INFORMATION AND CONTROL 21, 235–260 (1972)

String Adjunct Grammars: II. Equational Representation, Null Symbols, and Linguistic Relevance*

A. K. JOSHI,† S. R. KOSARAJU,‡ AND H. M. YAMADA§

The Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, Pennsylvania 19104

In this paper, we continue the study of String Adjunct Grammars (AG) introduced in Joshi et al. (1972). In particular, equational representations of LAG's, LAG's with null symbols, and some special cases of LAG's are studied. Linguistic relevance of these grammars is also discussed in some detail.

1. INTRODUCTION

String Adjunct Grammars (AG) were introduced in Joshi *et al.* (1972). We continue the study of these grammars in this paper. In Section 2 we will study an equational representation of the local adjunct grammars (LAG). In Section 3 we will discuss a generalization of AG's by allowing certain very restricted kinds of nonterminals in the sense of Phrase Structure Grammars (PSG). A few additional points of comparison with PSG's are stated in Section 4. Several special cases of AG's motivated by linguistic considerations are discussed in Section 5. Finally, in Section 6 we will discuss in some detail the linguistic relevance of AG's and other related grammars.

2. EQUATIONAL REPRESENTATION OF LAG

2.1. *Graph of LAG*

Let $G = (\Sigma_c, J)$ be an LAG. We shall define the graph $\Gamma(G)$ of G as follows. Σ_h is the set of all basic host strings, Σ_a is the set of all adjunct

* This work was partially supported by NSF Grant GS-159, NSF Grant GP-5561 and U.S. Army Research Office, Durham (DA-31-124 ARO(D)-98).

† Also in the Department of Linguistics. At present on leave at The Institute for Advanced Study, Princeton, NJ.

‡ Now at Johns Hopkins University, Baltimore, MD.

§ At present on leave at the Information Science Laboratory, Faculty of Science, University of Tokyo, Bunkyo-ku, Tokyo, Japan.

strings, and $\Sigma = \Sigma_c \cup \Sigma_h \cup \Sigma_a$. Then $\Gamma(G) = (\Sigma, J)$ is the labeled directed graph such that Σ is the node set, and J is the set of labeled directed branches. Each $(\sigma_i, \sigma_j, \xi_k) \in J$ has arrow from σ_j to σ_i with ξ_k as its label. Finally, the nodes of $\sigma_i \in \Sigma_c$ are double circled.

EXAMPLE 2.1.1. Let G be defined by $\Sigma_c = \{ab, abc\}$, and $J = \{u_1 = (ab, ac, r_1), u_2 = (ab, a, r_1), u_3 = (ac, bc, l_1), u_4 = (bc, ac, r_2), u_5 = (bc, a, l_2), u_6 = (a, bbcc, r_1), u_7 = (a, ac, r_1), u_8 = (ca, ac, l_1), u_9 = (bc, bc, r_1)\}$. Then $\Gamma(G) = (\Sigma, J)$ is given by $\Sigma = \{a, ab, abc, ac, bbcc, bc, ca\}$ and J as shown in Fig. 1, where u's in branch designations are for identification purposes.

Denote adjunction rule $(\sigma_{h_i}, \sigma_{a_i}, \xi_i) \in J$ by u_i. Define a relation ">" on J by $u_i > u_j \Leftrightarrow \sigma_{h_i} = \sigma_{a_j}$.

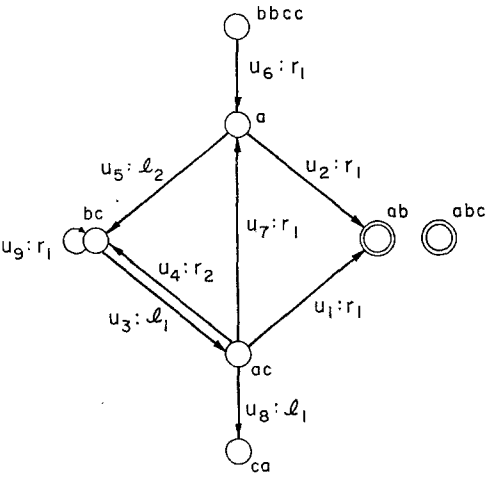

FIG. 1. Graph $\Gamma(G) = (\Sigma, J)$ of grammar in Example 2.1.1.

EXAMPLE 2.1.2. The relation ">" on J in the above example is as shown in Fig. 2, where a double circle indicates that the host string is also a center string.

A cycle of J, if any, is defined by a sequence $u_{i_1} u_{i_2} \cdots u_{i_s} \in J^*$, $s \geqslant 1$, such that (a) if $s = 1$, $u_{i_1} > u_{i_1}$, otherwise (b) $u_{i_j} > u_{i_{j+1}}$, $1 \leqslant j < s$, and $u_{i_s} > u_{i_1}$. We call s the *length* of cycle. Note that if $u_1 \cdots u_s$ is a cycle, then $u_1 \cdots u_s u_1 \cdots u_s$ is also a cycle. If the host strings of adjunction rules in a cycle are all distinct, then we call the cycle *simple*, otherwise *complex*. If J does not contain any cycles, the J and the LAG is said to be *cyclefree*, otherwise *cycled*.

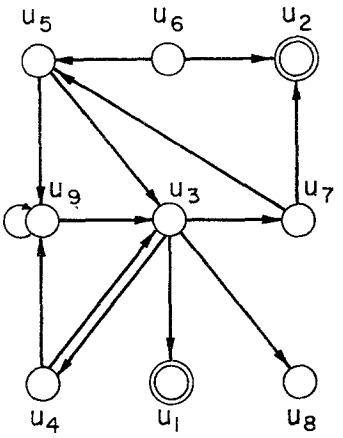

FIG. 2. Relation ">" on J.

EXAMPLE 2.1.3. In the above example, $u_9 = (bc, bc, r_1)$ is a simple cycle of length 1, $u_5u_3u_7$ is a simple cycle of length 3, and $u_5u_3u_4u_9u_3u_7$ is a complex cycle of length 6. Since J contains cycles, G is cycled.

Let relation "$\geqslant^*$" on J be the reflexive and transitive closure of $>$. Then J is cyclefree if and only if $\geqslant^*$ is also antisymmetric and there is no rule $u_i \in J$ such that $u_i > u_i$.

Define a relation "R" on J by $u_i \, R \, u_j \Leftrightarrow u_i \geqslant^* u_j \,\&\, u_j \geqslant^* u_i$. Then R is an equivalence relation of strong connectedness. Denote by $[u_i]$ the equivalence class containing u_i. We note that (1) a cycle is contained in an equivalence class; (2) an equivalence class $[u_i]$ such that $\#[u_i] = 1$ contains a cycle if and only if $\sigma_{h_i} = \sigma_{a_i}$ in u_i; (3) given an equivalence class $[u_i]$, $\#[u_i] > 1$ if and only if it contains a simple cycle such that its length s satisfies $1 < s \leqslant \#[u_i]$; and (4) if an equivalence class contains a cycle, then there exists a cycle which contains all elements of the equivalence class.

A relation ">" is defined on the set of equivalence classes of J by $[u_i] > [u_j] \Leftrightarrow (\exists u_i' \in [u_i])(\exists u_j' \in [u_j])(u_i' > u_j')$. This relation is extended to "$\geqslant^*$" which is the reflexive and transitive closure of $>$. It is easily seen that the relation $\geqslant^*$ over the equivalence classes of J is a partial ordering.

In Example 2.1.1 we have a partial ordering among the equivalence classes of J as in Fig. 3, where ( ) indicates that the host string is in Σ_c. Note that u_8 is ineffective and by eliminating it the grammar becomes effective.

We defined the relation $>$ on J. For later use, we define a similar relation $>_\sigma$ on $\Sigma = \Sigma_c \cup \Sigma_h \cup \Sigma_a$ by $\sigma_i >_\sigma \sigma_j \Leftrightarrow (\exists u \in J)(u = (\sigma_j, \sigma_i, \xi))$. It is clear then that $\Gamma(G) = (\Sigma, J)$ is the graph of the relation $>_\sigma$ on Σ *except* that

parallel branches of the same directivity be considered as one branch. Thus the relation $>_\sigma$ on Σ of the previous example can be represented by the same graph as in Fig. 1, if we ignore the branch labels. Cycles, relations $\geqslant_\sigma^*$ and R_σ can similarly be defined for Σ as in the case for J, and similar propositions hold here as before.

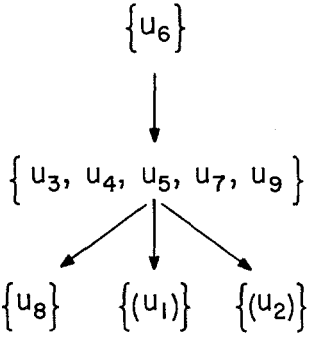

FIGURE 3

2.2. *Equational Representation of LAG*

Let x and y denote, in some well-defined manner, sets of finite strings on A. We define, in the usual manner, three operations on x and y; (1) concatenation "$\cdot$", (2) disjunction "$\mathbf{v}$", and (3) Kleene closure "$*$".

