are all English sentences:

(13)(i) If S_1, then S_2.

(ii) Either S_3, or S_4.

(iii) The man who said that S_5, is arriving today.

These sentences have dependencies between "if"-"then", "either"-"or", "man"-"is". But we can choose S_1, S_3, S_5 which appear between the interdependent words, as (13i), (13ii), or (13iii) themselves. Proceeding to construct sentences in this way we arrive at subparts of English with just the mirror image properties of the languages L_1 and L_2 of (12). Consequently, English fails condition (11). English is not a finite-state language, and we are forced to reject the theory of language under discussion as failing condition (3).

We might avoid this consequence by an arbitrary decree that there is a finite upper limit to sentence length in English. This would serve no useful purpose, however. The point is that there are processes of sentence formation that this elementary model for language is intrinsically incapable of handling. If no finite limit is set for the operation of these processes, we can prove the literal inapplicability of this model. If the processes have a limit, then the construction of a finite-state grammar will not be literally impossible (since a list is a trivial finite-state grammar), but this grammar will be so complex as to be of little use or interest. Below, we shall study a model for grammars that can handle mirror-image languages. The extra power of such a model in the infinite case is reflected in the fact that it is much more useful and revealing if an upper limit is set. In general, the assumption that languages are infinite is made for the purpose of simplifying the description.[5] If a grammar has no recursive steps (closed loops, in the model discussed above) it will be prohibitively complex-- it will, in fact, turn out to be little better than a list of strings or of morpheme class sequences in the case of natural languages. If it does have recursive devices, it will produce infinitely many sentences.

2.4 Although we have found that no finite-state Markov process that produces sentences from left to right can serve as an English grammar, we might inquire into the possibility of constructing a sequence of such devices that, in some nontrivial way, come closer and closer to matching the output of a satisfactory English grammar. Suppose, for example, that for fixed n we construct a finite-state grammar in the following manner: one state of the grammar is associated with each sequence of English words of length n and the probability that the word X will be produced when the system is in the state S_i is equal to the conditional probability of X, given the sequence of n words which defines S_i. The output of such grammar is customarily called an $n+1^{st}$ order approximation to English. Evidently, as n increases, the output of such grammars will come to look more and more like English, since longer and longer sequences have a high probability of being taken directly from the sample of English in which the probabilities were determined. This fact has occasionally led to the suggestion that a theory of linguistic structure might be fashioned on such a model.

Whatever the other interest of statistical approximation in this sense may be, it is clear that it can shed no light on the problems of grammar. There is no general relation between the frequency of a string (or its component parts) and its grammaticalness. We can see this most clearly by considering such strings as

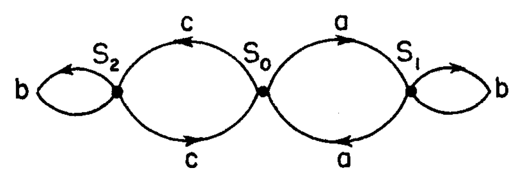
(14) colorless green ideas sleep furiously

which is a grammatical sentence, even though it is fair to assume that no pair of its words may ever have occurred together in the past. Notice that a speaker of English will read (14) with the ordinary intonation pattern of an English sentence, while he will read the equally unfamiliar string

(15) furiously sleep ideas green colorless

with a falling intonation on each word, as in the case of any ungrammatical string. Thus (14) differs from (15) exactly as (1) differs from (2); our tentative operational criterion for grammaticalness supports our intuitive feeling that (14) is a grammatical sentence and that (15) is not. We might state the problem of grammar, in part, as that of explaining and reconstructing the ability of an English speaker to recognize (1), (14), etc., as grammatical, while rejecting (2), (15), etc. But no order of approximation model can distinguish (14) from (15) (or an indefinite number of similar pairs). As n increases, an n^{th} order approximation to English will exclude (as more and more improbable) an ever-increasing number of grammatical sentences, while it still contains vast numbers of completely ungrammatical strings.[6] We are forced to conclude that there is apparently no significant approach to the problems of grammar in this direction.

Notice that although for every n, a process of n-order approximation can be represented as a finite-state Markov process, the converse is not true. For example, consider the three-state process with (S_o,S_1), (S_1,S_1), (S_1,S_o), (S_o,S_2), (S_2,S_2), (S_2,S_o) as its only connected states, and with a, b, a, c, b, c as the respective transition symbols. This process can be represented by the following state diagram:

(16) 

This process can produce the sentences $a^\frown a$, $a^\frown b^\frown a$, $a^\frown b^\frown b^\frown a$, $a^\frown b^\frown b^\frown b^\frown a, \ldots$, $c^\frown c$, $c^\frown b^\frown c$, $c^\frown b^\frown b^\frown c$, $c^\frown b^\frown b^\frown b^\frown c, \ldots$, but not $a^\frown b^\frown b^\frown c$, $c^\frown b^\frown b^\frown a$, etc. The generated language has sentences with dependencies of any finite length.

In § 2.4 we argued that there is no significant correlation between order of approximation and grammaticalness. If we order the strings of a given length in terms of order of approximation to English, we shall find both grammatical and ungrammatical strings scattered throughout the list, from top to bottom. Hence the notion of statistical approximation appears to be irrelevant to grammar. In § 2.3 we pointed out that a much broader class of processes, namely, all finite-state Markov processes that produce transition symbols, does not include an English grammar. That is, if we construct a finite-state grammar that produces only English sentences, we know that it will fail to produce an infinite number of these sentences; in particular, it will fail to produce an infinite number of true sentences, false sentences, reasonable questions that could be intelligibly asked, and the like. Below, we shall investigate a still broader class of processes that might provide us with an English grammar.

