INFORMATION AND CONTROL 6, 131–136 (1963)

Three Theorems on Phrase Structure Grammars of Type 1

PETER S. LANDWEBER*

Burroughs Corporation, Burroughs Laboratories, Paoli, Pennsylvania

It is shown that the class of languages generated by type 1 phrase structure grammars is not enlarged by allowing end markers, that this class is closed under the operation of intersection, and that those languages representable by linear bounded automata belong to this class.

INTRODUCTION

The investigation of type 1 phrase structure grammars, as defined by Chomsky (1959), runs into several difficulties. Although these grammars generate primitive recursive sets, the author has shown in his paper (1962) that a number of decision problems for these grammars are undecidable, including most decision problems which are decidable for grammars of types 2 and 3. Furthermore, there are no sharp necessary conditions for a language to be type 1; for example, it is not known whether the class of type 1 languages is closed under complementation, although the author suspects that this is not the case.

We shall first prove that the class of languages generated by type 1 grammars with end markers coincides with the class of languages generated by type 1 grammars without end markers. Therefore, we can use either kind of grammar without loss of generality.

Next, we shall show that the class of type 1 languages is closed under the operation of intersection. This result came as quite a surprise, and provides a significant extension of the theorem due to Bar-Hillel, Perles, and Shamir (1961), which states that the intersection of a type 2 language with a type 3 language is a type 2 language. The generation of the intersection $L_{G_1} \cap L_{G_2}$ of two type 1 languages which we shall give proceeds as follows. In a single grammar G, we generate nonterminal

* Present address: Mathematics Department, Harvard University, Cambridge, Massachusetts.

strings which correspond to certain pairs of strings; the first member of such a pair belongs to L_{G_1}, and the second to L_{G_2}. The correspondence is established by using doubly indexed symbols. If the two members of the pair corresponding to one of these nonterminal strings coincide, then the string can be transformed into a terminal string of G.

Finally, it will be proved that every language representable by a linear bounded automation is a type 1 language. Since Myhill has shown (1960) that all rudimentary languages are representable by linear bounded automata, it follows that a considerable family of explicitly definable languages are of type 1. This partially accounts for the difficulty in finding languages of a simple form which are not of type 1.

By an *alphabet* we mean a finite nonvoid set. Any set of finite nonempty words over an alphabet will be called a *language*. (The null word will be systematically excluded.) Since type 1 grammars and linear bounded automata have been defined and studied elsewhere, we shall give only brief formal definitions of these devices.

A type 1 *grammar* with end marker is a quintuple $G = (V_N, V_T, S, \divideontimes, P)$, where V_N and V_T are disjoint alphabets, S and $\divideontimes$ are special symbols of V_N, and P is a finite set of rewriting rules $\varphi \rightarrow \psi$ satisfying the following conditions:

(1) φ and ψ are words over $V = V_N \cup V_T$; φ involves at least one symbol of V_N other than $\divideontimes$; the length of ψ is not less than the length of φ.

(2) φ and ψ may have any of the forms W, $\divideontimes W$, or $W \divideontimes$ where W is a word not involving $\divideontimes$; φ and ψ must be of the same form.

V_N, V_T, and $V = V_N \cup V_T$ are the nonterminal, terminal, and total vocabularies of G, respectively. S is the sentence symbol and $\divideontimes$ is the end marker.

We write $\varphi_1 \rightarrow \varphi_2$, in case φ_2 arises from φ_1 by application of one of the rewriting rules. If φ_2 arises from φ_1 after a finite number of applications of the rewriting rules, we write $\varphi_1 \Rightarrow \varphi_2$. The grammar G generates the language

$$L_G = \{x | \divideontimes S \divideontimes \Rightarrow \divideontimes x \divideontimes\},$$

where x denotes an arbitrary word over the terminal alphabet V_T.

A type 1 grammar without end marker is a quadruple $G = (V_N, V_T, S, P)$, defined as above, except that $\divideontimes$ does not occur in the rewriting rules, and

$$L_G = \{x | S \Rightarrow x\}.$$

A linear bounded *automaton* is a quintuple $A = (S, \Sigma, s, F, \Phi)$, where S and Σ are alphabets, s is a special element and F a special subset of S, and Φ is a function

$$\Phi: \Sigma \times S \to \Sigma \times \{L, C, R\} \times S.$$

S is the set of states of A, Σ the alphabet, s the initial state, F the set of final states, and Φ the behavior function. If $\Phi(a_1, s_1) = (a_2, L, s_2)$, and symbol a_1 is being scanned in state s_1, the machine prints a_2 in place of a_1 and scans the symbol to the left of a_1 in state s_2; R denotes a move to the right, and C indicates that the scanner does not move. The machine is provided initially with a word x over Σ presented on a tape whose length is equal to that of x, and begins by scanning the leftmost symbol of x in the initial state. The machine stops if the scanner goes off the tape to the left or right. The word x is *accepted* by the automaton A if the machine stops off the right-hand end of the tape in one of its final states. The language L_A determined by A is defined to be the set of accepted words.