Given an LAG (Σ_c, J), obtain the graph $\Gamma(G) = (\Sigma, J)$ of G. Associate a grammar $G(\sigma_i) = (\{\sigma_i\}, J)$ to each node σ_i, and let $\lambda(\sigma_i)$ denote the language generated by $(\{\sigma_i\}, J)$. It is immediate that $G(\sigma_i)$ generates the set $\lambda(\sigma_i)$ of strings which are either (1) generated by J of G using last in their derivation a rule $u \in J$ such that $u = (\sigma_i, \sigma_j, \xi_k)$ for some σ_j and ξ_k, if $\sigma_i \in \Sigma_h$, or (2) $\sigma_i \notin \Sigma_h$ and $G(\sigma_i)$ generates a unit set $\{\sigma_i\}$, either $\sigma_i \in \Sigma_a - \Sigma_h$ or $\sigma_i \in \Sigma_c - \Sigma_h$, or in both. In general, $G(\sigma_i)$ is no longer effective even if G is. Then the language $L(G)$ of G is denoted by

$$L(G) = \bigvee_{\sigma_i \in \Sigma_c} \lambda(\sigma_i).$$

Given a node σ_i, a node $\sigma_k{}^i$ is said to be a *predecessor* of σ_i if $\sigma_k{}^i >_\sigma \sigma_i$. Suppose a node σ_i of $\Gamma(G)$, labeled with $\lambda(\sigma_i)$ has predecessors $\sigma_1{}^i, \sigma_2{}^i, \ldots, \sigma_m{}^i$, $m \geqslant 0$, respectively, labeled with $\lambda(\sigma_1{}^i)$, $\lambda(\sigma_2{}^i), \ldots, \lambda(\sigma_m{}^i)$, and the branches incident upon σ_i from each $\sigma_r{}^i$, $1 \leqslant r \leqslant m$, are labeled with $\xi_{r1}, \xi_{r2}, \ldots, \xi_{rn_r}$. Let $X = \{\xi_{11}, \ldots, \xi_{1n_1}, \xi_{21}, \ldots, \xi_{2n_2}, \ldots, \xi_{m1}, \ldots, \xi_{mn_m}\}$ be the set of all those

labels. Order X such that $\xi_s < \xi_t$ either if $s < t$, or if $s = t$, $\xi_s = l_s$, and $\xi_t = r_s$. There may be a set of branch designations which are identical, and if such is the case, they should be represented as one but separately accounted for. Let the ordering of X be $(\xi_1, \xi_2, \ldots, \xi_n)$, where only distinct elements are shown, i.e., $n \leqslant n_1 + n_2 + \cdots + n_m$. Take σ_i and factor it into $\sigma_i = \sigma_{i_1}\sigma_{i_2} \cdots \sigma_{i_{n+1}}$ such that for all k, $1 \leqslant k \leqslant n$, if $\xi_k = l_{n_k}$ then $\lg(\sigma_{i_1}\sigma_{i_2} \cdots \sigma_{i_k}) = n_k - 1$, but if $\xi_k = r_{n_k}$ then $\lg(\sigma_{i_1}\sigma_{i_2} \cdots \sigma_{i_k}) = n_k$, where lg denotes length. Note that some of σ_{i_k} may be null, i.e., $\sigma_{i_k} = \epsilon$. The *defining equation* of $\lambda(\sigma_i)$ associated with node σ_i is now

$$\lambda(\sigma_i) = \sigma_{i_1}\mu_{i_1}\sigma_{i_2}\mu_{i_2} \cdots \sigma_{i_n}\mu_{i_n}\sigma_{i_{n+1}},$$

where

$$\mu_{i_k} = (\lambda(\sigma_k{}^i))^*,$$

if there is only one branch which has ξ_k as its label, $1 \leqslant k \leqslant n$, and

$$\mu_{i_k} = [\lambda(\sigma^i_{k_1}) \vee \lambda(\sigma^i_{k_2}) \vee \cdots \vee \lambda(\sigma^i_{k_r})]^*,$$

if there are r incident branches with the same ξ_k and $\lambda(\sigma^i_{k_1}), \ldots, \lambda(\sigma^i_{k_r})$ are the labels of nodes from which branches with ξ_k are incident upon node σ_i. Now if $\sigma_i \in (\Sigma_c \cup \Sigma_a) - \Sigma_h$, then $\lambda(\sigma_i) = \sigma_i$. In general, the right-hand side of the defining equation of $\lambda(\sigma_i)$, $\sigma_i \in \Sigma_h$, may involve any of $\lambda(\sigma_k)$ such that $\sigma_k \in \Sigma_a$, but not others. We shall denote the defining equation of $\lambda(\sigma_i)$ for $\sigma_i \in \Sigma_h$ by

$$\lambda(\sigma_i) = f_i(\lambda(\sigma_{j1}), \lambda(\sigma_{j2}), \ldots, \lambda(\sigma_{js}), \sigma_i),$$

where $s = \#\Sigma_a$, and $\sigma_{jr} \in \Sigma_a$, $1 \leqslant r \leqslant s$. In summary, we have

THEOREM 2.2.1. *The language $L(G)$ of G is defined by the following set of simultaneous equations:*

$$\lambda(\sigma_i) = f_1(\lambda(\sigma_{j1}), \lambda(\sigma_{j2}), \ldots, \lambda(\sigma_{js}), \sigma_i),\ \sigma_i \in \Sigma_h,\ \sigma_{jr} \in \Sigma_a,\ s = \#\Sigma_a,$$

$$L(G) = \bigvee_{\sigma_i \in \Sigma_c} \lambda(\sigma_i).$$

EXAMPLE 2.2.1. Let $G = (\Sigma_c, J)$, where $\Sigma_c = \{ab, abc\}$, and $J = \{u_1 = (ab, ac, r_1), u_2 = (ab, a, r_1), u_3 = (ac, bc, l_1), u_4 = (ac, bc, r_2), u_5 = (bc, a, l_2), u_6 = (a, bbcc, r_1), u_7 = (a, ac, r_1), u_8 = (ca, ac, l_1), u_9 = (bc, bc, r_1)\}$. Then we can write (1) $\lambda(a) = a(\lambda(bbcc) \vee \lambda(ac))^*$; (2) $\lambda(ab) = a(\lambda(a) \vee \lambda(ac))^*b$; (3) $\lambda(ac) = (\lambda(bc))^*ac(\lambda(bc))^*$; (4) $\lambda(bc) = b(\lambda(bc))^*(\lambda(a))^*c$; (5) $\lambda(ca) = (\lambda(ac))^*ca$;

(6) $\lambda(abc) = abc$; (7) $\lambda(bbcc) = bbcc$; and (8) $L(G) = \lambda(ab) \vee \lambda(abc)$. Since $\lambda(ca)$ does not contribute to $L(G)$ at all, Eqs. (1)–(8) except (5) will define $L(G)$.

From the construction, it follows that if a node σ_i of $\Gamma(G)$ has incoming branches with $\xi = l_n$ such that $n > 1$, and there are no incoming branches with $\xi = r_{n-1}$, then all such l_n can be replaced by r_{n-1} without changing the language. Similarly, if σ_i has incoming branches with $\xi = r_n$ such that $n < \lg(\sigma_i)$, and there are no incoming branches with $\xi = l_{n+1}$, then all such r_n can be replaced by l_{n+1} without changing the laguage. Such changes of branch labels can be incorporated in the rules of J of G. If all such possible changes are made, we call the grammar *right normalized* for the former, and *left normalized* for the latter.

EXAMPLE 2.2.2. In the previous example, the grammar is already right normalized. Note that u_5 cannot be changed to (bc, a, r_1) because of the conflict with u_9. The left normalization of G changes u_1 to $u_1' = (ab, ac, l_2)$, u_2 to $u_2' = (ab, a, l_2)$. u_9 cannot be changed to (bc, be, l_2) because of the conflict with u_5.

2.3. *Set of Simultaneous Equations Defining LAG*

We now show that a certain set of simultaneous equations defines an LAG. Let A be a finite alphabet and A^* be the free monoid on A. Let $W = \{x_1, x_2, \ldots, x_m\}$ be a finite set of variables to denote languages defined on A, and $P(W)$ be the power set of W. We shall use three operators on the elements of $A \cup W$, namely, concatenation, disjunction, and Kleene closure. For each

$$w_i = \{x_{i_1}, x_{i_2}, \ldots, x_{i_{n_i}}\} \in P(W),$$

let

$$\bigvee w_i = x_{i_1} \vee x_{i_2} \vee \cdots \vee x_{i_{n_i}}.$$

For $\sigma_{ij} \in A^*$, define a set of m simultaneous equations by

$$x_i = (\vee w_{i1})^* \sigma_{i1} (\vee w_{i2})^* \sigma_{i2} \cdots \sigma_{in_i} (\vee w_{i(n_i+1)})^*,$$

$1 \leqslant i \leqslant m$, such that for $1 \leqslant i, j \leqslant m$, if $i \neq j$, then

$$\sigma_{i1}\sigma_{i2} \cdots \sigma_{in_i} \neq \sigma_{j1}\sigma_{j2} \cdots \sigma_{jn_j},$$

and for any $1 \leqslant i \leqslant m$ and any $1 < j < n_i - 2$, σ_{ij} and $\sigma_{i(j+1)}$ are not simultaneously equal to ϵ [i.e., not more than two $(\vee w)^*$ are directly

concatenated], and σ_{i1} and σ_{in_i} are not ϵ, i.e., there is no more than one $(\vee w)^*$ at each end. Note that, since $\varnothing \in P(W)$ and $\vee \varnothing = \varnothing^* = \epsilon$, $(\vee w_{ir})^*$ may be ϵ for $r = 1$ or $r = n_i + 1$. Without loss of generality we assume that $(\vee w_{ir}) \neq \epsilon$ for $1 \leqslant r \leqslant n_i$; otherwise we can simply rename $\sigma_{i(r-1)}\sigma_{ir}$ as $\sigma'_{i(r-1)}$.