3. Phrase Structure.

3.1. Customarily, syntactic description is given in terms of what is called "immediate constituent analysis." In description of this sort the words of a sentence are grouped into phrases, these are grouped into smaller constituent phrases and so on, until the ultimate constituents (generally morphemes[3]) are reached. These phrases are then classified as noun phrases (NP), verb phrases (VP), etc. For example, the sentence (17) might be analyzed as in the accompanying diagram.

(17)

the man	took	the book
NP	Verb	NP
	VP	
Sentence		

Evidently, description of sentences in such terms permits considerable simplification over the word-by-word model, since the composition of a complex class of expressions such as NP can be stated just once in the grammar, and this class can be used as a building block at various points in the construction of sentences. We now ask what form of grammar corresponds to this conception of linguistic structure.

3.2. A phrase-structure grammar is defined by a finite vocabulary (alphabet) V_P, a finite set Σ of initial strings in V_P, and a finite set F of rules of the form: $X \rightarrow Y$, where X and Y are strings in V_P. Each such rule is interpreted as the instruction: rewrite X as Y. For reasons that will appear directly, we require that in each such $[\Sigma, F]$ grammar

$$(18) \quad \Sigma : \Sigma_1, \ldots, \Sigma_n$$

$$F: \; X_1 \rightarrow Y_1 \\ \vdots \\ X_m \rightarrow Y_m$$

Y_i is formed from X_i by the replacement of a single symbol of X_i by some string. Neither the replaced symbol nor the replacing string may be the identity element U of footnote 4.

Given the $[\Sigma, F]$ grammar (18), we say that:

(19)(i) a string β follows from a string α if $\alpha = Z \frown X_i \frown W$ and $\beta = Z \frown Y_i \frown W$, for some $i \le m$;[7]

(ii) a derivation of the string S_t is a sequence $D = (S_1, \ldots, S_t)$ of strings, where $S_1 \epsilon \Sigma$ and for each $i < t$, S_{i+1} follows from S_i;

(iii) a string S is derivable from (18) if there is a derivation of S in terms of (18);

(iv) a derivation of S_t is terminated if there is no string that follows from S_t;

(v) a string S_t is a terminal string if it is the last line of a terminated derivation.

A derivation is thus roughly analogous to a proof, with Σ taken as the axiom system and F as the rules of inference. We say that L is a derivable language if L is the set of strings that are derivable from some $[\Sigma, F]$ grammar, and we say that L is a terminal language if it is the set of terminal strings from some system $[\Sigma, F]$.

In every interesting case there will be a terminal vocabulary V_T ($V_T \subset V_P$) that exactly characterizes the terminal strings, in the sense that every terminal string is a string in V_T and no symbol of V_T is rewritten in any of the rules of F. In such a case we can interpret the terminal strings as constituting the language under analysis (with V_T as its vocabulary), and the derivations of these strings as providing their phrase structure.

3.3. As a simple example of a system of the form (18), consider the following small part of English grammar:

$$(20) \quad \Sigma : \# \frown \text{Sentence} \frown \#$$

$$\begin{aligned} F: \text{Sentence} &\rightarrow NP \frown VP \\ VP &\rightarrow \text{Verb} \frown NP \\ NP &\rightarrow \text{the} \frown \text{man}, \; \text{the} \frown \text{book} \\ \text{Verb} &\rightarrow \text{took} \end{aligned}$$

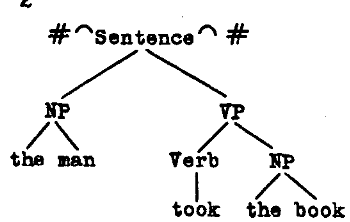
Among the derivations from (20) we have, in particular:

$$(21) \quad D_1 : \begin{array}{l} \# \frown \text{Sentence} \frown \# \\ \# \frown NP \frown VP \frown \# \\ \# \frown NP \frown \text{Verb} \frown NP \frown \# \\ \# \frown \text{the} \frown \text{man} \frown \text{Verb} \frown NP \frown \# \\ \# \frown \text{the} \frown \text{man} \frown \text{Verb} \frown \text{the} \frown \text{book} \frown \# \\ \# \frown \text{the} \frown \text{man} \frown \text{took} \frown \text{the} \frown \text{book} \frown \# \end{array}$$

$$D_2 : \begin{array}{l} \# \frown \text{Sentence} \frown \# \\ \# \frown NP \frown VP \frown \# \\ \# \frown \text{the} \frown \text{man} \frown VP \frown \# \\ \# \frown \text{the} \frown \text{man} \frown \text{Verb} \frown NP \frown \# \\ \# \frown \text{the} \frown \text{man} \frown \text{took} \frown NP \frown \# \\ \# \frown \text{the} \frown \text{man} \frown \text{took} \frown \text{the} \frown \text{book} \frown \# \end{array}$$

These derivations are evidently equivalent; they differ only in the order in which the rules are applied. We can represent this equivalence graphically by constructing diagrams that correspond, in an obvious way, to derivations. Both D_1 and D_2 reduce to the diagram:

(22)

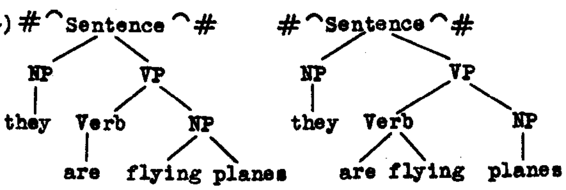


The diagram (22) gives the phrase structure of the terminal sentence "the man took the book," just as in (17). In general, given a derivation D of a string S, we say that a substring s of S is an X if in the diagram corresponding to D, s is traceable back to a single node, and this node is labelled X. Thus given D_1 or D_2, corresponding to (22), we say that "the⌢man" is an NP, "took⌢the⌢book" is a VP, "the⌢book" is an NP, "the⌢man⌢took⌢the⌢book" is a Sentence. "man⌢took," however, is not a phrase of this string at all, since it is not traceable back to any node.

