INFORMATION AND CONTROL 11, 269–289 (1967)

A Characterization of Parenthesis Languages

DONALD E. KNUTH*

California Institute of Technology, Pasadena, California 91109

A parenthesis language is a context-free language possessing a grammar in which each application of a production introduces a unique pair of parentheses, delimiting the scope of that production. Parenthesis languages are nontrivial since only one kind of parenthesis is used. In this paper it is shown that algorithms exist to determine if a context-free language is a parenthesis language, or if it is equal to the language defined by a given parenthesis grammar. A possible merit of these results lies in the fact that parenthesis languages are the most general class of languages for which such problems are now known to be solvable; in fact, other problems which are very similar to the one solved here are known to be recursively unsolvable.

1. INTRODUCTION

McNaughton (1967) has defined a special type of context-free grammer called a *parenthesis grammar* and he has shown it is possible to determine if two parenthesis grammars generate the same language. He also has raised the question whether or not it is possible to decide if a given context-free language is a parenthesis language. An affirmative answer to this question is given below, in connection with some techniques for manipulating context-free grammars which may be of independent interest. A new proof of McNaughton's theorem is also given.

Let us define a *context-free grammar* $\mathcal{G}$ as a quadruple $(\Sigma, V, \mathcal{S}, \mathcal{P})$, where the "vocabulary" V is a finite set of letters; the "terminal alphabet" Σ is a subset of V; the "start set" $\mathcal{S}$ is a finite subset of V^* (where V^* as usual denotes the set of all strings on V); and the "production set" $\mathcal{P}$, a finite set of relations of the form $A \rightarrow \theta$ where A is in the set of "nonterminal symbols" $N = V - \Sigma$, and where θ is in V^*. For any

* The preparation of this paper was supported in part by NSF grant GP-3909.

strings α, ω in V^* we write $\alpha A \omega \longrightarrow \alpha \theta \omega$, if $A \longrightarrow \theta$ is a production in $\mathcal{P}$. The transitive completion of this relation is denoted $\longrightarrow^+$, so that $\varphi \longrightarrow^+ \psi$ means there exist strings $\varphi_0, \varphi_1, \cdots, \varphi_n$ such that $\varphi = \varphi_0$, $\varphi_j \longrightarrow \varphi_{j+1}$ for $0 \leqq j < n$, and $\varphi_n = \psi$, and $n \geqq 1$. The same relation with $n \geqq 0$ is denoted by $\longrightarrow^*$, so that $\varphi \longrightarrow^* \psi$ if and only if $\varphi \longrightarrow^+ \psi$ or $\varphi = \psi$. The *language* $L(\mathcal{G})$ defined by grammar $\mathcal{G}$ is the set

$$\{\theta \text{ in } \Sigma^* \mid \sigma \longrightarrow^* \theta \text{ for some } \sigma \text{ in } \mathcal{S}\}. \tag{1}$$

Two grammars $\mathcal{G}$, $\mathcal{G}'$ are called *equivalent* if $L(\mathcal{G}) = L(\mathcal{G}')$.

For the purposes of this paper we will always assume the terminal alphabet Σ contains two distinguished characters (and). We write $T = \Sigma - \{(,)\}$ to stand for the other letters of the terminal alphabet, and $U = V - \{(,)\}$ to stand for the other letters of the vocabulary.

The context-free grammar $\mathcal{G}$ is a *parenthesis grammar* if $\mathcal{S}$ is a subset of U^*, i.e. $\mathcal{S}$ contains no parenthesis symbols, and if all productions have the form $A \longrightarrow (\theta)$ where θ is in U^*. A set of strings $L \subseteq \Sigma^*$ is called a *parenthesis language* if $L = L(\mathcal{G})$ for some parenthesis grammar $\mathcal{G}$.

This definition is not identical to the one given by McNaughton, but it is easy to verify that L is a parenthesis language in our sense if and only if (L) is a parenthesis language in McNaughton's sense. Our definition has the slight advantage that parenthesis languages are closed under concatenation.

Our goal is to find a method to take an arbitrary context-free grammar $\mathcal{G}$ and to determine whether or not there is an equivalent parenthesis grammar $\mathcal{G}'$. Throughout the constructions below we will assume no *useless* nonterminal symbols are present in the grammars we deal with. A nonterminal symbol A is called useless if it has no influence on $L(\mathcal{G})$, i.e. unless there exist σ in $\mathcal{S}$ and α, θ, ω in Σ^* such that $\sigma \longrightarrow^* \alpha A \omega \longrightarrow^* \alpha\theta\omega$. Well-known methods exist to recognize and remove all useless nonterminals from a grammar.