THEOREM 2.3.1. *The set of simultaneous equations*

$$x_i = (\vee w_{i1})^* \sigma_{i1} (\vee w_{i2})^* \sigma_{i2} \cdots \sigma_{in_i} (\vee w_{i(n_i+1)})^*, \qquad 1 \leqslant i \leqslant m,$$

defined above, together with $L = \vee w_k$ *for some* k, *determines an* LAG.

Proof. For each i, $1 \leqslant i \leqslant m$, let $\sigma(x_i) = \sigma_i = \sigma_{i1}\sigma_{i2} \cdots \sigma_{in_i}$, and let σ_i be the node associated with x_i. Let $\Sigma = \{\sigma_i\}$ be the node set of a graph to be constructed, and let $\Sigma_c \subset \Sigma$ be the set of those nodes which are associated with $x_i \in w_k$ of $L = \vee w_k$, therefore, the nodes in the set are double circled. For each x_i, examine each $\vee w_{i_k}$ in the equation for x_i and, if $x_j \in w_{ik}$, then draw a branch from node $\sigma(x_j)$ to node $\sigma(x_i)$, and label it with ξ_{n_j}, which is found as follows. First, if $j = 1$, i.e., $\vee w_{ij} = \vee w_{i1}$, then $\xi_j = l_1$. If $j \neq 1$, then examine the equation for x_i and see $\sigma_{i(j-1)} = \epsilon$. (Note $\sigma_{i1} \neq \epsilon$ and $\sigma_{in_i} \neq \epsilon$ by definition.) If it is, then let $\xi_j = l_{n_j}$, where $n_j = \lg(\sigma_{i1}\sigma_{i2} \cdots \sigma_{i(j-1)}) + 1$. Otherwise let $\xi_j = r_{n_j}$, where $n_j = \lg(\sigma_{i1}\sigma_{i2} \cdots \sigma_{i(j-1)})$. From the construction, it is obvious that the resulting graph is unique for a given set of simultaneous equations, and can be interpreted as the graph of an LAG. Furthermore, from the graph, if we construct the set of simultaneous equations for the grammar of the graph, we recover the set of simultaneous equations we started out from, except the renaming of variables. ∎

In the above construction, Σ_a is the set of node labels which correspond to those variables which appear on the right-hand side of the equations; Σ_h is the set of node labels which correspond to those variables whose defining equation has some variables appearing also on the right-hand side of the equation; and Σ_c is the set of node labels which correspond to those variables which are in w_k such that $L = \vee w_k$.

EXAMPLE 2.3.1. Let $A = \{a, b, c, d\}$, and let $W = \{x, y, z, u, v, w\}$. Let $x = ab(x \vee z)^* u^*a$; $y = ca(u \vee w)^* bdu^*$; $z = z^*cdcby^*(z \vee u)^* ab$; $u = u^*b$; $v = ax^*by^*z^*cu^*v^*dv^*$; $w = (x \vee w)^* adau^*$; and $L = y \vee v$. Then $\sigma(x) = aba$; $\sigma(y) = cabd$; $\sigma(z) = cdcbab$; $\sigma(u) = b$; $\sigma(v) = abcd$; and $\sigma(w) = ada$, which are all distinct, and form Σ_h. $\Sigma_c = \{cabd, abcd\}$. Since all elements of W appear on the right side of some equations for x through w,

$\Sigma_a = \Sigma$. Furthermore, since each of equations for x through w has some variable appearing on the right side, $\Sigma_h = \Sigma$. We will not write explicitly the set J of right normalized adjunction rules (20 rules in this case). They can be read from the graph $\Gamma(G)$ in Fig. 4.

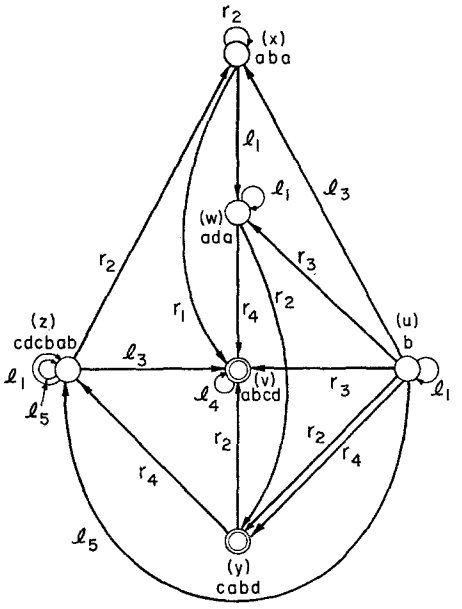

FIG. 4. Graph $\Gamma(G)$ of Example 2.3.1.

The restrictions assumed on the form of the set of simultaneous equations are sufficient conditions for the set to define an LAG. However, they are not necessary conditions. It can be shown that the condition that σ_{i1} and σ_{in_i} [where $n_i = \lg(\sigma_i)$, and $\sigma_i = \sigma_{i1}\sigma_{i2} \cdots \sigma_{in_i}$] are not null is not a necessary condition. Also it is easy to see that, if the set of LAL's is closed under homomorphism (or even under union) which we do not know, the restriction $\sigma_{i1}\sigma_{i2} \cdots \sigma_{in_i} \neq \sigma_{j1}\sigma_{j2} \cdots \sigma_{jn_j}$ is not required, and if the set is closed even under ϵ homomorphism, which we also do not know, then no restrictions are required at all. (See Section 3.2 and Fig. 5 for some closure properties.)

2.4. *Regularity of Cyclefree LAG*

Let $G = (\Sigma_c, J)$ be an effective and cyclefree LAG, and let $\Gamma(G) = (\Sigma, J)$ be its graph. We shall show by construction that the language $L(G)$ of G is

regular. Since G is cyclefree, the equivalence classes $[u_i]$ (defined previously by R such that $u_i \, R \, u_j \Leftrightarrow u_i \geqslant^* u_j \,\&\, u_j \geqslant^* u_i$) are all of unit set. Hence, if G is cyclefree, then $\geqslant^*$ on J is a partial ordering.

EXAMPLE 2.4.1. Let $G = (\Sigma_c, J)$ be determined by $\Sigma_c = \{ab, abc\}$ and $J = \{u_1 = (ab, ac, r_1), u_2 = (ab, a, r_1), u_3 = (ac, bc, l_1), u_4 = (ac, bc, r_2), u_5 = (bc, a, l_2), u_6 = (a, bbcc, r_1)\}$.

Clearly G is effective and cyclefree, and $\Gamma(G)$ is a partial ordering on J. Its minimal elements are $\{ab, abc\}$, and its maximal elements are $\{bbcc, abc\}$. View $\Gamma(G)$ as the relation graph of $\geqslant_\sigma^*$ on Σ. Then $\sigma_i \in \Sigma_h \cup \Sigma_a$ is a minimal element under $\geqslant_\sigma^*$ if and only if σ_i is the host string of some minimal element $u_j \in J$ under $\geqslant^*$; and $\sigma_i \in \Sigma_h \cup \Sigma_a$ is a maximal element under $\geqslant_\sigma^*$ if and only if σ_i is the adjunct string of some maximal element $u_j \in J$ under $\geqslant^*$. It is possible to define inductively the *regular expression* $\rho(\sigma_i)$ *associated with a note* σ_i *of* $\Gamma(G)$ in a manner similar to that of Section 2.2.

(1) If σ_i is a maximal node, then $\rho(\sigma_i) = \sigma_i$.