When we attempt to construct the simplest possible $[\Sigma, F]$ grammar for English we find that certain sentences automatically receive non-equivalent derivations. Along with (20), the grammar of English will certainly have to contain such rules as

(23) Verb → are⁀flying
Verb → are
NP → they
NP → planes
NP → flying⁀planes

in order to account for such sentences as "they are flying - a plane" (NP-Verb-NP), "(flying) planes - are - noisy" (NP-Verb-Adjective), etc. But this set of rules provides us with two nonequivalent derivations of the sentence "they are flying planes", reducing to the diagrams:

(24)


Hence this sentence will have two phrase structures assigned to it; it can be analyzed as "they - are - flying planes" or "they - are flying - planes." And in fact, this sentence is ambiguous in just this way; we can understand it as meaning that "those specks on the horizon - are - flying planes" or "those pilots - are flying - planes." When the simplest grammar automatically provides nonequivalent derivations for some sentence, we say that we have a case of <u>constructional</u> <u>homonymity</u>, and we can suggest this formal property as an explanation for the semantic ambiguity of the sentence in question. In §1 we posed the requirement that grammars offer insight into the use and understanding of language (cf.(5)). One way to test the adequacy of a grammar is by determining whether or not the cases of constructional homonymity are actually cases of semantic ambiguity, as in (24) We return to this important problem in § 6.

In (20)-(24) the element # indicated sentence (later, word) boundary. It can be taken as an element of the terminal vocabulary V_T discussed in the final paragraph of § 3.2.

3.4. These segments of English grammar are much oversimplified in several respects. For one thing, each rule of (20) and (23) has only a single symbol on the left, although we placed no such limitation on $[\Sigma, F]$ grammars in § 3.2. A rule of the form

(25) $Z⁀X⁀W \to Z⁀Y⁀W$

indicates that X can be rewritten as Y only in the context Z--W. It can easily be shown that the grammar will be much simplified if we permit such rules. In § 3.2 we required that in such a rule as (25), X must be a single symbol. This ensures that a phrase-structure diagram will be constructible from any derivation. The grammar can also be simplified very greatly if we order the rules and require that they be applied in sequence (beginning again with the first rule after applying the final rule of the sequence), and if we distinguish between <u>obligatory</u> rules which must be applied when we reach them in the sequence and <u>optional</u> rules which may or may not be applied. These revisions do not modify the generative power of the grammar, although they lead to considerable simplification.

It seems reasonable to require for significance some guarantee that the grammar will actually generate a large number of sentences in a limited amount of time; more specifically, that it be impossible to run through the sequence of rules vacuously (applying no rule) unless the last line of the derivation under construction is a terminal string. We can meet this requirement by posing certain conditions on the occurrence of obligatory rules in the sequence of rules. We define a <u>proper grammar</u> as a system $[\Sigma, Q]$, where Σ is a set of initial strings and Q a sequence of rules $X_i \to Y_i$ as in (18), with the additional condition that for each i there must be at least one j such that $X_i = X_j$ and $X_j \to Y_j$ is an obligatory rule. Thus, each left-hand term of the rules of (18) must appear in at least one obligatory rule. This is the weakest simple condition that guarantees that a nonterminated derivation must advance at least one step every time we run through the rules. It provides that if X_i can be rewritten as one of $Y_{i_1}, \ldots, Y_{i_k}$ then at least one of these rewritings must take place. However, proper grammars are essentially different from $[\Sigma, F]$ grammars. Let D(G) be the set of derivations producible from a phrase structure grammar G, whether proper or not. Let $D_F = \{D(G) \mid G \text{ a } [\Sigma, F] \text{ grammar}\}$ and $D_Q = \{D(G) \mid G \text{ a proper grammar}\}$. Then

(26). D_F and D_Q are incomparable; i.e., $D_F \not\subset D_Q$ and $D_Q \not\subset D_F$.

That is, there are systems of phrase structure that can be described by $[\Sigma, F]$ grammars but not by proper grammars, and others that can be described by proper grammars but not by $[\Sigma, F]$ grammars.

3.5. We have defined three types of language: finite-state languages (in §2.1), derivable and terminal languages (in §3.2). These are related in the following way:

(27)(i) every finite-state language is a terminal language, but not conversely;
(ii) every derivable language is a terminal language, but not conversely;
(iii) there are derivable, nonfinite-state languages and finite-state, nonderivable languages.

Suppose that L_G is a finite-state language with the finite-state grammar G as in § 2.1. We construct a [Σ ,F] grammar in the following manner: $\Sigma = \{S_o\}$; F contains a rule of the form (28i) for each i,j,k such that $(S_i, S_j) \epsilon C$, $j \neq 0$, and $k \leq N_{ij}$; F contains a rule of the form (28ii) for each i,k such that $(S_i, S_o) \epsilon C$ and $k \leq N_{io}$.

$$(28)(i) \quad S_i \rightarrow a_{ijk} \frown S_j$$
$$(ii) \quad S_i \rightarrow a_{iok}$$

Clearly, the terminal language from this [Σ ,F] grammar will be exactly L_G, establishing the first part of (27i).

In § 2.2 we found that L_1, L_2 and L_3 of (12) were not finite-state languages. L_1 and L_2, however, are terminal languages. For L_1, e.g., we have the [Σ ,F] grammar

$$(29) \quad \begin{array}{ll} \Sigma: & Z \\ F: & Z \rightarrow a \frown b \\ & Z \rightarrow a \frown Z \frown b \end{array}$$

This establishes (27i).

Suppose that L_4 is a derivable language with the vocabulary $V_P = \{a_1, \ldots, a_n\}$. Suppose that we add to the grammar of L_4 a finite set of rules $a_i \rightarrow b_i$, where the b_i's are not in V_P and are all distinct. Then this new grammar gives a terminal language which is simply a notational variant of L_4. Thus every derivable language is also terminal.