In the discussion below we generally let upper case letters A, B, $\cdots$ stand for nonterminals; lower case letters a, b, $\cdots$ for elements of T; lower case letters x, y, for elements of V; and lower case Greek letters for strings. The symbol ϵ denotes the empty string. The statement "α is an initial substring of θ" means there exists a string ω such that $\alpha\omega = \theta$. The notation $|\, \theta \,|$ stands for the length of θ, so that $|\, \epsilon \,| = 0$ and $|\, \theta x \,| = |\, \theta \,| + 1$.

2. PARENTHESIS STRUCTURE

For any θ in Σ^* we define the functions "content" $c(\theta)$ and "deficiency" $d(\theta)$ as follows:

$$c(x) = \begin{cases} +1, \text{ if } x = (\\ 0, \text{ if } x \in T \\ -1, \text{ if } x =) \end{cases} \qquad d(x) = \begin{cases} 0, \text{ if } x = (\\ 0, \text{ if } x \in T \\ 1, \text{ if } x =) \end{cases} \tag{2}$$

$c(\epsilon) = d(\epsilon) = 0$, $c(\theta x) = c(\theta) + c(x)$, $d(\theta x) = \max(d(\theta), d(x) - c(\theta))$. It follows that for all θ, φ we have

$$c(\theta\varphi) = c(\theta) + c(\varphi); \quad d(\theta\varphi) = \max(d(\theta), d(\varphi) - c(\theta)). \tag{3}$$

The value $c(\theta)$ is clearly the excess of left parentheses over right parentheses in θ, and $d(\theta)$ is the greatest deficiency of left parentheses from right parentheses in any initial substring of θ. We say θ is *balanced* if $c(\theta) = d(\theta) = 0$.

The left and right parentheses in the string $\alpha(\theta)\omega$ are said to *match* if θ is balanced. The left parenthesis in the string $\alpha(\theta$ is said to be *free* if $d(\theta) = 0$. The right parenthesis in the string $\theta)\omega$ is said to be *free* if $c(\theta) \leqq 0$ and $d(\theta) = -c(\theta)$. It follows that every parenthesis in a string is either free or has a unique mate, i.e., all non-free parentheses can be classified into matching pairs in a unique way. This corresponds to the familiar rules for parenthesis grouping, and we see that balanced strings are precisely those strings whose parentheses all have mates in the conventional sense. Moreover we have the general situation given in the following lemma:

LEMMA 1. *If φ is any string on Σ, φ can be written uniquely in the form*

$$\varphi = \varphi_0)\varphi_1) \cdots)\varphi_p(\varphi_{p+1}(\cdots (\varphi_q\,, \tag{4}$$

where $0 \leqq p \leqq q$ and each φ_j is balanced. Moreover, the p right parentheses and the $q - p$ left parentheses indicated in (4) *are precisely the free parentheses of φ, and*

$$p = d(\varphi), \qquad q - p = c(\varphi) + d(\varphi). \tag{5}$$

Proof. Although this lemma is intuitively clear it may be worthwhile to indicate a formal proof. Let us use induction on $|\varphi|$.

Case 1. φ is balanced. Then all parentheses within φ have mates and there are no free parentheses. For example if φ has the form $\alpha(\theta$ it follows that $c(\alpha) \geqq 0$ and $c(\theta) = c(\varphi) - c(\alpha) - 1 < 0$; hence the

left parenthesis is not free and θ contains a shortest initial substring θ' for which $c(\theta') < 0$. Clearly θ' has the form $\theta'')$ where θ'' is balanced. A dual argument shows each right parenthesis within φ has a mate. Hence in any representation such as (4) the parentheses shown are precisely the free parentheses when $\varphi_0, \cdots, \varphi_q$ are balanced; and this proves there is at most one such representation for any string φ. When φ is balanced the only possibility is $\varphi = \varphi_0$.

Case 2. $c(\varphi) > 0$, $d(\varphi) = 0$. Let φ_0 be the longest initial substring of φ for which $c(\varphi) = 0$; then clearly φ has the form $\varphi_0(\varphi'$ where φ_0 is balanced, $c(\varphi') = c(\varphi) - 1$, and $d(\varphi') = 0$. By induction the lemma holds for φ', so it holds also for φ.