(2) For a node σ_i, assume that each node $\sigma_k{}^i$ in $\{\sigma_k{}^i \mid \sigma_k{}^i >_\sigma \sigma_i\}$, i.e., a predecessor, has its associated regular expression $\rho(\sigma_k{}^i)$ defined. Order all branch designations $X = \{\xi_k \mid (\sigma_i, \sigma_k{}^i, \xi_k) \in J\}$ such that $\xi_s < \xi_t$ if either $s < t$, or $s = t$ and $\xi_s = l_s$ and $\xi_t = r_s$. There may be a set of branch designations which are identical, and if such is the case, they should be represented as one but be separately accounted for. Let the ordering of X be $(\xi_1, \xi_2, \ldots, \xi_n)$, where only distinct elements are shown. Take σ_i and factor it into $\sigma_i = \sigma_{i_1}\sigma_{i_2} \cdots \sigma_{i_{n+1}}$ such that for all k, $1 \leqslant k \leqslant n$, if $\xi_k = l_{n_k}$ then $\lg(\sigma_{i_1}\sigma_{i_2} \cdots \sigma_{i_k}) = n_k - 1$, but if $\xi_k = r_{n_k}$ then $\lg(\sigma_{i_1}\sigma_{i_2} \cdots \sigma_{i_k}) = n_k$, where lg denotes the length. Note that some of σ_{i_k} may be null, i.e., $\sigma_{i_k} = \epsilon$. The associated regular expression $\rho(\sigma_i)$ of σ_i is now defined by

$$\rho(\sigma_i) = \sigma_{i_1}\mu_{i_1}\sigma_{i_2}\mu_{i_2} \cdots \mu_{i_n}\sigma_{i_{n+1}},$$

where $\mu_{i_k} = (\rho(\sigma_k{}^i))^*$ if there is only one branch which has ξ_k as its label, and $\mu_{i_k} = [\rho(\sigma_{k_1}^i) \vee \cdots \vee \rho(\sigma_{k_r}^i)]^*$ if there are r incident branches with the same ξ_k and $\rho(\sigma_{k_1}^i), \ldots, \rho(\sigma_{k_r}^i)$ are the associated regular expressions of the nodes from which branches with ξ_k are incident upon node σ_i.

Note that if σ_i is a maximal node, then there are no predecessors to σ_i, and case (1) above is a special case of case (2).

(3) Since $\Gamma(G)$ is partially ordered, there always exists a set of maximal nodes, and (1) above can be applied. Furthermore, it is easy to see that, unless all nodes are labeled with associated regular expressions, there is at least

one node to which (2) above can be applied. Hence the procedure will associate regular expressions to all nodes of $\Gamma(G)$, and terminate. Given an effective (but not necessarily cyclefree) LAG $G = (\Sigma_c, J)$ and $\sigma_i \in \Sigma = \Sigma_c \cup \Sigma_h \cup \Sigma_a$, define $G(\sigma_i) = (\{\sigma_i\}, J)$.

LEMMA 2.4.1. *Regular expression $\rho(\sigma_i)$ associated with node σ_i of cyclefree $\Gamma(G)$ denotes the language generated by $G(\sigma_i)$.*

Proof. If σ_i is a maximal element of $\Gamma(G)$, be it in $\Sigma_a - \Sigma_h$, or in $\Sigma_c - \Sigma_h$, the assertion is obviously true.

Assume the assertion is true for all predecessors $\sigma_1^i, \sigma_2^i, \ldots, \sigma_n^i$. Since $\Gamma(G)$ is partially ordered, $\rho(\sigma_1^i), \rho(\sigma_2^i), \ldots, \rho(\sigma_n^i)$ are the only possible adjunctions to σ_i. If the adjunction point ξ has a unique rule $(\sigma_i, \sigma_k^i, \xi)$ to apply, then the adjunction string $(\rho(\sigma_k^i))^*$ is inserted at ξ. If, on the other hand, there is more than one rule which applies at ξ, then $(\rho(\sigma_{k_1}^i) \vee \rho(\sigma_{k_2}^i) \vee \cdots \vee \rho(\sigma_{k_r}^i))^*$ is adjoined, because this allows the adjunction in any order. ∎

Given an effective (but not necessarily cyclefree) LAG $G = (\Sigma_c, J)$ and $\sigma_i \in \Sigma = \Sigma_c \cup \Sigma_h \cup \Sigma_a$, define $G(\sigma_i) = (\{\sigma_i\}, J)$ as in Section 2.2. Then we have

THEOREM 2.4.1. *A regular expression $\rho(\sigma_i)$ associated with node σ_i of cyclefree $\Gamma(G)$ denotes the language generated by $G(\sigma_i)$. The language of effective and cyclefree* LAG (Σ_c, J) *is a regular language which is denoted by the regular expression*

$$\rho(\Sigma_c, J) = \bigvee_{\sigma_i \in \Sigma_c} \rho(\sigma_i).$$

EXAMPLE 2.4.2. In Example 2.4.1, $\rho(\Sigma_c, J) = \rho(abc) \vee \rho(ab)$, which can be shown to be equal to

$$abc \vee a((b(a(bbcc)^*)^* c)^* ac(b(a(bbcc)^*)^* c)^* \vee a(bbcc)^*)^* b.$$

3. STRING ADJUNCT GRAMMARS WITH NULL SYMBOLS

3.1. *String Adjunct Grammars with Null Symbols (AGN)*

We will now introduce a somewhat modified form of AG's (LAG's or DAG's) called string adjuncts grammars with null symbols (AGN). The modification consists of allowing in the alphabet a very special type of "nonterminal" symbols called *null symbols*. The main idea is to use the null

symbols to tag the strings in Σ. The null symbols have no points of adjunction associated with them and they do not receive any adjuncts. The null symbols are ultimately erased (i.e, rewritten as a null string ϵ). [For linguistic relevance see Section 6 and for some recent results concerning LAGN's see Levy (1972) and Hart (1972).]

DEFINITION 3.1.1. An LAGN (or DAGN) G is a 7-tuple $(A, N, \Sigma, \Sigma_c, \Sigma_h, \Sigma_a, J)$, where A is a finite alphabet, N is a finite set (possibly empty) of null symbols, Σ is a finite set of basic strings, $\Sigma_c \subset \Sigma$ is the set of basic center strings, Σ_h is the set of basic host strings and Σ_a is the set of basic adjunct strings, $\Sigma = \Sigma_c \cup \Sigma_h \cup \Sigma_a$ and J is a finite set of adjunction rules. Further,

(a) $A \cap N = \varnothing$;

(b) If $\sigma \in \Sigma$ then $\sigma \in (A \cup N)(A \cup N)^*$;

(c) There is no rule in J which adjoints adjuncts to the left or right of a null symbol, i.e., null symbols have no points of adjunction. Thus for a $\sigma_i \in \Sigma$ the adjunction vectors are the same as those that can be defined for the same σ_i without the null symbols, i.e., as far as adjunctions are concerned we ignore the null symbols.

We will use Greek letters for the null symbols, and unless otherwise necessary, we will write an LAGN (or DAGN) G as just the pair (Σ_c, J). Clearly, an LAGN (or DAGN) G is an LAG (or DAG) if $N = \varnothing$.

3.2. *String Adjunct Languages with Null Symbols (LALN)*

$\hat{\Sigma}$, and $\hat{\Sigma}(\Sigma_c)$ can be defined in exactly the same manner as for LAL (or DAL) (Joshi *et al.*, 1972). We now define the language corresponding to an LAGN (or DAGN).

DEFINITION 3.2.1. Let $G = (\Sigma_c, J)$ be an LAGN (or DAGN). Then the corresponding language LALN (or DALN) $L(G)$ is

$$L(G) = h(H(\hat{\Sigma}(\Sigma_c))),$$

where H is the homomorphism defined in Sections 2.2 and 3.2 of Joshi *et al.* (1972) and h is the homomorphism. $h(\alpha_i) = \epsilon$, $\alpha_i \in N$, $h(a_i) = a_i$, $a_i \in A$, and h is extended to strings on $N \cup A$, i.e., h erases the null symbols.

Remarks. 1. An LAL is an LALN and a DAL is a DALN. Thus, clearly, the class of LAL's is contained in the class of LALN's, and the class of DAL's

is contained in the class of DALN's. Recently, Levy (1972) has shown that the class of LAL's is *properly* contained in the class of LALN's.

2. Let $G = (\Sigma_c, J)$ be an LAGN (or DAGN) such that for every σ_i, $\sigma_j \in \Sigma$, $(\sigma_i \neq \sigma_j)$, $h(\sigma_i) \neq h(\sigma_j)$. There is an LAG (or DAG) G' such that $L(G) = L(G')$.

3. Let $G = (\Sigma_c, J)$ be an LAGN (or DAGN). G is a *normal* LAGN (DAGN) if for every $\sigma_i \in \Sigma$, either σ_i does not contain any null symbols or it contains exactly one null symbol and it is in the initial position of σ_i, i.e., at most one null symbol is used as a prefix.

4. For every LAGN (or DAGN) G there is a normal LAGN (or DAGN) G' such that $L(G) = L(G')$.

EXAMPLE 3.2.1. Let $L = (a(ab)^* b)^+$. Let $G = (\Sigma_c, J)$ where $\Sigma_c = \{\alpha ab\}$, and $J = \{u_1 = (\alpha ab, \alpha ab, r_2); u_2 = (\alpha ab, \beta ab, r_1)\}$. Here $A = \{a, b\}$, $N = \{\alpha, \beta\}$, $\Sigma = \{\alpha ab, \beta ab\}$, $\Sigma_h = \{\alpha ab\}$, and $\Sigma_a = \{\alpha ab, \beta ab\}$. Then $L(G) = L$. Note that in u_1, r_2 is the point of adjunction to the second symbol in αab, ignoring the null symbol α. Similarly, in u_2, r_1 is the point of adjunction to the right of the first symbol in αab, again ignoring the null symbol α.

