QUARTERLY PROGRESS REPORT

No. 60

JANUARY 15, 1961

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
RESEARCH LABORATORY OF ELECTRONICS
CAMBRIDGE, MASSACHUSETTS

XXI. LINGUISTICS*

Prof. R. Jakobson
Prof. A. N. Chomsky
Prof. M. Halle
Dr. R. H. Abernathy
Dr. G. H. Matthews
R. J. Parikh
Carlotta S. Smith

RESEARCH OBJECTIVES

This group sees as its central task the development of a general theory of language. The theory will attempt to integrate all that is known about language and to reveal the lawful interrelations among the structural properties of different languages as well as of the separate aspects of a given language, such as its syntax, morphology, and phonology. The search for linguistic universals and the development of a comprehensive typology of languages are primary research objectives.

Work now in progress deals with specific problems in phonology, morphology, syntax, language learning and language disturbances, linguistic change, semantics, as well as with the logical foundations of the general theory of language. The development of the theory influences the various special studies and, at the same time, is influenced by the results of these studies. Several of the studies are parts of complete linguistic descriptions of particular languages (English, Russian, Siouan) that are now in preparation.

Since many of the problems of language lie in the area in which several disciplines overlap, an adequate and exhaustive treatment of language demands close cooperation of linguistics with other sciences. The inquiry into the structural principles of human language suggests a comparison of these principles with those of other sign systems, which, in turn, leads naturally to the elaboration of a general theory of signs, semiotics. Here linguistics touches upon problems that have been studied by modern logic. Other problems of interest to logicians – and also to mathematicians – are touched upon in the studies devoted to the formal features of a general theory of language. The study of language in its poetic function brings linguistics into contact with the theory and history of literature. The social function of language cannot be properly illuminated without the help of anthropologists and sociologists. The problems that are common to linguistics and the theory of communication, the psychology of language, the acoustics and physiology of speech, and the study of language disturbances are too well known to need further comment here. The exploration of these interdisciplinary problems, a major objective of this group, will be of benefit not only to linguistics; it is certain to provide workers in the other fields with stimulating insight and new methods of attack, as well as to suggest to them new problems for investigation and fruitful reformulations of questions that have been asked for a long time.

R. Jakobson, A. N. Chomsky, M. Halle

A. LANGUAGE-GENERATING DEVICES

In this report we investigate certain properties of context-free (CF or type 2) grammars like Chomsky's (4) and, in particular, questions regarding structure, possible ambiguity, and relationship to finite automata. We present the following results:

(a) The language generated by a context-free grammar is linear in a sense that will be defined precisely.

(b) The requirement of unambiguity – that every sentence has a unique phrase structure – weakens the grammar in the sense that there exists a CF language that cannot be generated unambiguously by a CF grammar.

(c) The result that not every CF language is a finite automaton (FA) language is

*This work was supported in part by National Science Foundation.

improved in the following way. There exists a CF Language L such that for any $L' \subseteq L$, if L' is FA, then we can find an $L'' \subseteq L$ such that L'' is also FA, $L' \subseteq L''$ and L'' contains infinitely many sentences not in L'.

(d) We define a type of grammar that is intermediate between type 1 and type 2 grammars. We show that this type of grammar is essentially stronger than type 2 grammars and has the advantage over type 1 grammars that the phrase structure of a grammatical sentence is unique, once the derivation is given.

1. Preliminaries

Definition 1: By a phrase-structure grammar G we mean a set V of symbols and a set R of rules R_i of the form R_i: $\omega_i \rightarrow \eta_i$, where ω_i and η_i are strings (possibly null) composed of members of V.

Definition 2: We say that the grammar G is of type 1 (a context grammar) if all the rules are of the type: $R_i = \phi_i A_i \psi_i \rightarrow \phi_i \omega_i \psi_i$, where A_i are individual symbols of V, ϕ_i, ω_i, ψ_i are some strings on V, and ω_i are not null. We will also assume that $S = A_i$ for at least one i.

Definition 3: We say that a type 1 grammar G is of type 2 or context-free (CF) if all the ϕ_i, ψ_i as given above are null.

Definition 4: If G is a type 1 grammar, then by V_N we mean the subset $\{A_i\}$ of V. By V_T we mean $V - V_N$ (T means terminal; N means nonterminal).

Convention 1: Hereafter, when talking about type 1 grammars we will use the following convention. Capital Roman letters denote strings on V_N, small Roman letters denote strings on V_T, and Greek letters denote strings on V. Early letters of the alphabet denote individual symbols; late letters denote arbitrary (possibly empty) strings. The boundary symbol # will always belong to V_T (though it does not belong to the Roman alphabet). In discussions of type 2 grammars, this symbol will often be omitted.

Several results from papers by Chomsky and others will be used. While this report does not presuppose acquaintance with these papers, they form the context of this report.

Definition 5: By the set of ϕ-generable strings of a phrase-structure grammar G we mean the smallest set A_ϕ such that

(i) $\phi \in A_\phi$

(ii) if $\phi_1 \omega_i \phi_2 \in A_\phi$ and $\omega_i \rightarrow \eta_i$ is a rule, then $\phi_1 \eta_i \phi_2 \in A_\phi$. If a string ψ belongs to this set we will call it ϕ-generable, and write $\phi \Rightarrow \psi$. The set of generable strings will be the set of #S#-generable strings. A member of this set will be called generable.