As an example of a terminal, nonderivable language consider the language L_5 containing just the strings

$$(30) \quad a\frown b,\ c\frown a\frown b\frown d,\ c\frown c\frown a\frown b\frown d\frown d,\ c\frown c\frown c\frown a\frown b\frown d\frown d\frown d, \ldots$$

An infinite derivable language must contain an infinite set of strings that can be arranged in a sequence $S_1, S_2, \ldots$ in such a way that for some rule $X \rightarrow Y$, S_i follows from S_{i-1} by application of this rule, for each $i > 1$. And Y in this rule must be formed from X by replacement of a single symbol of X by a string (cf. (18)). This is evidently impossible in the case of L_5. This language is, however, the terminal language given by the following grammar:

$$(31) \quad \begin{array}{ll} \Sigma: & Z \\ F: & Z \rightarrow a \frown b \\ & Z \rightarrow c \frown Z \frown d \end{array}$$

An example of a finite-state, nonderivable language is the language L_6 containing all and only the strings consisting of 2n or 3n occurrences of a, for n=1,2,.... Language L_1 of (12) is a derivable, nonfinite-state language, with the initial string $a\frown b$ and the rule: $a\frown b \rightarrow a\frown a\frown b\frown b$.

The major import of Theorem (27) is that description in terms of phrase structure is essentially more powerful (not just simpler) than description in terms of the finite-state grammars that produce sentences from left to right. In § 2.3 we found that English is literally beyond the bounds of these grammars because of mirror-image properties that it shares with L_1 and L_2 of (12). We have just seen, however, that L_1 is a terminal language and the same is true of L_2. Hence, the considerations that led us to reject the finite-state model do not similarly lead us to reject the more powerful phrase-structure model.

Note that the latter is more abstract than the finite-state model in the sense that symbols that are not included in the vocabulary of a language enter into the description of this language. In the terms of § 3.2, V_P properly includes V_T. Thus in the case of (29), we describe L_1 in terms of an element Z which is not in L_1; and in the case of (20)-(24), we introduce such symbols as Sentence, NP, VP, etc., which are not words of English, into the description of English structure.

3.6. We can interpret a [Σ ,F] grammar of the form (18) as a rather elementary finite-state process in the following way. Consider a system that has a finite number of states $S_o, \ldots, S_q$. When in state S_o, it can produce any of the strings of Σ , thereby moving into a new state. Its state at any point is determined by the subset of elements of $X_1, \ldots, X_m$ contained as substrings in the last produced string, and it moves to a new state by applying one of the rules to this string, thus producing a new string. The system returns to state S_o with the production of a terminal string. This system thus produces derivations, in the sense of § 3.2. The process is determined at any point by its present state and by the last string that has been produced, and there is a finite upper bound on the amount of inspection of this string that is necessary before the process can continue, producing a new string that differs in one of a finite number of ways from its last output.

It is not difficult to construct languages that are beyond the range of description of [Σ ,F] grammars. In fact, the language L_3 of (12iii) is evidently not a terminal language. I do not know whether English is actually a terminal language or whether there are other actual languages that are literally beyond the bounds of phrase structure description. Hence I see no way to disqualify this theory of linguistic structure on the basis of consideration (3). When we turn to the question of the complexity of description (cf. (4)), however, we find that there are ample grounds for the conclusion that this theory of linguistic structure is fundamentally inadequate. We shall now investigate a few of the problems that arise when we attempt to extend (20) to a full-scale grammar of English.

4. Inadequacies of Phrase-Structure Grammar

4.1. In (20) we considered only one way of developing the element Verb, namely, as "took". But even with the verb stem fixed there are a great many other forms that could appear in the context "the man -- the book," e.g., "takes," "has taken," "has been taking," "is taking," "has been taken," "will be taking," and so on. A direct description of this set of elements would be fairly complex, because of the heavy dependencies among them (e.g., "has taken" but not "has taking," "is being taken" but not "is being taking," etc.). We can, in fact, give a very simple analysis of "Verb" as a sequence of independent elements, but only by selecting as elements certain discontinuous strings. For example, in the phrase "has been taking" we can separate out the discontinuous elements "has..en," "be..ing," and "take", and we can then say that these elements combine freely. Following this course systematically, we replace the last rule in (20) by

(32) (i) $\text{Verb} \rightarrow \text{Auxiliary}^\frown \text{V}$
(ii) $\text{V} \rightarrow \text{take, eat,} \ldots$
(iii) $\text{Auxiliary} \rightarrow \text{C(M)(have}^\frown\text{en) (be}^\frown\text{ing) (be}^\frown\text{en)}$
(iv) $\text{M} \rightarrow \text{will, can, shall, may, must}$
(v) $\text{C} \rightarrow \text{past, present}$

The notations in (32iii) are to be interpreted as follows: in developing "Auxiliary" in a derivation we must choose the unparenthesized element C, and we may choose zero or more of the parenthesized elements, in the order given. Thus, in continuing the derivation D_1 of (21) below line five, we might proceed as follows:

(33) $\#^\frown\text{the}^\frown\text{man}^\frown\text{Verb}^\frown\text{the}^\frown\text{book}^\frown\#$
[from D_1 of (21)]
$\#^\frown\text{the}^\frown\text{man}^\frown\text{Auxiliary}^\frown\text{V}^\frown\text{the}^\frown\text{book}^\frown\#$
[(32i)]
$\#^\frown\text{the}^\frown\text{man}^\frown\text{Auxiliary}^\frown\text{take}^\frown\text{the}^\frown\text{book}^\frown\#$
[(32ii)]
$\#^\frown\text{the}^\frown\text{man}^\frown\text{C}^\frown\text{have}^\frown\text{en}^\frown\text{be}^\frown\text{ing}^\frown\text{take}^\frown\text{the}^\frown\text{book}^\frown\#$
[(32iii), choosing the elements C, $\text{have}^\frown\text{en}$, and $\text{be}^\frown\text{ing}$]
$\#^\frown\text{the}^\frown\text{man}^\frown\text{past}^\frown\text{have}^\frown\text{en}^\frown\text{be}^\frown\text{ing}^\frown\text{take}^\frown\text{the}^\frown\text{book}^\frown\#$
[(32v)]