There is an equivalent LAG for this L [see Example 2.2.4 *in* (Joshi *et al.* (1972)] which is somewhat more complicated as compared to the LAGN above. This is because in an LAG (or DAG) we cannot use the same string to play two different "roles."

EXAMPLE 3.2.2. Let $L = a(pq)^* bc \vee a(rs)^* bc$. Let $G = (\Sigma_c, J)$, where $\Sigma_c = \{\alpha abc, \beta abc\}$, $J = \{u_1 = (\alpha abc, pq, r_1); u_2 = (\beta abc, rs, r_1)\}$. Then $L(G) = L$. Here also there is an equivalent LAG for this L [see Example 5.1.1 *in* (Joshi *et al.* (1972)].

All the results concerning LAL's (or DAL's) can be easily extended to LALN's (or DALN's). We have, however, the following additional results.

LEMMA 3.2.1. *For every* LAGN (*or* DAGN) G *there is an equivalent* LAGN (*or* DAGN) G' *such that* Σ_c', *the set of basic center strings of* G', *and* Σ_a', *the set of basic adjunct strings of* G', *are disjoint.* (*We will call a grammar such as* G' *a* center-adjunct disjoint *grammar.*)

The following example will illustrate the main idea in the proof of Lemma 3.2.1.

EXAMPLE 3.2.3. Let $G = (\Sigma_c, J)$ be the LAGN in Example 3.2.1. G is not center-adjunct disjoint. Let $G' = (\Sigma_c', J')$ be an LAGN where

$\Sigma_c' = \{\gamma ab\}$, and $J' = \{u_1 = (\gamma ab, \delta ab, r_2), u_2 = (\delta ab, \mu ab, r_1), u_3 = (\gamma ab, \mu ab, r_1)\}$. Here $A' = \{a, b\} = A$, $N' = \{\gamma, \delta, \mu\}$, $\Sigma' = \{\gamma ab, \delta ab, \mu ab\}$, $\Sigma_h' = \{\gamma ab, \delta ab\}$, and $\Sigma_a' = \{\delta ab, \mu ab\}$. G' is center-adjunct disjoint and $L(G') = L(G)$.

THEOREM 3.2.1. *The class of* LALN's (*or* DALN's) *is closed under the operations* $\cup$, $\cdot$, *and* $*$. (*Closure properties of some subclasses of* AL's *are summarized in Fig.* 5.)

Proof. Since identical strings on the alphabet A playing different "roles" can be tagged by different null symbols, the interactions among rules can be avoided. Closure under $\cup$, and $\cdot$ can then be easily established. Using Lemma 3.2.1, closure under $*$ can also be easily established. (ϵ must be added to Σ_c, if necessary.) ∎

THEOREM 3.2.2. *Every regular language* (*regular set*) *is an* LAGN.

Operation / Language	Union $\cup$	Set product $\cdot$	Kleene closure $*$	Intersection $\cap$	Complementation $\neg$
LAL				No	
LALN	Yes	Yes	Yes	No	No
DAL				No	No
DALN	Yes	Yes	Yes	No	No

FIG. 5. Closure properties.

3.3. *Cyclefree LAGN's and Regular Sets*

LEMMA 3.3.1. *If G is a cyclefree* LAGN *then L(G) is regular.*
(A trivial modification of the proof of Theorem 2.4.1 establishes this lemma.)

Let G_1 and G_2 be two cyclefree LAGN's. It is easy to see that cyclefree LAGN's, $\bar{G}$ and $\bar{\bar{G}}$ can be constructed such that $L(\bar{G}) = L(G_1) \cup L(G_2)$ and $L(\bar{\bar{G}}) = L(G_1) \cdot L(G_2)$. Now let $G = (\Sigma_c, J)$ be a cyclefree LAGN and we can assume that G is a center-adjunct disjoint grammar without any loss of generality. Let $G' = (\Sigma_c', J')$, where $\Sigma_c' = \Sigma_c \cup \{\epsilon\}$, and $J' = J \cup J_c$, where $J_c = \{(\sigma_i, \sigma_j, r_{|\sigma_i|}) \mid \sigma_i, \sigma_j \in \Sigma_c, \sigma_i \neq \epsilon, \sigma_j \neq \epsilon\}$. Then $L(G') = (L(G))^*$. G' is not a cyclefree LAGN. However, it has a very special type of cycles. Every cycle in G' consists of a sequence of rules such that each rule is right-concatenative [a rule $(\sigma_i, \sigma_j, \xi_k)$ is right-concatenative if $\xi_k = r_{|\sigma_i|}$,

where $|\sigma_i|$ is the length of σ_i]; further, for every rule $u_i \in J_c$ either u_i is a cycle (of length one) or there is a $u_j \in J_c$ such that $u_i u_j$ is a cycle (of length two). Note also that since G is center-adjunct disjoint, none of the strings in the rules in J_c appear as adjunct strings in the rules in J. Now, it is possible to construct a cyclefree LAGN equivalent to a grammar such as G' above. Rather than proving this in detail, we will give an example to illustrate the main idea of the proof, which essentially consists of using the technique described in Example 5.1.1 in Joshi *et al.* (1972).

EXAMPLE 3.3.1. Let G be the LAGN in Example 3.2.3. $\Sigma_c = \{\gamma ab\}$, $J = \{u_1 = (\gamma ab, \delta ab, r_2),\ u_2 = (\delta ab, \mu ab, r_1),\ u_3 = (\gamma ab, \mu ab, r_1)\}$. G is center-adjunct disjoint and also cyclefree. Let $G' = (\Sigma_c', J')$. where $\Sigma_c' = \{\gamma ab\} \cup \{\epsilon\}$ and $J' = J \cup J_c$, where $J_c = \{(\gamma ab, \gamma ab, r_2)\}$. Then $L(G') = (L(G))^*$. Now a cyclefree LAGN equivalent to G' can be constructed by using the technique referred to above and introducing new null symbols if necessary. Let $G'' = (\Sigma_c'', J'')$ be a LAGN, where

$$\Sigma_c'' = \Sigma_c' \cup \{\pi abab\} = \{\gamma ab, \pi abab, \epsilon\},$$

and

$$J'' = J \cup \{(\pi abab, \gamma ab, r_4), (\pi abab, \mu ab, r_1), (\pi abab, \mu ab, r_3)\}.$$

Here $A'' = \{a, b\}$, $N'' = \{\gamma, \delta, \mu, \pi\}$, $\Sigma_h'' = \{\gamma ab, \delta ab, \pi abab\}$, $\Sigma_a'' = \{\delta ab, \mu ab, \gamma ab\}$, and $\Sigma'' = \{\gamma ab, \delta ab, \pi abab, \mu ab, \epsilon\}$. G'' is cyclefree and it can be easily seen that $L(G') = L(G'')$.

Hence, we have the following *exact characterization of regular sets in terms of the class of cyclefree LAGN's.*

THEOREM 3.3.1. *$L \subset A^*$ is a regular set if and only if there is a cyclefree* LAGN *G such that $L(G) = L$.*

3.4. *Bounded Semilinear Sets, DAGN's and DAG's*

Linear and semilinear sets were considered by Parikh (1961). His results can be easily extended to DAL's, i.e., if L is a DAL then $\psi(L)$, the commutative immage of L, is a semilinear set. We will now establish some results concerning bounded semilinear sets.

DEFINITION 3.4.1. A linear set $L(c; p_1, p_2, \ldots, p_m)$ is said to be a *constant dominated linear set* if and only if for every i and j, if the j-th component of p_i is nonzero then the j-th component of c is nonzero.

DEFINITION 3.4.2. A set L is said to be a constant dominated semilinear set if it is a finite union of constant dominated linear sets.

LEMMA 3.4.1. *For every semilinear set an equivalent constant dominated semilinear set can be effectively found.*

Proof. It is sufficient to consider the reduction of a linear set to an equivalent constant dominated semilinear set. Let $L = L(c; p_i, p_2, \ldots, p_m)$ be a linear set. Define

$$L_{i_1 i_2 \cdots i_m} = L(c + i_1 p_1 + i_2 p_2 + \cdots + i_m p_m ; i_1 p_1, i_2 p_2, \ldots, i_m p_m),$$

where each i_j can take the values 0 or 1. Clearly,

$$L = \bigcup_{i_1, i_2, \ldots, i_n = 0,1} L_{i_1 i_2 \cdots i_m} \quad \blacksquare$$

THEOREM 3.4.1. *For every bounded semilinear language L a* DAGN G *such that $L(G) = L$, and $\Sigma_h \cap \Sigma_a = \varnothing$, i.e., the hosts and adjuncts are disjoint, can be effectively found. [L is bounded semilinear language if L is a bounded language (Ginsburg, 1966) and $\psi(L)$ is a semilinear set. A semilinear set is bounded if it is the commutative image of a bounded language.]*

Proof. It is sufficient to consider a constant dominated language L. Since L is bounded, $L \subset w_1^* \cdots w_n^*$ for some $w_1, w_2, \ldots, w_n$, $w_i \in A^*$, $i = 1, 2, \ldots, n$.