Definition 6: The language L generated by G will be the set of those strings on V_T that are generable. Such strings will be referred to as sentences of G or L. Thus a sentence is a generable string which contains no nonterminal symbols. A language will be said to be of type X if it can be generated by a grammar of type X.

Note: Hereafter, all grammars will be type 1 grammars unless otherwise specified. For example, the A_i, ω_i of definition 7 refer to definition 2.

Definition 7: We say that (R_i, j) is a ϕ-derivation of ψ if $\phi = \eta_1\phi_i A_i\psi_i\eta_2$, A_j is the j^{th} symbol of ϕ and $\psi = \eta_1\phi_i\omega_i\psi_i\eta_2$. We will write $\phi \xrightarrow[j]{R_i} \psi$. The members of ω_i in ψ will be said to be the descendents of A_i (here A_i refers not only to the particular member of V but also to the particular occurrence of it in ϕ) with respect to (R_i, j). The members of η_1, ϕ_i, ψ_i, η_2 in ψ will be the descendents of their counterparts in ϕ with respect to (R_i, j).

Definition 8: We say that $D = (Ri_1, j_1), \ldots (Ri_n, j_n)$ is a ϕ-derivation of ψ if there exists a sequence $\phi = \phi_o, \phi_1, \ldots \phi_n = \psi$ such that (Ri_k, j_k) is a ϕ_{k-1} derivation of ϕ_k. We will say that β in ψ is a descendent of α in ϕ with respect to D if there exist $\alpha = \alpha_o, \ldots, \alpha_n = \beta$ such that α_ℓ is a descendent of $\alpha_{\ell-1}$ with respect to (Ri_ℓ, j_ℓ).

Definition 9: Let ϕ be a generable string and let ϕ_1 be a substring of ϕ. Then we will say that ϕ_1 is a phrase of ϕ of type A with respect to D, where $D = (Ri_1, j_1) \ldots (Ri_n, j_n)$ if there exists a corresponding sequence of strings $\#S\# = \phi_o, \phi_1, \ldots \phi_n = \phi$ and an occurrence A_ℓ of A in some ϕ_k such that ϕ_1 is the set of all descendants in ϕ of A_ℓ in ϕ_k with respect to the derivation $D' = (Ri_{k+1}, j_{k+1}), \ldots (Ri_n, j_n)$. We will say "$\phi_1$ is a phrase of ϕ of type A" if there exists a D as above.

Remark: It is easy to see that if two occurrences α and β of symbols in a string ϕ belong to the same phrase, then so do all the occurrences between these two.

Definition 10: We say that a grammar $\underline{G}$ has unambiguous phrase structure if, given two derivations D, D' from S of a member x of L, and a substring x' of x, x' is a phrase of x of type A with respect to D' if and only if x' is a phrase of x of type A with respect to D.

Definition 11: A grammar has unique phrase structure if, given any two phrases in a sentence, either they are disjoint or one is a part of the other.

Theorem 1: If a grammar has unambiguous phrase structure, it has unique phrase structure.

Proof: Let x_1 and x_2 be two subphrases of a sentence (that is, a generable string on V_T) x, and say x_1, x_2 are of types A_1 and A_2 with respect to derivations D, D' of x. Now by unambiguity we may assume that D = D'. If x_1, x_2 are disjoint there is nothing to prove. So pick a in both x_1 and x_2. (Caution: a refers not only to a member of V_T but to a particular occurrence of this member.) Now a is descended from an occurrence α of A_1, and a is descended from an occurrence β of A_2. It is easy to see now that either β is descended from α, or α is descended from β, or $\alpha = \beta$. Hence either $X_1 \subseteq X_2$, or $X_2 \subseteq X_1$, or $X_1 = X_2$. Q. E. D.

Remark: Later we will give an example of a CF language that has no CF grammar with unique phrase structure. It follows that in that case ambiguity is unavoidable.

2. Principal Results

Lemma 1: Every CF language L has a CF grammar $\underline{G}$ such that if $A \in V_n$ and $A \neq S$ then there exist terminal strings x, y, and z, x not null, and at most one of y and z null, such that $A \rightarrow x$ and $A \rightarrow yAz$ are rules of $\underline{G}$. Moreover, if L has a CF grammar with unique phrase structure, then $\underline{G}$ can be assumed to have unique phrase structure.