Suppose that we define the class Af as containing the affixes "en", "ing", and the C's; and the class v as including all V's, M's, "have", and "be." We can then convert the last line of (33) into a properly ordered sequence of morphemes by the following rule:

(34) $\text{Af}^\frown\text{v} \rightarrow \text{v}^\frown\text{Af}^\frown\#$

Applying this rule to each of the three $\text{Af}^\frown\text{v}$ sequences in the last line of (33), we derive

(35) $\#^\frown\text{the}^\frown\text{man}^\frown\text{have}^\frown\text{past}^\frown\#^\frown\text{be}^\frown\text{en}^\frown\#^\frown\text{take}^\frown\text{ing}^\frown\#^\frown\text{the}^\frown\text{book}^\frown\#.$

In the first paragraph of § 2.2 we mentioned that a grammar will contain a set of rules (called morphophonemic rules) which convert strings of morphemes into strings of phonemes. In the morphophonemics of English, we shall have such rules as the following (we use conventional, rather than phonemic orthography):

(36) $\text{have}^\frown\text{past} \rightarrow \text{had}$
$\text{be}^\frown\text{en} \rightarrow \text{been}$
$\text{take}^\frown\text{ing} \rightarrow \text{taking}$
$\text{will}^\frown\text{past} \rightarrow \text{would}$
$\text{can}^\frown\text{past} \rightarrow \text{could}$
$\text{M}^\frown\text{present} \rightarrow \text{M}$
$\text{walk}^\frown\text{past} \rightarrow \text{walked}$
$\text{take}^\frown\text{past} \rightarrow \text{took}$
etc.

Applying the morphophonemic rules to (35) we derive the sentence:

(37) the man had been taking the book.

Similarly, with one major exception to be discussed below (and several minor ones that we shall overlook here), the rules (32), (34) will give all the other forms of the verb in declarative sentences, and only these forms.

This very simple analysis, however, goes beyond the bounds of $[\Sigma, F]$ grammars in several respects. The rule (34), although it is quite simple, cannot be incorporated within a $[\Sigma, F]$ grammar, which has no place for discontinuous elements. Furthermore, to apply the rule (34) to the last line of (33) we must know that "take" is a V, hence, a v. In other words, in order to apply this rule it is necessary to inspect more than just the string to which the rule applies; it is necessary to know some of the constituent structure of this string, or equivalently (cf. § 3.3), to inspect certain earlier lines in its derivation. Since (34) requires knowledge of the 'history of derivation' of a string, it violates the elementary property of $[\Sigma, F]$ grammars discussed in § 3.6.

4.2. The fact that this simple analysis of the verb phrase as a sequence of independently chosen units goes beyond the bounds of $[\Sigma, F]$ grammars, suggests that such grammars are too limited to give a true picture of linguistic structure. Further study of the verb phrase lends additional support to this conclusion. There is one major limitation on the independence of the elements introduced in (32). If we choose an intransitive verb (e.g., "come," "occur," etc.) as V in (32), we cannot select $\text{be}^\frown\text{en}$ as an auxiliary. We cannot have such phrases as "John has been come," "John is occurred," and the like. Furthermore, the element $\text{be}^\frown\text{en}$ cannot be chosen independently of the context of the phrase "Verb." If we have the element "Verb" in the context "the man -- the food," we are constrained not to select be⁀en in applying (32), although we are free to choose any other element of (32). That is, we can have "the man is eating the food," "the man would have been eating the food," etc., but not "the man is eaten the food," "the man would have been eaten the food," etc. On the other hand, if the context of the phrase "Verb" is, e.g., "the food -- by the man," we are required to select be⁀en. We can have "the food is eaten by the man," but not "the food is eating by the man," etc. In short, we find that the element be⁀en enters into a detailed network of restrictions which distinguish it from all the other elements introduced in the analysis of "Verb" in (32). This complex and unique behavior of be⁀en suggests that it would be desirable to exclude it from (32) and to introduce passives into the grammar in some other way.

There is, in fact, a very simple way to incorporate sentences with be⁀en (i.e., passives) into the grammar. Notice that for every active sentence such as "the man ate the food" we have a corresponding passive "the food was eaten by the man" and conversely. Suppose then that we drop the element be⁀en from (32iii), and then add to the grammar the following rule:

(38) If S is a sentence of the form NP_1-Auxiliary-V-NP_2, then the corresponding string of the form NP_2-Auxiliary⁀be⁀en-V-by⁀NP_1 is also a sentence.

For example, if "the man - past - eat the food" (NP_1-Auxiliary-V-NP_2) is a sentence, then "the food - past be en - eat - by the man" (NP_2-Auxiliary⁀be⁀en-V-by⁀NP_1) is also a sentence. Rules (34) and (36) would convert the first of these into "the man ate the food" and the second into "the food was eaten by the man."

The advantages of this analysis of passives are unmistakable. Since the element be⁀en has been dropped from (32) it is no longer necessary to qualify (32) with the complex of restrictions discussed above. The fact that be⁀en can occur only with transitive verbs, that it is excluded in the context "the man -- the food" and that it is required in the context "the food -- by the man," is now, in each case, an automatic consequence of the analysis we have just given.