It should be remarked that certain of the symbols in the alphabet Σ of A may be auxiliary. If Σ_0 is the subset of nonauxiliary symbols of Σ, then L_{A,Σ_0} will denote the set of words over Σ_0 which are accepted by A. Notice that $L_{A,\Sigma_0} = L_A \cap (\Sigma_0)^*$, where $(\Sigma_0)^*$ denotes the set of all words over Σ_0. We shall prove in Theorem 3 that L_A is always a type 1 language. Since $(\Sigma_0)^*$ is a type 3 language, it will follow from the intersection theorem that L_{A,Σ_0} is a type 1 language. Hence, we shall have no need to consider a subalphabet Σ_0 of Σ to obtain the general result that L_{A,Σ_0} is always a type 1 language.

TYPE 1 GRAMMARS

We shall prove here that every type 1 grammar with end marker is equivalent to a type 1 grammar without end marker. Thus, the class of languages generated by type 1 grammars is not enlarged if end markers are permitted. Of course, there are occasions when end markers are extremely convenient, as in Theorem 3.

THEOREM 1. *Every type 1 grammar with end marker is equivalent to a type 1 grammar without end marker.*

PROOF: Let $G = (V_N, V_T, S, \aleph, P)$ be a type 1 grammar with end marker. We shall construct a type 1 grammar $G_1 = (V_{N1}, V_{T1}, S_1, P_1)$ without end marker such that $L_{G_1} = L_G$.

For each α in the total vocabulary V of G other than $\aleph$, introduce

new symbols ${}^{\#}\alpha$, $\alpha^{\#}$, and ${}^{\#}\alpha^{\#}$; the nonterminal vocabulary V_{N1} of G_1 will consist of all symbols ${}^{\#}\alpha$, $\alpha^{\#}$, and ${}^{\#}\alpha^{\#}$, as well as all symbols A in V_N other than $\#$. If $\varphi = \alpha_1 \alpha_2 \cdots \alpha_r$ ($r \geqq 1$) is a word over V not involving $\#$, let ${}^{\#}\varphi = {}^{\#}\alpha_1 \alpha_2 \cdots \alpha_r$, $\varphi^{\#} = \alpha_1 \alpha_2 \cdots \alpha_r{}^{\#}$, and ${}^{\#}\varphi^{\#} = {}^{\#}\alpha_1 \alpha_2 \cdots \alpha_r{}^{\#}$; thus, ${}^{\#}\varphi$, $\varphi^{\#}$, and ${}^{\#}\varphi^{\#}$ are defined for all (nonempty) words over V not involving $\#$.

The rewriting rules of G_1 will now be defined in several groups; of course, we take $V_{T1} = V_T$ and $S_1 = {}^{\#}S^{\#}$.

I. If $\varphi \to \psi$ is a rule of G, let $\varphi \to \psi$, ${}^{\#}\varphi \to {}^{\#}\psi$, $\varphi^{\#} \to \psi^{\#}$, and ${}^{\#}\varphi^{\#} \to {}^{\#}\psi^{\#}$ be rules of G_1.

II. If $\#\varphi \to \#\psi$ is a rule of G, let ${}^{\#}\varphi \to {}^{\#}\psi$ and ${}^{\#}\varphi^{\#} \to {}^{\#}\psi^{\#}$ be rules of G_1.

III. If $\varphi\# \to \psi\#$ is a rule of G, let $\varphi^{\#} \to \psi^{\#}$ and ${}^{\#}\varphi^{\#} \to {}^{\#}\psi^{\#}$ be rules of G_1.

IV. For $a \in V_T$, let ${}^{\#}a \to a$, $a^{\#} \to a$, and ${}^{\#}a^{\#} \to a$ be rules of G_1.

It is easy to see that there is a G derivation $\#S\# \Rightarrow \#x\#$ if and only if there is a G_1 derivation $S_1 \Rightarrow x$; in fact, we can convert G derivations from $\#S\#$ to G_1 derivations from S_1, and conversely, if we identify $\#\varphi$ with ${}^{\#}\varphi$, $\varphi\#$ with $\varphi^{\#}$, and $\#\varphi\#$ with ${}^{\#}\varphi^{\#}$. Thus, $L_{G_1} = L_G$, and the proof is complete.