Now consider the constant dominated linear set

$$L_1 = L((c_1, c_2, \ldots, c_n); (p_{11}, p_{12}, \ldots, p_{1n}), (p_{21}, p_{22}, \ldots, p_{2n}), \ldots, (p_{m1}, p_{m2}, \ldots, p_{mn}))$$

and let $L = \psi^{-1}(L_1)$. Let $G = (\Sigma_c, J)$ be a DAGN such that

$$\Sigma_c = \{\alpha w_1^{c_1} w_2^{c_2} \cdots w_n^{c_n}\},$$

and

$$\begin{aligned} J = \{&(\alpha w_1^{c_1} w_2^{c_2} \cdots w_n^{c_n}, (\beta_1 w_1^{p_{11}})(w_2^{p_{12}}) \cdots (w_n^{p_{1n}}), \\ &r_{|w_1|} r_{|w_1|+|w_2|} \cdots r_{|w_1|+|w_2|+\cdots|w_n|}), (\alpha w_1^{c_1} w_2^{c_2} \cdots w_n^{c_n}, \\ &(\beta_2 w_1^{p_{21}})(w_2^{p_{22}}) \cdots (w_n^{p_{2n}}), r_{|w_1|} r_{|w_1|+|w_2|} \cdots r_{|w_1|+|w_2|+\cdots|w_n|}), \\ &\vdots \\ &(\alpha w_1^{c_1} w_2^{c_2} \cdots w_n^{c_n}, (\beta_m w_1^{p_{m1}})(w_2^{p_{m2}}) \cdots (w_n^{p_{mn}}) \\ &r_{|w_1|} r_{|w_1|+|w_2|} \cdots r_{|w_1|+|w_2|+\cdots+|w_n|})\}. \end{aligned}$$

The case when some c_i's are 0 can be handled in an obvious way. The hosts and adjuncts are obviously disjoint. It is easily seen that $L(G) = L$. Since DALN's are closed under finite union, the above construction can be extended to constant dominated semilinear sets. ▮

EXAMPLE 3.4.2. Let $L_1 = L((1, 1, 1); (1, 1, 2), (0, 1, 1))$, $L \subset (ab)^* a^*$, and $L = \psi^{-1}(L_1)$. Let $G = (\Sigma_c, J)$ be a DAGN such that $\Sigma_c = \{\alpha abab\}$ and $J = \{(\alpha abab, (\beta_1 ab)(a)(b), r_2 r_3 r_4), (\alpha abab, (\beta_2 ab)(a)(b), r_2 r_3 r_4)\}$.

It is also possible to show that in Theorem 3.4.1 we can replace the DAGN by a DAG, i.e., by a grammar without the null symbols (we will omit the proof here).

4. COMPARISON OF AG'S WITH PSG'S

In the earlier sections we have seen several results connecting AG's with PSG's. We will now state a few additional points of comparison.[1]

4.1. *Terminal and Nonterminal Symbols*

In an AG the alphabet A corresponds to the terminal alphabet V_T of a PSG. In an AG we do not have nonterminals in the sense of the nonterminal alphabet of a PSG. We have, however, auxiliary symbols used implicitly such as the ξ_k's corresponding to the points of adjunctions. But these auxiliary symbols are used purely as position markers and do not have the same interpretation as the nonterminals in a PSG (i.e., the auxiliary symbols ξ_k's do not correspond to phrase types). If we consider the marking symbol, $\hat{}$, used in the recursive definition of $\hat{\Sigma}$ also as an auxiliary symbol, then one can possibly consider $\hat{a}_i$ $(a_i \in A)$ as a nonterminal which can be interpreted as a phrase type but with *the added interpretation* that a phrase type $\hat{a}_i$ has a_i as the "head" (or "center") of the phrase. A phrase type in a PSG does not necessarily have such an interpretation. Further the notion of the "head" of a phrase type cannot be naturally formulated in a PSG.

In an LAGN (or DAGN) the null symbols are, however, like the nonterminals in a PSG although highly restricted. The null symbols are used to

[1] Recently, many subclasses of PSG's have been studied, e.g., Matthews (1964), Aho (1967), Rosenkrantz (1967), Greibach and Hopcroft (1968), etc. Some of these are motivated by linguistic considerations and others by structures in programming languages. There does not seem to be a simple relationship between AG's and these classes.

tag basic strings and therefore they are not used as position markers; in fact, they have no positional interpretation. The null symbols as nonterminals are highly restricted because they are never "rewritten" (in the sense of a PSG) into any other string except the null string. Recently, Levy (1972) has shown that the class of LALN's *properly* contains the class of LAL's. All the examples of LALN's in this paper are such that equivalent LAG's can be constructed for them. However, Levy's result shows that the null symbols, in general, cannot be eliminated.

4.2. *Mixed Styles*

LAGN's (DAGN's) can be considered as grammars of a mixed style as they have both adjunction rules and a very special type of "rewrite rule," i.e., these grammars have rules of different formal character (or styles). These are, of course, very simple and rather trivial examples of mixed grammars. More interesting classes of *mixed grammars* have been studied by Joshi (1969). In particular, consider a mixed grammar, $G = (\Sigma_c, J, R)$, where Σ_c is the finite set of basic center strings; Σ and Σ_c are strings in $(A \cup \{S\})^*$, A is the alphabet as before and S is a single "nonterminal" in the sense of a PSG, J is a finite set of adjunction rules (local or distributed), and R is a finite set of "rewrite rules" of the form $\langle \sigma_i, \sigma_j \rangle$, $\sigma_i \in \Sigma$, and σ_i contains at least one S. The meaning of a rule, $\langle \sigma_i, \sigma_j \rangle$ is that from σ_i one can derive a string by replacing some occurrence of S in σ_i by σ_j (hence these rules will be called "replacement rules" for convenience). A study of the properties of such grammars (and their generalizations) and their use in the construction of transformational grammars has been carried out by Joshi (1969, 1972) and Levy (1970) (see also Section 6).

4.3. *Basic Center Strings and the Initial Symbol*

It follows from Section 4.1 that in an AG we do not have a symbol corresponding to the initial symbol S in a PSG. The basic center strings can be considered somewhat analogous to the symbol S. But there is an important distinction. In generating a string, say w, in an AG we do not start from a certain center string but rather start from the "innermost" adjunct string (strings) and generate the string w from "inside out", and finally use the center string. Thus a generation in an AG can be considered as "inside out". The "inside out" generation turns out to be a decided advantage in constructing transformational grammars based on these grammars (Joshi, 1969, 1972).

4.4. *Elementary Sentences (Elementary Sentence Forms)*

In an AG G the set of strings in Σ_e can be considered as elementary sentences (in the linguistic context symbols in A are category symbols and hence strings in Σ_e are elementary sentence forms and not elementary sentences), and for every $w \in L(G)$ there is an elementary sentence underlying it (for different readings of an ambiguous w the underlying elementary sentences may or may not be different).

4.5. *Hierarchies of Languages*

Various results in earlier sections show that the hierarchy due to AG's cuts across the hierarchy due to the PSG's in many interesting ways (although there are many open problems). Figure 6 summarizes the results concerning

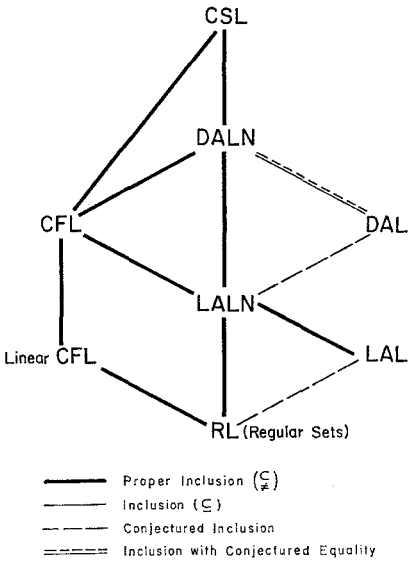

FIG. 6. Hierarchy of certain subclasses of AL's in relation to the phrase structure hierarchy.

the hierarchy of certain subclasses of AL's and also its relation to the phrase structure hierarchy. The proper containment of CFL's in DALN's was recently shown by Levy (1971).

5. Special Cases of AG's

5.1. *Repeatable and Nonrepeatable Adjunction Rules*

In this section, we will point out some special cases of AG'S, all of which are linguistically motivated (see Section 6).

Let G be an AG. If $u_r = (\sigma_i, (\sigma_j), \xi_k)$ is a rule in J then u_r can be applied to σ_i, or to a σ_i-derived string, arbitrarily many times in a derivation in G. Of course, u_r need not be applied at all. We will say that u_r is a *repeatable* adjunction rule. In this sense, all the adjunction rules considered so far are repeatable adjunction rules.

We now consider adjunction rules which are *nonrepeatable* in the sense that such a rule, say, u_r would be applied no more than once (i.e., zero or one time) to each occurrence of its host string in a derivation in G. If u_r is a nonrepeatable rule (nr rule) we will write it as $u_r = (\sigma_i, (\sigma_j), \xi_k)^1$. In general we can consider a k-repeatable rule, $k = 1, 2, \ldots$. A k-repeatable rule could be applied no more than k times to each occurrence of its host string. A nonrepeatable rule is then a 1-repeatable rule.

Definition 5.1.1. An AG G will be called an nr-AG if and only if J contains at least one nr rule.