A rule of the form (38), however, is well beyond the limits of phrase-structure grammars. Like (34), it rearranges the elements of the string to which it applies, and it requires considerable information about the constituent structure of this string. When we carry the detailed study of English syntax further, we find that there are many other cases in which the grammar can be simplified if the $[\Sigma, F]$ system is supplemented by rules of the same general form as (38). Let us call each such rule a grammatical transformation. As our third model for the description of linguistic structure, we now consider briefly the formal properties of a transformational grammar that can be adjoined to the $[\Sigma, F]$ grammar of phrase structure.[8]

5. Transformational Grammar.

5.1. Each grammatical transformation T will essentially be a rule that converts every sentence with a given constituent structure into a new sentence with derived constituent structure. The transform and its derived structure must be related in a fixed and constant way to the structure of the transformed string, for each T. We can characterize T by stating, in structural terms, the domain of strings to which it applies and the change that it effects on any such string.

Let us suppose in the following discussion that we have a $[\Sigma, F]$ grammar with a vocabulary V_P and a terminal vocabulary $V_T \subset V_P$, as in § 3.2.

In § 3.3 we showed that a $[\Sigma, F]$ grammar permits the derivation of terminal strings, and we pointed out that in general a given terminal string will have several equivalent derivations. Two derivations were said to be equivalent if they reduce to the same diagram of the form (22), etc.[9] Suppose that $D_1, \ldots, D_n$ constitute a maximal set of equivalent derivations of a terminal string S. Then we define a phrase marker of S as the set of strings that occur as lines in the derivations $D_1, \ldots, D_n$. A string will have more than one phrase marker if and only if it has nonequivalent derivations (cf. (24)).

Suppose that K is a phrase marker of S. We say that

(39) (S,K) is analyzable into $(X_1, \ldots, X_n)$ if and only if there are strings $s_1, \ldots, s_n$ such that

(i) $S = s_1 ⁀ \ldots ⁀ s_n$

(ii) for each $i \leq n$, K contains the string $s_1 ⁀ \ldots ⁀ s_{i-1} ⁀ X_i ⁀ s_{i+1} ⁀ \ldots ⁀ s_n$

(40) In this case, s_i is an X_i in S with respect to K.[10]

The relation defined in (40) is exactly the relation "is a" defined in § 3.3; i.e., s_i is an X_i in the sense of (40) if and only if s_i is a substring of S which is traceable back to a single node of the diagram of the form (22), etc., and this node is labelled X_i.

The notion of analyzability defined above allows us to specify precisely the domain of application of any transformation. We associate with each transformation a restricting class R defined as follows:

(41) R is a restricting class if and only if for some r,m, R is the set of sequences:

$$X_1^1, \ldots, X_r^1$$
$$\vdots$$
$$X_1^m, \ldots, X_r^m$$

where X_i^j is a string in the vocabulary V_P, for each i,j. We then say that a string S with the phrase marker K belongs to the domain of the transformation

T if the restricting class R associated with T contains a sequence $(X_1^j, .., X_r^j)$ into which (S,K) is analyzable. The domain of a transformation is thus a set of ordered pairs (S,K) of a string S and a phrase marker K of S. A transformation may be applicable to S with one phrase marker, but not with a second phrase marker, in the case of a string S with ambiguous constituent structure.

In particular, the passive transformation described in (38) has associated with it a restricting class R_p containing just one sequence:

$$(42)\quad R_p = \left\{ (\text{NP, Auxiliary, V, NP}) \right\}.$$

This transformation can thus be applied to any string that is analyzable into an NP followed by an Auxiliary followed by a V followed by an NP. For example, it can be applied to the string (43) analyzed into substrings $s_1, .., s_4$ in accordance with the dashes.

(43) the man – past – eat – the food.

5.2. In this way, we can describe in structural terms the set of strings (with phrase markers) to which any transformation applies. We must now specify the structural change that a transformation effects on any string in its domain. An <u>elementary transformation</u> t is defined by the following property:

(44) for each pair of integers n,r $(n \leq r)$, there is a unique sequence of integers $(a_o, a_1, ..., a_k)$ and a unique sequence of strings in V_P $(Z_1, .., Z_{k+1})$ such that (i) $a_o = 0$; $k \geq 0$; $1 \leq a_j \leq r$ for $1 \leq j \leq k$; $Y_o = U$[11]

(ii) for each $Y_1, .., Y_r$,

$t(Y_1, .., Y_n; Y_n, .., Y_r) = Y_{a_o}{}^\frown Z_1{}^\frown Y_{a_1}{}^\frown Z_2{}^\frown Y_{a_2}{}^\frown .. {}^\frown Y_{a_k}{}^\frown Z_{k+1}$:

Thus t can be understood as converting the occurrence of Y_n in the context

$$(45)\quad Y_1{}^\frown .. {}^\frown Y_{n-1}{}^\frown - - {}^\frown Y_{n+1}{}^\frown .. {}^\frown Y_r$$

into a certain string $Y_{a_o}{}^\frown Z_1{}^\frown .. {}^\frown Y_{a_k}{}^\frown Z_{k+1}$ which is unique, given the sequence of terms $(Y_1, .., Y_r)$ into which $Y_1{}^\frown .. {}^\frown Y_r$ is subdivided. t carries the string $Y_1 - .. - Y_r$ into a new string $W_1 - .. - W_r$ which is related in a fixed way to $Y_1 - .. - Y_r$. More precisely, we associate with t the derived transformation t*:

(46) t* is the <u>derived</u> <u>transformation</u> of t if and only if for all $Y_1, .., Y_r$, $t^*(Y_1, .., Y_r) = W_1{}^\frown .. {}^\frown W_r$, where $W_n = t(Y_1, .., Y_n; Y_n, .., Y_r)$ for each $n \leq r$.