Proof: If for some nonterminal A there is a terminal x such that $A \Rightarrow x$, then we add the rule $A \rightarrow x$. If there is no such x, we eliminate A and every rule in which A occurs. This does not reduce the generable V_T strings because if any rule with A on the right-hand side is used, then the result cannot lead to a V_T string. However, some nonterminal symbols may become terminal. Then we eliminate these also. This process must have an end because each time we eliminate at least one symbol. Finally, for every $A \in V_N$ except S we have a rule $A \rightarrow x$, with x terminal. Now if there exists an A for which there is no rule $A \rightarrow \phi_1 A \phi_2$ with at least one of ϕ_1, ϕ_2 not null, then we eliminate A and for every rule of the form $B \rightarrow \phi_1 A \phi_2$ and for every rule $A \rightarrow \psi$ we replace these two by the rule $B \rightarrow \phi_1 \psi \phi_2$ (ϕ_1, ϕ_2 may also contain A, in which case we repeat this process with $B \rightarrow \phi_1 \psi \phi_2$), and finally we only have symbols A such that there exist X, ϕ_1, ϕ_2 with $A \rightarrow x$ and $A \rightarrow \phi_1 A \phi_2$ as rules and at least one of ϕ_1, ϕ_2 not null. But then we must also have terminal y, z so that $\phi_1 \Rightarrow y$, $\phi_2 \Rightarrow z$ and not both y and z are null. So we add the rule $A \rightarrow yAz$. This does not change the membership of L. In this entire process, we never added a rule that was not equivalent to a derivation. Hence, no new phrases were created and the new grammar must have unique phrase structure if the old one did. (If x_1, x_2 are phrases by the new grammar, then also by the old grammar they are phrases and then they must be disjoint or one is a part of the other.) Q. E. D.

Lemma 2: If a language L has a CF grammar, then it has a CF grammar in which $A \Rightarrow B$ is never true for A, B in V_N.

Proof: Let us define $A \equiv B$ if $A \Rightarrow B$ and $B \Rightarrow A$. Replacing all the congruence classes by one element each, we get a grammar $\underline{G}'$ in which $A \Rightarrow B$ is a partial ordering of V_N. Now, for every minimal B in this ordering and every rule $A \rightarrow B$, we eliminate the rule $A \rightarrow B$ and replace it by the rules $A \rightarrow \omega$ whenever $B \rightarrow \omega$ is a rule. This would not create any more rules of the form $A \rightarrow C$, since by minimality of B, $B \Rightarrow C$ (and hence $B \rightarrow C$) is impossible. Now we have reduced the number of rules of the form $C \rightarrow D$ without changing L. We continue until all such rules are eliminated. Now, every rule that does not increase length will replace a nonterminal symbol by a terminal one and $A \Rightarrow B$ is impossible. Q. E. D.

Definition 12: Let J denote the non-negative integers. Let J^n denote the direct product of J taken n times. Then J^n is a commutative associative semigroup with identity, under componentwise addition. (For example, in J^2: (2, 3)+(5, 0)=(7, 3), etc.)

We will say that a subset A of J^n is linear if there exist members α, β_1, ... β_m

of J^n such that

$$A = \{x \mid x=a+n_1\beta_1+\ldots+n_m\beta_m, n_i \in J\}$$

We will say that A is semilinear if A is the union of a finite number of linear sets.

Definition 13: Let L be any CF language on the terminal symbols $a_1 \ldots a_n$, #. Define Φ from L into J^n as follows:

$$\Phi(a_1) = (1, \ldots 0, 0, 0)$$

$$\Phi(a_2) = (0, 1, 0, \ldots 0)$$

. . . . . .

$$\Phi(a_n) = (0 \ldots 0, \ldots 1)$$

$$\Phi(\#) = (0 \ldots 0, \ldots 0)$$

$$\Phi(xy) = \Phi(x) + \Phi(y)$$

Then we call $\Phi(x)$ the commutative image of x and $\Phi(L)$ the commutative map of L. [Note that Φ depends on the order of the a_i but we will ignore this fact.]

Theorem 2: Let G be a CF grammar generating the language L. Let $\Phi(L)$ be the commutative map of L. Then $\Phi(L)$ is a semilinear subset S of J^n for the proper n. Moreover, a canonical description of S in the form

$$S = A_1 \cup A_2 \ldots \cup A_m$$

where

$$A_j = \{x \mid x=a_j+n_1\beta_{j1}+n_2\beta_{j2}+\ldots+n_{k_j}\beta_{jk_j}, n_i \in J\}$$

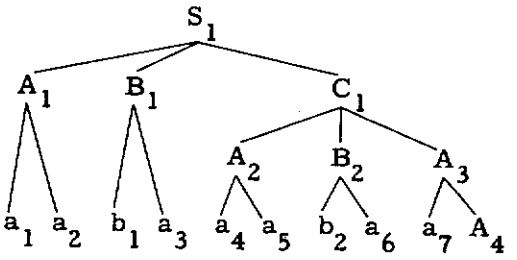
can be found effectively from G.

Conversely, if S is a semilinear subset of J^n, then a CF grammar G to generate L such that $\Phi(L) = S$, can be found effectively from a canonical description of S. (Note: the symbol S is used here both to indicate a subset of J^n and a member of V_N. However, it is always clear from the context which one is meant.)

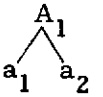
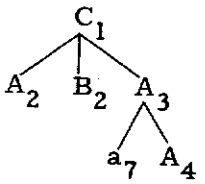
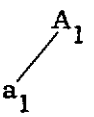
Proof: Let V' be a subset of V. Consider the set L' of all members x of L such that in some derivation D of x, the members of V' are precisely the symbols that are used. It is enough to find a canonical description for L', since L is a finite union of such L'. Obviously, L' is empty unless V' contains S (and #, if used). Since no rule involving some symbol outside V' can be used in such a D, we can assume without loss of generality that V' is V.