THE INTERSECTION THEOREM

THEOREM 2. *Let* $G_1 = (V_{N1}, V_{T1}, S_1, P_1)$ *and*

$$G_2 = (V_{N2}, V_{T2}, S_2, P_2)$$

be type 1 grammars. Then there is a type 1 grammar G *which generates the language* $L_{G_1} \cap L_{G_2}$.

PROOF: We will suppose that G_1 and G_2 have the same terminal vocabularies:

$$V_{T1} = V_{T2} = \Sigma = \{a_1, a_2, \cdots, a_p\}.$$

Let α_i ($i = 1, 2, \cdots, p$) be new symbols.

Modify G_1 by replacing all occurrences of the terminal symbols a_i with α_i; let $\bar{G}_1$ denote the resulting grammar. The total vocabulary $\bar{V}_1$ of $\bar{G}_1$ is $\{\alpha_1, \alpha_2, \cdots, \alpha_p\} \cup V_{N1}$, which we may list as $\{\alpha_1, \alpha_2, \cdots, \alpha_n\}$, with α_n the sentence symbol of $\bar{G}_1$; thus, $\bar{V}_{N1} = \{\alpha_{p+1}, \cdots, \alpha_n\}$. G_2 is not changed, but we will list the total vocabulary V_2 of G_2 as $\{\beta_1, \beta_2, \cdots, \beta_m\}$, with β_m the sentence symbol, and $\beta_i = a_i$ for $i = 1, 2, \cdots, p$.

Introduce further symbols

$$\Gamma = \{\gamma_{i,j}\} \ (i = 1, 2, \cdots, n; j = 1, 2, \cdots, m)$$

to represent pairs (α_i, β_j). A grammar $G = (V_N, V_T, S, P)$ will now be defined. We take $V_N = \bar{V}_1 \cup \Gamma$, $V_T = \Sigma$ and $S = \gamma_{n,m}$. The productions of P will be defined in four groups.

I. If $\alpha_{i_1} \cdots \alpha_{i_q} \to \alpha_{i_1} \cdots \alpha_{j_r}$ is a rule of $\bar{G}_1$, then it is a rule of G in addition to $\alpha_{i_1} \cdots \alpha_{i_{q-1}} \gamma_{i_q,m} \to \alpha_{j_1} \cdots \alpha_{j_{r-1}} \gamma_{j_r,m}$.

II. If $\beta_{i_1} \cdots \beta_{i_q} \to \beta_{j_1} \cdots \beta_{j_r}$ is a rule of G_2, then $\alpha_{k_1} \cdots \alpha_{k_s} \gamma_{k_{s+1} i_1} \cdots \gamma_{k_r,i_q} \to \gamma_{k_1,j_1} \cdots \gamma_{k_r,j_r}$ (where $k_1, k_2, \cdots, k_r = 1, 2, \cdots, p$, and $s \geqq 0$) is a rule of G; note that there are p^r rules of G for each rule of G_2.

III. $\gamma_{i,j}\alpha_k \to \alpha_i \gamma_{k,j}$ $\quad (i, k = 1, 2, \cdots, p; j = 1, 2, \cdots, m)$

and $\alpha_i \gamma_{k,j} \to \gamma_{i,j}\alpha_k$ $\quad (i, k = 1, 2, \cdots, p; j = 1, 2, \cdots, m)$

are rules of G.

IV. $\gamma_{i,i} \to a_i$ is a rule of G for $i = 1, 2, \cdots, p$.

The rules of group I generate the derivable strings of $\bar{G}_1$, $a_{i_1} \cdots a_{i_q}$, with the rightmost character replaced by $\gamma_{i_q,m}$; i.e., $a_{i_1} \cdots a_{i_{q-1}} \gamma_{i_q,m}$. The rules of group II allow the productions of G_2 to be carried out within the strings (resulting from the group I rules) that correspond to terminal strings in $\bar{G}_1$. The strings of G_2 are encoded in $\gamma_{i,j}$'s along with the $\bar{G}_1$ strings which remain unchanged. The rules of group III allow the γ's to slide across the α's, but do not change the coded $\bar{G}_1$ or G_2 strings. These rules permit the strings to be rearranged for the group II rules. Finally, the rules of group IV permit a string x over Σ to be delivered if it has been derived in group I $(\bar{G}_1)$ and in groups II and III (G_2).