We now associate with each transformation T an elementary transformation t. For example, with the passive transformation (38) we associate the elementary transformation t_p defined as follows:

$$\begin{aligned}
(47)\quad & t_p(Y_1; Y_1, .., Y_4) = Y_4 \\
& t_p(Y_1, Y_2; Y_2, Y_3, Y_4) = Y_2{}^\frown \text{be}{}^\frown \text{en} \\
& t_p(Y_1, Y_2, Y_3; Y_3, Y_4) = Y_3 \\
& t_p(Y_1, .., Y_4; Y_4) = \text{by}{}^\frown Y_1 \\
& t_p(Y_1, .., Y_n; Y_n, .., Y_r) = Y_n \text{ for all } n \leq r \neq 4.
\end{aligned}$$

The derived transformation t_p^* thus has the following effect:

(48)(i) $t_p^*(Y_1, .., Y_4) = Y_1 - Y_2{}^\frown \text{be}{}^\frown \text{en} - Y_3 - \text{by}{}^\frown Y_1$

(ii) t_p^*(the⁀man, past, eat, the⁀food) = the⁀food – past⁀be⁀en – eat – by⁀the⁀man.

The rules (34),(36) carry the right-hand side of (48ii) into "the food was eaten by the man," just as they carry (43) into the corresponding active "the man ate the food."

The pair (R_p, t_p) as in (42),(47) completely characterizes the passive transformation as described in (38). R_p tells us to which strings this transformation applies (given the phrase markers of these strings) and how to subdivide these strings in order to apply the transformation, and t_p tells us what structural change to effect on the subdivided string.

A grammatical transformation is specified completely by a restricting class R and an elementary transformation t, each of which is finitely characterizable, as in the case of the passive. It is not difficult to define rigorously the manner of this specification, along the lines sketched above. To complete the development of transformational grammar it is necessary to show how a transformation automatically assigns a derived phrase marker to each transform and to generalize to transformations on sets of strings. (These and related topics are treated in reference [3].) A transformation will then carry a string S with a phrase marker K (or a set of such pairs) into a string S' with a derived phrase marker K'.

5.3. From these considerations we are led to a picture of grammars as possessing a tripartite structure. Corresponding to the phrase structure analysis we have a sequence of rules of the form X→Y, e.g., (20), (23), (32). Following this we have a sequence of transformational rules such as (34) and (38). Finally, we have a sequence of morphophonemic rules such as (36), again of the form X→Y. To generate a sentence from such a grammar we construct an extended derivation beginning with an initial string of the phrase structure grammar, e.g., #⁀Sentence⁀#, as in (20). We then run through the rules of phrase structure, producing a terminal string. We then apply certain transformations, giving a string of morphemes in the correct order, perhaps quite a different string from the original terminal string. Application of the morphophonemic rules converts this into a string of phonemes. We might run through the phrase structure grammar several times and then apply a generalized transformation to the resulting set of terminal strings.

In § 3.4 we noted that it is advantageous to order the rules of phrase structure into a sequence, and to distinguish obligatory from optional rules. The same is true of the transformational part of the grammar. In § 4 we discussed the transformation (34), which converts a sequence affix-verb into the sequence verb-affix, and the passive transformation (38). Notice that (34) must be applied in every extended derivation, or the result will not be a grammatical sentence. Rule (34), then, is an obligatory transformation. The passive transformation, however, may or may not be applied; either way we have a sentence. The passive is thus an optional transformation. This distinction between optional and obligatory transformations leads us to distinguish between two classes of sentences of the language. We have, on the one hand, a kernel of basic sentences that are derived from the terminal strings of the phrase-structure grammar by application of only obligatory transformations. We then have a set of derived sentences that are generated by applying optional transformations to the strings underlying kernel sentences.

When we actually carry out a detailed study of English structure, we find that the grammar can be greatly simplified if we limit the kernel to a very small set of simple, active, declarative sentences (in fact, probably a finite set) such as "the man ate the food," etc. We then derive questions, passives, sentences with conjunction, sentences with compound noun phrases (e.g., "proving that theorem was difficult," with the NP "proving that theorem"),[12] etc., by transformations. Since the result of a transformation is a sentence with derived constituent structure, transformations can be compounded, and we can form questions from passives (e.g., "was the food eaten by the man"), etc. The actual sentences of real life are usually not kernel sentences, but rather complicated transforms of these. We find, however, that the transformations are, by and large, meaning-preserving, so that we can view the kernel sentences underlying a given sentence as being, in some sense, the elementary "content elements" in terms of which the actual transform is "understood." We discuss this problem briefly in § 6, more extensively in references [1], [2].

In § 3.6 we pointed out that a grammar of phrase structure is a rather elementary type of finite-state process that is determined at each point by its present state and a bounded amount of its last output. We discovered in § 4 that this limitation is too severe, and that the grammar can be simplified by adding transformational rules that take into account a certain amount of constituent structure (i.e., a certain history of derivation). However, each transformation is still finitely characterizable (cf. §§ 5.1-2), and the finite restricting class (41) associated with a transformation indicates how much information about a string is needed in order to apply this transformation. The grammar can therefore still be regarded as an elementary finite-state process of the type corresponding to phrase structure. There is still a bound, for each grammar, on how much of the past output must be inspected in order for the process of derivation to continue, even though more than just the last output (the last line of the derivation) must be known.

6. Explanatory Power of Linguistic Theories

We have thus far considered the relative adequacy of theories of linguistic structure only in terms of such essentially formal criteria as simplicity. In § 1 we suggested that there are other relevant considerations of adequacy for such theories. We can ask (cf.(5)) whether or not the syntactic structure revealed by these theories provides insight into the use and understanding of language. We can barely touch on this problem here, but even this brief discussion will suggest that this criterion provides the same order of relative adequacy for the three models we have considered.

If the grammar of a language is to provide insight into the way the language is understood, it must be true, in particular, that if a sentence is ambiguous (understood in more than one way), then this sentence is provided with alternative analyses by the grammar. In other words, if a certain sentence S is ambiguous, we can test the adequacy of a given linguistic theory by asking whether or not the simplest grammar constructible in terms of this theory for the language in question automatically provides distinct ways of generating the sentence S. It is instructive to compare the Markov process, phrase-structure, and transformational models in the light of this test.