At this point, we introduce the notion of a tree by means of an illustration. Suppose that we have the rules $S \rightarrow ABC$, $A \rightarrow aA$, $A \rightarrow aa$, $B \rightarrow ba$, $C \rightarrow ABA$. Then we could have

the derivation $S \to ABC \to aaBC \to aabaC \to aabaABA \to aaabaaaBA \to aabaaabaA \to aabaaabaaA$. This could be written diagrammatically as follows:



(The different occurrences of a, b, c, etc. are numbered for convenience.) The order in which the rules are applied is not preserved but nothing essential is lost. (It is possible to define a tree as an equivalence class of derivations, but in that case intuitively obvious facts would have to be proved. Here, it is enough for us to see that such a formal and more rigorous approach is possible.) We shall illustrate some notions.

SA_1a_1, SB_1 are chains; SA_1b_1 is not; a_1 is descended from A_1 and S but not from B_1; A_4 is descended from A_3, hence A is descended from itself;  is a subtree and so is  but  is not. The string aabaaabaaA is the product of the tree.

Now, for every α in V_N we define two sets S_α and T_α. We say that ϕ is in S_α if (a) ϕ contains α and α is the only nonterminal symbol in ϕ; and (b) there is a tree with α at the vertex such that ϕ is the product of the tree and no symbol occurs more than n times in any chain of the tree, where n is the number of elements in V.

T_α is defined analogously except that condition (a) is replaced by the condition that ϕ be terminal.

We claim that there are only finitely many trees satisfying condition (b), since in any such tree the length of any chain cannot be greater than the square of the number of symbols in V. Hence S_α and T_α are finite and can be found effectively from G.

For each α, let $v_1^\alpha, v_2^\alpha, \ldots, v_r^\alpha$ be the vectors obtained by removing α from a member ϕ of S_α and then taking the image under Φ. (See definition 13.) Let $u_1 \ldots u_k$ be the images under Φ of the members of T_S. Set

$$A_\ell = \left\{ x \,\middle|\, x = u_\ell + n_1 v_1^\alpha + \ldots + n_r v_r^\alpha + n_1' v_1^\beta + \ldots + n_{r'}' v_{r'}^\beta + \ldots ; n_i, n_i' \ldots \in J \right\}$$

Then $\Phi(L') = A_1 \cup A_2 \ldots \cup A_k$.

For, certainly, if some string y is in L' and $a \in V_N$, then a must occur somewhere in a tree for y. Then in the place where a occurs we could imbed (for any v_i^a that we please) a tree with a product string ϕ, $\phi \in S_a$, and $\Phi(\phi - a) = v_i$. Hence $\Phi(y) + v_\ell^i$ is also in $\Phi(L')$.

On the other hand, if a string has a tree with more than n a's in it then, for some β, we can find n + 1 β's in a descending sequence $\beta_1, \ldots, \beta_{n+1}$ such that all of them occur in a chain which, moreover, has the property that there is no chain entirely below β_1 which contains more than n occurrences of any symbol. Now suppose that we replace the tree following β_i by the tree following β_{i+1}. Then we have reduced the product of the entire tree by exactly a member of S_β. The new tree may not contain all the symbols from V. However, since there are only n - 1 symbols in V apart from β, there must be a choice of i, $1 \leq i \leq n$, such that the new tree contains all the members of V if the old one did. We continue this process until we have a tree in which any chain has, at most, n occurrences of any one symbol, and its product must be a member of T_S. This proves the first part of the theorem.

As for the converse, we may assume that S is linear (since CF languages are effectively closed under union) and let

$$S = \{x \mid x = a + n_1\beta_1 + \ldots + n_r\beta_r; n_i \in J\}$$

Then let y, $y_1 \ldots y_r$ be strings whose images under Φ are a, $\beta_1, \ldots, \beta_r$ and consider the rules $S \rightarrow y$, $S \rightarrow Sy_i$, i = 1, ..., r.

It is easy to see that these rules give the desired result. Q. E. D.

Corollary 1: The following questions regarding the language L generated by a CF grammar G are effectively decidable.

(a) Is L empty? (L is empty if and only if $\Phi(L)$ is empty.)

(b) Is L infinite? (L is infinite if and only if $\Phi(L)$ is infinite, if and only if $k_j \neq 0$ for some j in the statement of Theorem 2.)

Corollary 2: Every CF language is equivalent to an FA language modulo permutations. (See definition 14 and Chomsky (3).)

Corollary 3: Let L be any CF language and $a \in V_T$. Define a map Θ from $L \rightarrow J$ by $\Theta(a) = 1$, $\Theta(b) = 0$, $b \neq a$, $\Theta(a\beta) = \Theta(a) + \Theta(b)$. Then there exist integers m, m', $n_1 \ldots n_k$ such that if $n > m$, $n \in \Theta(L)$ if and only if $n \equiv n_i \pmod{m'}$ for some i.

Proof: It is easy to see that $\Theta(L)$ will be a semilinear subset of the integers. Let $A_1, \ldots, A_r, A_{r+1}, \ldots, A_s$ be the linear sets whose union it is. Here we assume that $A_1 \ldots A_r$ are finite and $A_{r+1} \ldots A_s$ have each a smallest vector $\delta_i \neq 0$ such that if $x \in A_i$, then $x + \delta_i \in A_i$. Here δ_i is, of course, an integer. Take m to be bigger than all the elements of the finite sets $A_1 \ldots A_r$, m' to be the product of all the δ_i, and $n_1 \ldots n_k$ the least members of $A_{r+1}, \ldots, A_s$ (where k=s-r).