It is easy to confirm that $S \Rightarrow a_{i_1} \cdots a_{i_q}$ if and only if $S \Rightarrow \gamma_{i_1,i_1} \cdots \gamma_{i_q,i_q}$. We must then verify that this last condition holds if and only if $a_{i_1} \cdots a_{i_q} \in L_{G_1} \cap L_{G_2}$. It is clear that from a derivation $S \Rightarrow \gamma_{i_1,i_1} \cdots \gamma_{i_q,i_q}$ we can obtain a G_1 derivation $S_1 \Rightarrow a_{i_1} \cdots a_{i_q}$ and a G_2 derivation $S_2 \Rightarrow a_{i_1} \cdots a_{i_q}$. If $a_{i_1} \cdots a_{i_q} \in L_{G_1} \cap L_{G_2}$, one can convince himself that the rules of groups I, II, and III provide for a derivation $S \Rightarrow \gamma_{i_1,i_1} \cdots \gamma_{i_q,i_q}$. Hence, $L_G = L_{G_1} \cap L_{G_2}$.

LINEAR BOUNDED AUTOMATA AND TYPE 1 GRAMMARS

THEOREM 3. *If A is a linear bounded automaton, then there is a type 1 grammar G such that $L_G = L_A$.*

PROOF: Let $A = (S, \Sigma, s_0, F, \Phi)$ be a linear bounded automaton. We shall construct a type 1 grammar G with end marker such that $L_G = L_A$; G simulates the reverse of the operation of the automaton A.

Suppose $\Sigma = \{a_1, a_2, \cdots, a_n\}$ and $S = \{s_0, s_1, \cdots, s_m\}$, with s_0 the initial state. We introduce new symbols b_{ij} $(i = 1, 2, \cdots, n; j = 0, 1, \cdots, m)$ to represent a_i and s_j simultaneously.

The grammar $G = (V_N, V_T, S_1, \$, P)$ will now be defined. We take $V_N = \{b_{ij}\} \cup \{S_1, A, \$\}$ and $V_T = \Sigma$, where S_1 is the sentence symbol of G. The rules of G will be defined in groups.

I. $S_1 \to b_{ij}$ and $S_1 \to A b_{ij}$ will be rules of G if the machine moves to the right and enters a final state (i.e., a state in F) when it scans a_i in state s_j. $A \to a_i$ and $A \to A a_i$ will be rules of G for $i = 1, 2, \cdots, n$. These rules permit derivations $\$ S_1 \$ \Rightarrow \$ a_{i_1} \cdots a_{i_{r-1}} b_{i_r j} \$$, provided that the machine moves to the right and enters a state of F when it scans a_{i_r} in state s_j.

II. If $\Phi(a_k, s_l) = (a_i, L, s_j)$, then $b_{rj} a_i \to a_r b_{kl}$ will be a rule of G for $r = 1, 2, \cdots, n$; if $\Phi(a_k, s_l) = (a_i, R, s_j)$, then $a_i b_{rj} \to b_{kl} a_r$ will be a rule of G for each r; and if $\Phi(a_k, s_l) = (a_i, C, s_j)$, then $b_{ij} \to b_{kl}$ will be a rule of G. It is clear that these rules mimic the inverses of the operations of A.

III. Finally, we include the rule $\$ b_{i0} \to \$ a_i$ for $i = 1, 2, \cdots, n$. Observe that b_{i0} indicates that the symbol a_i is being scanned in the initial state s_0 of the automaton.

The reader should have no difficulty in convincing himself that G is a type 1 grammar with end marker, and that $L_G = L_A$. This completes the proof of the theorem.

COROLLARY. *Every rudimentary language is a type 1 language.*

PROOF: Myhill has shown that every rudimentary language L is representable by a linear bounded automaton A; i.e., $L = L_A$ for some linear bounded automaton A. Then use Theorem 2.

ACKNOWLEDGMENT

The author is indebted to Mr. K. H. Speierman of the Systems Department, Burroughs Laboratories, Research Division, for his valuable guidance and encouragement.

RECEIVED: February 19, 1963

BAR-HILLEL, Y., PERLES, M., AND SHAMIR, E. (1961), On formula properties of simple phrase structure grammars. *Z. Phonetik, Sprachwiss. Kommunikationsforsch.* **14**, 143–172.

CHOMSKY, N. (1959), On certain formal properties of grammars. *Inform. and Control* **2**, 133–167.

LANDWEBER, P. S. (1962), Decision problems of phrase structure grammars. Burroughs Corporation, internal report.

MYHILL, J. (1960), Linear bounded automata. Wright Air Development Division, Tech. Note No. 60–165, Cincinnati, Ohio.