We will not give here a recursive definition of $\hat{\Sigma}$ or $\Sigma(\hat{\Sigma}_c)$ for an nr-AG. The definitions in Sections 2.2 and 3.2 of Joshi *et al.* (1972) can be extended roughly as follows. For a $\sigma_i \in \Sigma$, let $A_{\sigma_i} = \{u_k\}$, $k = 1, 2, \ldots, p$ be the set of nr rules such that σ_i is a host in each rule in A_{σ_i}. Now we attach p distinct markers to σ_i, one for each rule in A_{σ_i}. We erase the k-th marker if rule u_k is applied in the derivation of a σ_i-derived string. Further, a rule u_k cannot be applied to a σ_i-derived string if the k-th marker has already been erased. Thus we can define the corresponding languages nr-LAL or nr-DAL, etc.

Note that if G is an nr-AG and if $u_r = (\sigma_i, (\sigma_j), \xi_k)^1$ is an nr rule then for any r–l tree representation of a string $W \in L(G)$, each node labeled σ_i has no more than one branch labeled $u_r : \xi_k$ incident on it.

Example 5.1.1. Let $G = (\Sigma_c, J)$, where

$$\Sigma_c = \{ab\}, \qquad \text{and} \qquad J = \{u_1 = (ab, ab, r_1)^1\}.$$

Thus G is an nr-LAG because u_1 is an nr rule. This is the same grammar as in Example 2.2.3 of Joshi *et al.* (1972) except that u_1 is now an nr rule. It is easily seen that $L(G) = \{a^n b^n \mid n \geqslant 1\}$ which is not the same language as in Example 2.2.3. Note that $L(G) = \{a^n b^n \mid n \geqslant 1\}$ is a DAL and not an LAL.

It is, however, an nr-LAL. Thus the class of LAL's $\subset$ the class of nr-LAL's. $L_1 = \{a^m c b^m \mid m \geqslant 1\}, L_2 = \{a^m b^m a^n b^n \mid m \geqslant 1, n \geqslant 1\}$ are nr-LAL's. They are not LAL's.

5.2. *AG's with $\Sigma = \Sigma_c$*

We will now consider AG's with the restriction that $\Sigma = \Sigma_c$, i.e., every basic string is also a basic center string; thus every basic adjunct string is also a basic center string. Let $L_1 = 1\,0^*\,1$. Clearly, L_1 has an LAG $G = (\Sigma_c, J)$, where $\Sigma_c = \{1\,1\}$, and $J = \{u_1 = (1\,1, 0, r_1)\}$. Here $\Sigma = \{1\,1, 0\} \neq \Sigma_c$. Of course, if we add 0 to Σ_c then we will have an LAG with the condition that $\Sigma = \Sigma_c$.

The main interest for considering AG's with $\Sigma = \Sigma_c$ is as follows. Strings in Σ_c can be considered as elementary sentences (or sentence forms) in $L(G)$. If $\Sigma = \Sigma_c$ then every string in $L(G)$ can be decomposed into a set of elementary sentences (or sentence forms).

5.3. *AG's with $\Sigma_c \cap \Sigma_a = \varnothing$*

The condition $\Sigma_c \cap \Sigma_a = \varnothing$ means that no basic adjunct string is also a basic center string. We call an AG with $\Sigma_c \cap \Sigma_a = \varnothing$, a *center-adjunct disjoint* grammar (see Section 6). Obviously, if we have an AG with $\Sigma = \Sigma_c$ as in Section 5.2 then $\Sigma_c \cap \Sigma_a \neq \varnothing$; in fact, $\Sigma_a \subset \Sigma_c$.

It is not known whether for every LAL (or DAL) L there is a LAG (or DAG) G with $\Sigma_c \cap \Sigma_a = \varnothing$ such that $L(G) = L$.

5.4. *LAG's with a Uniform Set of Rules*

DEFINITION 5.4.1. Let $G = (\Sigma_c, J)$ be an LAG. A rule $(\sigma_i, \sigma_j, \xi_k)$ in J, $\sigma_i = a_{i_1} a_{i_2} \cdots a_{i_k} \cdots a_{i_m}$; $a_{i_p} \in A$; $p = 1, 2, \ldots, m$, is called *uniform* if and only if for every $\sigma_t = a_{t_1} a_{t_2} \cdots a_{t_s} \cdots a_{t_n}$; $a_{t_q} \in A$; $q = 1, 2, \ldots, n$; $\sigma_t \in \Sigma$, and $a_{i_k} = a_{t_s}$ for some s, there is a rule $(\sigma_t, \sigma_j, \xi_l)$ such that $(\xi_k = r_k \Leftrightarrow \xi_l = r_l)$ or $(\xi_k = l_k \Leftrightarrow \xi_l = l_l)$.

In other words, if in a rule σ_j adjoins to the left (or right) of some symbol $a_g \in A$ in the host, then there is, for every occurrence of a_g in any string $\sigma_t \in \Sigma$, an adjunction rule which adjoins σ_j to the left (or right) of a_g (for each occurrence of a_g in σ_t). That is, σ_j adjoins to the left (or right) of a_g wherever a_g occurs in any string in Σ. We can, therefore, say that σ_j is a left (or right) adjunct of a_g.

EXAMPLE 5.4.1. $G = (\Sigma_c, J)$, where $\Sigma_c = \{abc, ba\}$, and

$$J = \{u_1 = (abc, pq, r_1), u_2 = (ba, pq, r_2), u_3 = (abc, t, r_2)\}.$$

Here u_1 and u_2 are uniform and pq is an adjunct of a. u_3 is not uniform as t adjoins on the right of b only in the host abc and not in the host ba.

DEFINITION 5.4.2. Let $G = (\Sigma_c, J)$ be an LAG. Then G is a *uniform* LAG if and only if every rule in J is uniform.

EXAMPLE 5.4.2. If in the set of rules in the LAG G in Example 5.4.1, we add the rule $u_4 = (ba, t, r_1)$ then the resulting LAG, say G', is uniform.

If an LAG G is uniform then we can write $G = (\Sigma_c, J)$ where the rules have the form $u = [a_i, \sigma_j, \xi]$, $a_i \in A$, $\sigma_j \in \Sigma$, $\xi = l$ for a left adjunction rule and $\xi = r$ for a right adjunction rule. Adjunction thus becomes a property of a symbol in A independently of its being part of a basic string (strings). Thus G' in Example 5.4.2 can be written as $G = (\Sigma_c, J)$, where Σ_c is as in Example 5.4.1 and $J = \{u_1 = [a, pq, r], u_2 = [b, t, r]\}$. The above concepts of uniform rules and uniform grammars can be obviously extended to LAGN's but they cannot be naturally extended to DAG's.

6. LINGUISTIC RELEVANCE

Here we will discuss very briefly the relevance of AG's to language structure.

1. In the linguistic context the alphabet A in an AG G will consist of symbols which denote major dictionary classes (lexical classes) such as N (nouns), t (tense, auxiliaries), A (adjectives), V (verbs), P (prepositions), wh (who, which, whom), D (adverbs), Q (quantifiers), etc. N, t, A, V, etc., are thus preterminal symbols. The basic center strings thus correspond to basic (elementary) sentence forms, e.g., N t V (*John came*), N t V N (*Jim bought books*), N t V P N (*people rely on John*), etc. (A subcategorization of V's is implied here and is not explicitly shown.) Basic adjunct strings are basic adjunct forms, e.g., P N (*from Philadelphia*), A (*old*), wh N t V (*whom John saw*), wh t V N (*who saw Jim*), D (*quickly*), etc. Each derived string in $L(G)$ is thus a derived sentence form, e.g., (assuming suitable adjunction rules), N P N t V N (*a man from Philadelphia bought books*), A N t V (*an old man came*), N wh N t V t V D (*the man whom Bill saw ran quickly*), N wh N wh t

V N t V t V D (*the books (which) the man who met Jim bought will arrive soon*), etc. (ignoring articles for simplicity).

2. Lexical insertion is considered a separate activity. The reasons for this separation are well known (Chomsky, 1965). We are not interested here in a detailed description or formalization of this activity. However, the following is an important consideration. In an AG, lexical insertion takes place as each basic string is brought into the generation of a sentence. Let

$$\sigma_i = a_{i_1} a_{i_2} \cdots a_{i_m} ; a_{i_j} \in A$$

be a basic string. As σ_i is brought into the generation of a sentence, for each a_{i_j} a lexical item can be inserted immediately. Note that we are not replacing a_{i_j} by the lexical item but rather attaching it to a_{i_j} and it will be carried along with a_{i_j} as the derivation continues. The verification of restrictions (e.g., number and person agreement: *John sleeps here, the boys sleep here, I work in the morning, John works in the morning*; some verbs take human subjects: *try*; some verbs may not take abstract subjects: *eat*, etc.) that hold within the domain of a basic string can be immediately carried out as any pair of symbols of σ_i are at a bounded distance at this state. If the basic strings are properly chosen then most restrictions are brought to bear within the domain of some basic string, and indeed it turns out that basic strings (with reasonable linguistic interpretations) can be set up in this way.