Theorem 3: There exists a CF language L such that no CF grammar for L has

unique phrase structure.

Proof: We will show, first, that the language

$$L = \{x \mid x=a^n b^m a^{n'} b^m \quad \text{or} \quad x=a^n b^m a^n b^{m'}\}$$

for some n, n', m, m' in J−{0}, is CF.

For, consider the rules

$S \to AB$ $\quad A \to aAa,$ $\quad A \to aBa,$ $\quad B \to b,$ $\quad B \to bB$

$S \to CD$ $\quad C \to bCb,$ $\quad C \to bDb,$ $\quad D \to a,$ $\quad D \to aD$

The terminal descendants of B have the form b^n, $n > 0$. The terminal descendants of D have the form a^n, $n > 0$. Hence the terminal descendants of A must be $a^m b^n a^m$; the terminal descendants of C must be $b^m a^n b^m$.

It is easy to see that these rules generate L.

Suppose that L has a grammar with unique phrase structure. By lemma 2 we may assume that for every A in V_n there exist rules $A \to x$, $A \to yAz$ with x, y, z terminal, x not empty, and, at most, one of y and z not empty. We may also assume that every A in V_N is descended from S because the others cannot contribute to L.

The intuitive idea behind the proof is as follows. L contains precisely the strings of the form $a^i b^j a^k b^\ell$ with either $i = k$ or $j = \ell$, or both. Now the strings $a^i b^j a^i b^\ell$ will have subphrases of the form $a^i b^j a^i$, while the strings $a^i b^j a^k b^j$ will have subphrases of the form $b^j a^k b^j$. Hence the strings $a^i b^j a^i b^j$ must contain both and will therefore have overlapping phrases. This is the essence of the proof. The details follow.

Now we claim that there are only eight types of nonterminal symbols A which can occur in V:

1a. There exist x and y such that $A \to xAy$ is a rule and $x = a^m$, $y = a^{m'}$, and no b's are ever descended from A.

1b. Same as 1a except that there are b's descended from A and $m \neq m'$ in at least one pair x, y. However, there is an integer ℓ_A such that in any string descended from A there are less than ℓ_A b's.

2a. Same as 1a with a and b interchanged.

2b. Same as 1b with a and b interchanged.

3a. Whenever $A \to xAy$ is a rule, $x = y = a^m$ for some m. There are b's descended from A, but the number of b's in a string from A is bounded by b_A.

3b. Whenever $A \to xAy$ is a rule $x = y = a^m$ for some m. There are integers l_A, $F_A = f$ such that some string descended from A has ℓ_A b's; and if xby is a terminal string descended from A with at least ℓ_A b's, then $xbb^f y$ is also descended from A.

4a. Same as 3a with a and b reversed.

4b. Same as 3b with a and b reversed.

Proof of claim: First, it is easy to see that for every A either $A \Rightarrow xAy$ implies

$x = a^m$ $y = a^{m'}$ for some m, $m' \geq 0$, or $A \Rightarrow xAy$ implies $x = b^m$, $y = b^{m'}$ for some m, $m' \geq 0$.

Anything else would contradict one of two requirements:

(a) Every sentence has exactly two groups of a's and two groups of b's.

(b) Either the groups of a's are identical, or else the groups of b's are.

Now, if $A \rightarrow xAy$ with $x = a^m$, $y = a^{m'}$ with $m \neq m'$, then A can only occur in the derivation of a string $a^i b^j a^k b^j$. Now the number of b's generated by A must be fixed. Otherwise we could not have matching of the groups of b's. Hence A is of type 1a or 1b. Now let us assume that $A \rightarrow xAy$ with x and y powers of a, and A is not of type 1a, 1b or 3a. We will show it must belong to type 3b. We already know that if $A \rightarrow xAy$, then $x = y$ must be true.

Consider a string u descended from A which has more b's than the largest number occurring on the right-hand side of any rule. Then at the time in the derivation of u when the first b is generated, there must be a nonterminal symbol B left over. Now that string has the form $a^\ell \omega b \eta B \theta a^\ell$. (The existence of the a^ℓ at the two sides can be assumed because we could always have used the rule $A \rightarrow a^\ell A a^\ell$ before starting.) It is easy to see that if $B \rightarrow xBy$ is a rule, then x and y must be powers of b. Let $xy = b^{f_B}$. Let such an f_B be chosen for each B with a rule $B \rightarrow xBy$ attached to it and x and y powers of b. Now, in the string $a^n \omega b \eta B \theta a^n$ no a's could possibly come from η. Hence if u has the form zbz' we can also get the string $zbb^{f_A}z'$ from A, where f_A is the product of all the f_B taken above. Hence A is of type 3b.

Types 2, 4 are handled in a similar manner. The claim is proved.

Proof of Theorem 3: Let p be a number divisible by all the f_A described in types 3b, 4b. Let $\frac{n}{2}$ be larger than all the ℓ_A described above. Consider the string $x_o = a^{n+p_b n a^{n+p_b n+2p}}$.