In § 3.3 we pointed out that the simplest $[\Sigma, F]$ grammar for English happens to provide nonequivalent derivations for the sentence "they are flying planes," which is, in fact, ambiguous. This reasoning does not appear to carry over for finite-state grammars, however. That is, there is no obvious motivation for assigning two different paths to this ambiguous sentence in any finite-state grammar that might be proposed for a part of English. Such examples of constructional homonymity (there are many others) constitute independent evidence for the superiority of the phrase-structure model over finite-state grammars.

Further investigation of English brings to light examples that are not easily explained in terms of phrase structure. Consider the phrase

(49) the shooting of the hunters.

We can understand this phrase with "hunters" as the subject, analogously to (50), or as the object, analogously to (51).

(50) the growling of lions

(51) the raising of flowers.

Phrases (50) and (51), however, are not similarly ambiguous. Yet in terms of phrase structure, each of these phrases is represented as: the - V⁀ing - of⁀NP.

Careful analysis of English shows that we can simplify the grammar if we strike the phrases (49)-(51) out of the kernel and reintroduce them transformationally by a transformation T_1 that carries such sentences as "lions growl" into (50), and a transformation T_2 that carries such sentences as "they raise flowers" into (51). T_1 and T_2 will be similar to the nominalizing transformation described in fn.12, when they are correctly constructed. But both "hunters shoot" and "they shoot the hunters" are kernel sentences; and application of T_1 to the former and T_2 to the latter yields the result (49). Hence (49) has two distinct transformational origins. It is a case of constructional homonymity on the transformational level. The ambiguity of the grammatical relation in (49) is a consequence of the fact that the relation of "shoot" to "hunters" differs in the two underlying kernel sentences. We do not have this ambiguity in the case of (50), (51), since neither "they growl lions" nor "flowers raise" is a grammatical kernel sentence.

There are many other examples of the same general kind (cf. [1],[2]), and to my mind, they provide quite convincing evidence not only for the greater adequacy of the transformational conception of linguistic structure, but also for the view expressed in §5.4 that transformational analysis enables us to reduce partially the problem of explaining how we understand a sentence to that of explaining how we understand a kernel sentence.

In summary, then, we picture a language as having a small, possibly finite kernel of basic sentences with phrase structure in the sense of §3, along with a set of transformations which can be applied to kernel sentences or to earlier transforms to produce new and more complicated sentences from elementary components. We have seen certain indications that this approach may enable us to reduce the immense complexity of actual language to manageable proportions and, in addition, that it may provide considerable insight into the actual use and understanding of language.

Footnotes

1. Cf. [7]. Finite-state grammars can be represented graphically by state diagrams, as in [7], p.15f.

2. See [6], Appendix 2, for an axiomatization of concatenation algebras.

3. By 'morphemes' we refer to the smallest grammatically functioning elements of the language, e.g., "boy", "run", "ing" in "running", "s" in "books", etc.

4. In the case of L_1, b_j of (9ii) can be taken as an identity element U which has the property that for all X, $U^\frown X = X^\frown U = X$. Then D_m will also be a dependency set for a sentence of length 2m in L_1.

5. Note that a grammar must reflect and explain the ability of a speaker to produce and understand new sentences which may be much longer than any he has previously heard.

6. Thus we can always find sequences of n+1 words whose first n words and last n words may occur, but not in the same sentence (e.g. replace "is" by "are" in (13iii), and choose S_5 of any required length).

7. Z or W may be the identity element U (cf. fn.4) in this case. Note that since we limited (18) so as to exclude U from figuring significantly on either the right- or the left-hand side of a rule of F, and since we required that only a single symbol of the left-hand side may be replaced in any rule, it follows that Y_i must be at least as long as X_i. Thus we have a simple decision procedure for derivability and terminality in the sense of (19iii), (19v).

8. See [3] for a detailed development of an algebra of transformations for linguistic description and an account of transformational grammar. For further application of this type of description to linguistic material, see [1], [2], and from a somewhat different point of view, [4].

9. It is not difficult to give a rigorous definition of the equivalence relation in question, though this is fairly tedious.

10. The notion "is a" should actually be relativized further to a given occurrence of s_i in S. We can define an occurrence of s_i in S as an ordered pair (s_i, X), where X is an initial substring of S, and s_i is a final substring of X. Cf. [5], p.297.

11. Where U is the identity, as in fn. 4.

12. Notice that this sentence requires a generalized transformation that operates on a pair of strings with their phrase markers. Thus we have a transformation that converts S_1, S_2 of the forms $NP\text{-}VP_1$, $it\text{-}VP_2$, respectively, into the string: $ing^\frown VP_1 - VP_2$. It converts S_1= "they - prove that theorem", S_2= "it - was difficult" into "ing prove that theorem - was difficult," which by (34) becomes "proving that theorem was difficult." Cf. [1], [3] for details.

Bibliography

[1] Chomsky, N., The Logical Structure of Linguistic Theory (mimeographed).

[2] Chomsky, N., Syntactic Structures, to be published by Mouton & Co., 'S-Gravenhage, Netherlands.

[3] Chomsky, N., Transformational Analysis, Ph. D. Dissertation, University of Pennsylvania, June, 1955.

[4] Harris , Z.S., Discourse Analysis, Language 28,1 (1952).

[5] Quine, W.V., Mathematical Logic, revised edition, Harvard University Press., Cambridge, 1951.

[6] Rosenbloom, P., Elements of Mathematical Logic, Dover, New York, 1950

[7] Shannon & Weaver, The Mathematical Theory of Communication, University of Illinois Press, Urbana, 1949.