There are some restrictions which hold between a host and an adjunct string; e.g., in *N wh N t V t V* (*the man whom John met arrived*), *wh N t V* is an adjunct of *N t V* and the *N* in *N t V* is really the "object" of *V* in *wh N t V*. Some other examples are: Zeroing in conjoined sentences, e.g., *everyday, he runs* **and swims**; *he played tennis* **but she didn't**, etc. Restrictions between successive adjuncts at the same point of adjunction of the host (ordering restrictions) as in *I am looking for a book* **with a green cover which was lying here somewhere.** Restrictions between a host and two or more adjuncts at different points of adjunction of the host as in *boys* **who can swim** *distrust boys* **who can't.** All these can be easily verified.

The important point to note is that in an AG, lexical insertion takes place each time a new basic string is brought into the generation, i.e., it takes place as the generation proceeds string by string. In a PSG, lexical insertion takes place at the very end of generation and the entire process of lexical insertion together with the verification of restrictions becomes more complicated.

3. The relevance of AG-type grammars is due to the fact that most constituents (phrases in a PSG) either consist of a single word (of some category)

or contain a single word of the characterizing category plus adjunct words or strings of words adjoined to it. Such a constituent can be considered as "endocentric" (with the characterizing category as the "center" or "head"), e.g., in *books from the library*, *books* is the center. *Books from the library* is related to *books* as a constituent expansion. Thus for any sentence S_1 which can be represented as a sequence of immediate constituents each of which is endocentric we can obtain a sentence S_2 by replacing each constituent by its center. Then S_2 is the center of S_1 and S_1 is a constituent expansion of S_2, e.g., S_1 = *young boys from New York came* and S_2 = *boys came.*

AG's are well suited for formulating the "endocentric" properties in the sense that this aspect of a constituent can be explicitly brought out in the structural description. There are, however, constituents which are not "endocentric". These are "exocentric" in the sense that we cannot replace them by any word of a characterizing category contained in them such that the constituents can be considered as constituent expansions of the characterizing category; e.g., *whether he came* in *I don't know whether he came, who will represent us at the meeting* in *who will represent us at the meeting is unclear*, etc. AG's are not well suited for formulating the exocentric properties. These properties are better characterized by the use of "nonterminals" and "rewrite rules" in the sense of a PSG. Thus rules of different formal character bring out different aspects of language structure. Sentence adjuncts (e.g., *in general, probably*) can be handled well in an AG; in particular, that these adjuncts can occupy various sentence positions can be easily characterized in an AG. This is awkward to characterize in a PSG. However, the property that these adjuncts are adjuncts of a sentence is better characterized by the use of a nonterminal. This suggests that classes of formal grammars with mixed types of rules (mixed grammars) are required to bring out explicitly different aspects of language structure (Joshi, 1969, 1972); Levy, 1970) (see Section 4.2).

4. Distributed adjunction rules are required to handle cases such as *two and three are even and odd numbers, respectively*, which is a case of an intercalated structure. Such structures are not too frequent. However, if one tries to construct an AG-type grammar as a base for a transformational grammar then the need for intercalated structures is not so marginal. This is primarily because one tries to relate each adjunct to an elementary sentence [i.e., one tries to constitute the adjunct and host strings in such a way that the underlying elementary sentence(s) could be reconstructed from them]. Some examples are: **the** *man* **who came** ... (boldface indicates the distributed adjunct); **John's** *proof* **of the theorem**, etc. The kinds of intercalated struc-

tures possible in a DAG apparently are adequate for this purpose (Joshi, 1969, 1972).

5. Some adjuncts are nonrepeatable, e.g., Q (quantifier): *some, all,* T (articles): *the, a,* etc. Hence, the restrictions considered in Section 5.1 are relevant.

6. Many adjunction rules are almost uniform (see Section 5.4), e.g., left and right adjuncts of N, e.g., P N, A, etc., are more or less the same (with a few limitations) for every occurrence of N in the set of all basic strings. Hence, one can consider these as adjuncts of N without reference to the basic strings in which N occurs. Other examples are: adjuncts of V, adjuncts of A, etc. However, when one tries to construct an AG as a base for a transformational grammar the adjunction rules become far less uniform.

7. The restriction $\Sigma_c \cap \Sigma_a = \varnothing$ in Section 5.3 is relevant because adjuncts generally are not strings in Σ_c, e.g., P N, A, wh N t V, etc. The similarity of $N\ t\ V$ (center string): *John came* and $N\ t\ V$ (adjunct string): *John met* is only apparent. $N\ t\ V$ (adjunct string) is a variant of *wh* $N\ t\ V$: *whom John met* as in *the man John met.* Further the subclasses of V in $N\ t\ V$ (center string) and $N\ t\ V$ (adjunct string) are different.

8. As we have stated in the introduction, the classes of grammars considered in this paper have been motivated by the type of grammar proposed by Harris (1962, 1968).[2] Harris was not concerned with the study of formal properties of a class of grammars as such; however, if we examine the grammar (for English) in Harris (1962) we can observe the following: (1) Restrictions in Section 5.3 and 5.4 apply, i.e., $\Sigma_c \cap \Sigma_a = \varnothing$ and most rules are uniform. Of course, he has not stated these as restrictions on a general class of grammars. However, the fact that these conditions hold to a large extent does say something about the language structure. (2) The adjunction rules are almost all local. A few very restricted kinds of distributed rules are considered and are marginally used (3) Nonrepeatable rules have been considered. (4) Certain symbols have been used as nonterminals in the sense of a PSG.

9. If $\Sigma = \Sigma_c$ (see Section 5.2) then each string $w \in L(G)$ has a representation in terms of elementary "sentences" (or basic center strings), i.e., in the tree representation of the derivation of w, every node is either an elementary "sentence" or a derived "sentence". To what extent these are also sentences in the language and to what extent these must be considered as infrastructures is, of course, an important linguistic problem. In general, however, $\Sigma \neq \Sigma_c$.

[2] A syntactic-analysis program, incorporating a substantial part of English grammar, which is based on this type of grammar has been constructed by Sager (1967).

It is possible to construct a transformational grammar ($G_\tau = (G, \tau)$) where the base grammar G satisfies the condition $\Sigma = \Sigma_c$ (actually, a mixed grammar is needed; see Section 4.2) and τ is the transformational component which specifies a set of operations on the structures derived G. Strings in $L(G_\tau)$ (except for morphophonemic operations) are the strings (sentences) in the language. Such a class of transformational grammars has been studied from the point of view of its linguistic adequacy and its mathematical properties, e.g., conditions under which $L(G_\tau)$ is recursive (Joshi, 1972).

RECEIVED: November 2, 1970; REVISED: May 3, 1972

REFERENCES

AHO, A. V. (1967), Indexed grammars, Proceedings IEEE Eighth Annual Symposium on Switching and Automata Theory, Austin, TX.

CHOMSKY, N. (1965), "Aspects of Theory of Syntax," M.I.T. Press, Cambridge, MA.

GINSBURG, S. (1966), "Mathematical Theory of Context-Free Languages," McGraw-Hill, New York.

GREIBACH, S., AND HOPCROFT, J. (1968), Scattered context grammars, Proceedings IFIP International Conference, Edinburgh.

HARRIS, Z. S. (1962), "String Analysis of Language Structure," Mouton & Co., The Hague, Netherlands.

HARRIS, Z. S. (1968), "Mathematical Structures of Language," Vol. 28, Interscience, New York.

HART, J. M. (1972), Formal properties of local adjunct languages, Ph.D. dissertation, University of Pennsylvania, Philadelphia, PA.

JOSHI, A. K. (1969), Properties of formal grammars with mixed types of rules and their linguistic relevance, Proceedings International Symposium on Computational Linguistics, Sänga Säby, Sweden.

JOSHI, A. K. (1972), A class of transformational grammars, *in* "Formal Language Analysis" (M. Gross, M. Halle, and M. P. Schützenberger, Eds.), Mouton & Co., The Hague, Netherlands, in press.

JOSHI, A. K., KOSARAJU, S., AND YAMADA, H. M. (1972), String adjunct grammars. I. Local and distributed adjunction, *Information and Control* **21**, 93–116.

LEVY, L. S. (1970), Generalized local adjunction and replacement in adjunct languages, Ph.D. dissertation, University of Pennsylvania, Philadelphia, PA.

LEVY, L. S. (1971), Tree adjunct parenthesis, and distributed adjunct grammars, *in* "Theory of Machines and Computations" (A. Paz, Ed.), Academic Press, New York.

LEVY, L. S. (1972), Structural aspects of local adjunct languages, submitted for publication to *Information and Control.*

MATTHEWS, G. M. (1964), A note on asymmetry in phrase structure grammar, *Information and Control* **7**, 360–365.

PARIKH, R. J. (1961), Language generating devices, R.L.E. Report No. 60-1961, M.I.T., Cambridge, MA.

ROSENKRANTZ, D. J. (1967), Programmed grammars, Proceedings IEEE Eighth Annual Symposium on Switching and Automata Theory, Austin, TX.

SAGER, N. (1967), Syntactic analysis of natural languages, *in* "Advances in Computers" (M. Alt and M. Rubinoff, Eds.), Vol. 8, Academic Press, New York.