Now no derivable string can contain more than two symbols of type 1b or 2b. The string x_o cannot have contained in its derivation any symbols of types 1a, 1b, 3a, 4b. On the other hand, not enough a's could come from type 2b or 4a. Hence there must have been an occurrence of a symbol A of type 3b. If we apply the rule $A \rightarrow xAy$ enough extra times, we can get a string in which the phrase coming from A must be of the form $a^p zbz'a^p$. This can be changed, as before, to $a^p zb.b^p b^p z' a^p$. Thus we get the string $a^{n+2p_b n+2p_a n+2p_b n+2p}$ with the A phrase containing at least $a^p b^{n+2p} a^p$, and bounded on both sides by a's. Similarly, by duality between a and b, there is a phrase containing at least $b^p a^{n+2p} b^p$, and bounded on both sides by b's. But these phrases overlap, and yet one cannot include the other.

Hence G cannot have unique phrase structure. Q. E. D.

Corollary: L is a CF language for which there is no CF grammar with unambiguous phrase structure.

Proof: Follows immediately from Theorem 1.

Definition 14: A finite-state grammar G consists of a finite set $\underline{S}$, (called the internal states of G), a finite set W (called the vocabulary of G), two distinguished elements S_o and S_f of $\underline{S}$, and a subset $\underline{R}$ of $\underline{S} \times \underline{S} \times W'$ (called the rules of G) where $W' = W \cup \{\Lambda\}$ and Λ is the empty string.

Remark: Here we depart somewhat from the 1959 Chomsky definition in that we do not require a symbol to be emitted at every interstate transition. It is not difficult to show, however, that the difference is unimportant and that the same class of languages is generated.

Definition 15: Let G be a finite-state grammar. Then we will say that the sentence x is generated by G if there exists a sequence $(S_o, S_1, x_o), (S_1, S_2, x_2) \ldots (S_n, S_f, x_n)$ of members of $\underline{R}$ such that $x = x_o x_1 \ldots x_n$. The language generated by G is the set of all such sentences x.

Theorem: Every language generated by a finite-state grammar (FA language) is CF.

Proof: Has been given by Chomsky (4).

Theorem 4: There exists a CF language L such that given a grammar G' for an FA language L' with $L' \subseteq L$, we can effectively find a grammar G" for an FA language L" such that $L' \subseteq L'' \subseteq L$ and L" has infinitely many sentences not in L'.

Before we prove this theorem we give two definitions and prove a lemma.

Definition 16: A finite translator T consists of two finite sets V, V' (called the vocabularies of T) a set $\underline{S}$ (called the internal states of T) and a certain subset $\underline{R}$ (called the rules of T) of $V \times \underline{S} \times V'' \times \underline{S} \times \{0, 1\}$. Here $V'' = V' \cup \{\Lambda\}$ and Λ is the empty string. A member S_o of $\underline{S}$ is distinguished and called the initial state of T.

Definition 17: Given a finite translator $T = \{V, V', \underline{S}, \underline{R}\}$ and a sentence $x = x_1 \ldots x_m$ on V, we will say that sentence z is a translation of x by T, if there exists a sequence

$$\langle S_o, y_1, S_1, z_1, i_1 \rangle, \ \langle S_1, y_2, S_2, z_2, i_2 \rangle \ldots \langle S_n, y_{n+1}, S_{n+1}, z_{n+1}, i_{n+1} \rangle$$

of members of $\underline{R}$ such that $y_1 = x_1$. If $y_\ell = x_j$ and $i_\ell = 0$, then $y_{\ell+1} = x_j$ otherwise $y_\ell = x_{j+1}$, $y_{n+1} = x_m$, $i_{n+1} = 1$ and $z = z_1 \ldots z_{n+1}$ (where the z_i will of course, be either Λ or members of V').

Lemma 3: Let L be an FA language on a vocabulary V with grammar $G = \langle V, \underline{S}, \underline{R} \rangle$. Let $T = \langle V, V', \underline{S}_1, \underline{R}_1 \rangle$ be a finite-state translator. Then the set of all translations of members of L by T is an FA language L" on V' and a grammar G" for L" can be found effectively from G and T.

Proof: For the vocabulary of G" we take the set V'. For $\underline{S}''$ we take a set of ordered triples $\langle a, b, c \rangle$, where $a \in \underline{S}$, $b \in \underline{S}_1$, and $c \in V$ or $c = \Lambda$. We define $\underline{R}''$ as follows:

(a) Whenever $\langle S_1, S_2, x \rangle$ is a rule of G, $x \in V$ and $\langle t_1, x, t_2, z, 0 \rangle$ is a rule of T we introduce the rule $\langle \langle t_1, S_1, x \rangle, \langle t_2, S_1, x \rangle, z \rangle$ into $\underline{R}''$.

(b) Whenever $\langle S_1, S_2, x \rangle$ is a rule of G, $x \in V$ and $\langle t_1, x, t_2, z, 1 \rangle$ is a rule of T we

introduce the rules $\langle\langle t_1, S_1, x\rangle, \langle t_2, S_2, y\rangle, z\rangle$ into $\underline{R}''$ for any $y \in V'$ or $y = \Lambda$.

(c) If $\langle S_1, S_2, \Lambda\rangle$ is a rule of G, then for any t_1 we introduce the rules $\langle\langle t_1, S_1, \Lambda\rangle, \langle t_1, S_2, y\rangle, \Lambda\rangle$ for any $y \in V'$ or $y = \Lambda$.

We also introduce two more states I and F to be the initial and final states of G" and the rules $\langle I, \langle t_o, S_o, y\rangle, \Lambda\rangle$, where $y \in V'$ or $y = \Lambda$, and $\langle\langle t, S_f, \Lambda\rangle, F, \Lambda\rangle$, where S_o, S_f are the initial and final states of G, t is any state of T, and t_o is the initial state of T.

Now it is easy to see that G" produces exactly the translations of sentences produced by G. Consider the following cases:

(a) G is in state S_1, moves to state S_2, and produces x. The translator T in state t_1 translates x as z and the rule used is $\langle t_1, x, t_2, z, o\rangle$. Then, correspondingly, G" in state $\langle t_1, S_1, x\rangle$ produces z and moves to state $\langle t_2, S_1, x\rangle$. This continues until we get case (b).

(b) G is in state S_1, moves to state S_2 and produces x. The translator T in state t_1 translates x as z and the rule used is $\langle t_1, x, t_2, z, 1\rangle$. This means, then, that the translator is finished translating x. Then G" in state $\langle t_1, S_1, x\rangle$ produces z and may move to $\langle t_2, S_2, y\rangle$ for <u>any</u> y. Thus it is ready to translate the next symbol that G may produce.

(c) G moves from S_1 to S_2 and produces nothing. Then G" moves from $\langle t_1, S_1, \Lambda\rangle$ to $\langle t_1, S_2, y\rangle$ for any y. The translator is unaffected.

Thus the second and third parts of the states of G" trace out the states of G and symbols produced by G, while the first part traces out the reaction of T.

Proof of Theorem 4: The language $L^\circ = \{a^n b^m a^n \mid n,m \in J\}$ can be easily shown to be CF but not FA. (See Chomsky (4).) Consider the language $L = \{A^n B^m A^n\}$, where each A has the form ce^kc for some $k > 0$. Each B has the form df^kd for some $k > 0$. Consider the translator T defined by $V = \{a, b\}$, $V' = \{c, d, e, f\}$, $\underline{S} = (S_o, S_1, S_2, S_3, S_4)$, and the rules

(α) $(S_o, a, S_1, c, 0)$, $(S_1, a, S_2, e, 0)$, $(S_2, a, S_o, c, 1)$, $(S_1, a, S_1, e, 0)$

(β) $(S_o, b, S_3, d, 0)$, $(S_3, b, S_4, f, 0)$, $(S_4, b, S_o, d, 1)$, $(S_3, b, S_3, f, 0)$

It is easy to see that the language L is the map of L° under T.

On the other hand, define T' by $V = \{c, d, e, f\}$, $V' = \{a, b\}$, $\underline{S} = \{S_o, S_1, S_2\}$, and the rules

$$\langle S_o, c, S_1, a, 1\rangle, \langle S_1, e, S_1, \Lambda, 1\rangle, \langle S_1, c, S_o, \Lambda, 1\rangle$$

$$\langle S_o, d, S_2, b, 1\rangle, \langle S_1, f, S_1, \Lambda, 1\rangle, \langle S_1, d, S_o, \Lambda, 1\rangle$$

Then L° is the map of L under T'.

Now consider any FA language $L' \subseteq L$. Then $T(L') \subseteq T(L) = L^\circ$. But T(L') is FA and L° is not. Hence L° must contain a string x not in T(L'). A grammar for T(L') can

be found effectively by lemma 3, and it is easy to see how a grammar G° for $T(L')\cup\{x\}$ can be found effectively. (L° obviously has a decision procedure for membership. So do FA languages. We take the first x in $L^\circ - T(L')$ and construct the grammar, using x and the grammar for $T(L')$.)

But if $L^{\circ\prime}$ is the language generated by G°, then $L^{\circ\prime}\supset T(L')$. Hence $T'(L^{\circ\prime})\supset T'T(L')\supset L'$ and, in fact, must contain the infinitely many sentences obtained from x which cannot be in L'. Q. E. D.

Definition 18: We say that a type 1 grammar G is of type 1_A if there exists a function f from V into the non-negative integers such that if $\phi\alpha\psi \rightarrow \phi\beta\psi$ is a rule, then $f(\beta) < f(\alpha)$.

Definition 19: We say that a type 1 grammar is of type 1_B if there are no rules of the form $\phi A\psi \rightarrow \phi B\psi$ with $A, B \in V_N$.

Corollary: A type 1_B grammar is of type 1_A.

Proof: Let $f(\alpha) = 1$ if $\alpha \in V_N$.

$f(\alpha) = 0$ if $\alpha \in V_T$. Q. E. D.

Theorem 5: Let L be a type 1_A language generated by a type 1_A grammar G. Let ϕ and ψ be two strings on V such that $\phi\Rightarrow\psi$. Then the commutative images of ϕ and ψ are distinct. (See definition 13.)

Proof: Extend the function f of Definition 18 to all strings on V by taking $f(\omega\eta) = f(\eta\omega) = f(\omega) + f(\eta)$. Then f can be thought of as a function on $\Phi(L)$. But if ϕ and ψ have the same length and $\phi\rightarrow\psi$, then $f(\phi) > f(\psi)$. Hence if ϕ and ψ have the same length and $\phi\Rightarrow\psi$, then $f(\phi) > f(\psi)$. Hence $\Phi(\phi) \neq \Phi(\psi)$. Q. E. D.

Theorem 6: If G is of type 1_B, $x \in L$; D is a derivation of x, x_1 is a phrase of x, of type A with respect to D, and x_1 is a phrase of x of type B with respect to D, then A = B.

Proof: If $A \neq B$, then we would have $A\Rightarrow B$ or $B\Rightarrow A$. But this is impossible.

Q. E. D.

Remark: It is not difficult to show that there exist a type 1 grammar G and strings $\phi AB\psi$ $\phi BA\psi$ such that $\phi AB\psi\Rightarrow\phi BA\psi$. Such situations are obviously "unfortunate" from a grammatical point of view.

Theorem 7: Every type 2 language is a type 1_B language.

Proof: By lemma 2. Q. E. D.

Theorem 8: There are languages of type 1_B which are not of type 2.

Proof: Consider the rules.

(α) $S \rightarrow aX^2b$

(β) $X \rightarrow CX^4D$

1 $\begin{cases} DX \rightarrow XEX \\ EX \rightarrow EXD \\ XEXD \rightarrow X^4D \end{cases}$

$$2\begin{cases} XC \rightarrow XFX \\ XF \rightarrow CXF \\ CXFX \rightarrow CX^4 \end{cases}$$

$$3\begin{cases} aC \rightarrow ac \\ cC \rightarrow cc \\ cX \rightarrow ce \\ eX \rightarrow ee \\ eD \rightarrow ed \\ dD \rightarrow dd \end{cases}$$

Now notice that for an X to turn terminal, it is necessary that it should be preceded by a C. For a C to turn terminal it must be preceded by a or c. Hence if we have a generated string containing an X that turns into a terminal string, then it must have the form $ac^kX^\ell D^k b$. Notice that, by the rules of group 1, a D can only move right across X's, quadrupling them, but cannot move right across a C. Similarly, a C can move left across X's, quadrupling them, but cannot move left across a D. Hence, given two applications of rule β, one of the applications must occur "within" the other, or else the C or the D will get "stuck" and we will not get a terminal string. Hence the only terminal strings of the form (and all of that form)

$$\#ac^n e^{2+4^n} d^n b\#$$

are generated.

Under the map $\Theta(e) = 1$, $\Theta(a)$, $\Theta(b)$, $\Theta(c)$, $\Theta(d)$, $\Theta(\#) = 0$, $\Theta(\alpha\beta) = \Theta(\alpha) + \Theta(\beta)$, the numbers $2 + 4^n$ are generated. This is not a semilinear set. (See corollary to Theorem 2.) Q. E. D.

3. A remark on the reduction of CF grammars to a question regarding free rings

Let G be a CF grammar with vocabulary V. Consider the free ring $\underline{R}$ generated by V. Define an operator Θ over $\underline{R}$ as follows:

(1) If $a \in V_T$, $\Theta(a) = a$.

If $A \in V_N$ and $A \rightarrow \omega_i$ are the rules associated with A, then

(2) $\Theta(A) = \sum_i \omega_i$.

(3) $\Theta(\eta\eta') = \Theta(\eta)\Theta(\eta')$.

(4) $\Theta(\eta+\eta') = \Theta(\eta) + \Theta(\eta')$.

Then the generable strings are precisely the ones that appear as terms in some expression $\Theta^n(\#S\#)$ for some n.

For example, let $V = \{\#, a, b, A, S\}$

$S \rightarrow AS$, $A \rightarrow ab$, $A \rightarrow cd$, $S \rightarrow aAa$.

Then

$\Theta(S) = AS + aAa$

$\Theta(A) = ab + cd$

$\Theta(a) = a,\ \Theta(b) = b,\ \Theta(\#) = \#$

Now

$\Theta(\#S\#) = \#AS\# + \#aAa\#$

$\Theta^2(\#S\#) = \#abAS\# + \#cdAS\# + \#abaAa\# + \#cdaAd\# + \#aaba\# + \#acda\#$

etc.

and every derivable string will eventually appear on the right-hand side. Every sentence will be always on the right-hand side after a certain point. (Note: Θ is a homomorphism of $\underline{R}$ into itself. Moreover any homomorphism that does not take a generator of the ring into zero comes from a CF grammar.)

R. J. Parikh

References

1. Y. Bar-Hillel, M. Perles, and E. Shamir, On formal properties of simple phrase structure grammars, Technical Report No. 4, Office of Naval Research, Information Systems Branch, Washington, D. C., 1960.

2. N. Chomsky, Three models for the description of language, Trans. IRE, Vol. IT-2, No. 3, pp. 113-124, 1956.

3. N. Chomsky and G. A. Miller, Finite state languages, Information and Control 1, 91-112 (1958).

4. N. Chomsky, On certain formal properties of grammars, Information and Control 2, 137-167 (1959).

5. S. Scheinberg, On Boolean properties of phrase structure grammars (to be published in Information and Control).