Case 3. $d(\varphi) > 0$. Let ψ be the shortest initial substring of φ for which $c(\psi) < 0$. Then ψ has the form $\varphi_0)$ where φ_0 is balanced; and $\varphi = \varphi_0)\varphi'$ where $c(\varphi') = c(\varphi) + 1$, $d(\varphi') = d(\varphi) - 1$. By induction the lemma holds for φ.

Together with the concept of matching parentheses we have a relation of associate symbols: The symbols x, y in the string $\alpha\, x\, \theta\, y\, \omega$ are called *associates* if θy is balanced. The relation of being associates breaks any string into equivalence classes; for example in the string $ab(_1c(_2d)_3(_4e)_5f)_6g$ the sets of associates are $\{a, b,)_6, g\}$, $\{(_1, c,)_3,)_5, f\}$, $\{(_2, d\}$, $\{(_4, e)\}$. Here subscripts have been used to distinguish between appearances of parentheses.

LEMMA 2. *If φ is a nonempty string with $c(\varphi) = 0$, the string φ^n contains a set of at least n associate symbols.*

Proof. By lemma 1, we can write φ as $\varphi'\varphi''$ where φ' is nonempty, $c(\varphi') = -d(\varphi)$, $c(\varphi'') = d(\varphi)$, and $\varphi''\varphi'$ is balanced. Now $\varphi^n = \varphi'(\varphi''\varphi')^{n-1}\varphi''$, and the final characters of each φ' in this formula are associates.

A language $L \subseteq \Sigma^*$ is said to be *balanced* if every string in L is balanced. It is said to have *bounded associates* if there exists a constant m_0 such that if θ is in L and if x is a symbol of θ then x has at most m_0 associates in θ.

LEMMA 3. *Let $\mathcal{G} = (\Sigma, V, \mathcal{S}, \mathcal{P})$ be a parenthesis grammar. Then $L(\mathcal{G})$ is balanced and has bounded associates.*

Proof. Extend the above definitions from Σ to V by defining $c(A) = d(A) = 0$ for A in N. Suppose $\varphi \rightarrow \psi$ where φ is balanced and no symbol of φ has more than m associates. Then $\varphi = \alpha A \omega$ and $\psi = \alpha(\theta)\omega$ where θ is in U^*. It follows that ψ is balanced. Moreover, two symbols of ψ are associates if and only if they are associates in φ, or if one is the) and

the other is associated with A in φ, or if they are both part of $(\theta$. Hence no symbol of ψ has more than max $(m, \mid \theta \mid + 1)$ associates.

Consider now the relation $\sigma \to^* \theta$ for σ in $\mathfrak{S}$. By induction on the length of the derivation, namely the number of $\to$ steps implied by $\to^*$, we see that θ is balanced and its symbols have at most $m_0 = 1 +$ max $\{ \mid \theta \mid \mid \theta \in \mathfrak{S}$ or there is a production $A \to (\theta)$ in $\mathfrak{P}\}$ associates.

We shall eventually show that the converse of lemma 3 is true: Any context-free language which is balanced and has bounded associates must be a parenthesis language. First let us investigate balanced languages more closely.

LEMMA 4. *Let* $\mathfrak{G} = (\Sigma, V, \mathfrak{S}, \mathfrak{P})$ *be a context-free grammar for which* $L(\mathfrak{G})$ *is balanced. For each nonterminal* A *there exist numbers* $c(A)$, $d(A)$ *which can be found effectively from* $\mathfrak{G}$, *such that if* $A \to^* \theta \in \Sigma^*$ *then* $c(\theta) = c(A)$ *and* $d(\theta) \leqq d(A)$.

Proof. By our standard assumption, A is not useless, so we can find σ in $\mathfrak{S}$ and α, φ, ω in Σ^* such that $\sigma \to^* \alpha A \omega \to^* \alpha \varphi \omega$. Let $c(A) = c(\varphi)$ and $d(A) = c(\alpha)$. Then if $A \to^* \theta$ we have $\sigma \to^* \alpha\theta\omega$; so

$$c(\alpha) + c(\theta) + c(\omega) = c(\alpha\theta\omega) = 0 = c(\alpha\varphi\omega) = c(\alpha) + c(\varphi) + c(\omega),$$

and $c(\theta) = c(A)$. Also $d(\theta) - c(\alpha) \leqq d(\alpha\theta\omega) = 0$ so $d(\theta) \leqq d(A)$.

THEOREM 1. *If* $\mathfrak{G} = (\Sigma, V, \mathfrak{S}, \mathfrak{P})$ *is a context-free grammar, there is an effective algorithm which determines whether or not* $L(\mathfrak{G})$ *is balanced.*

Proof. Use the construction in the proof of the preceding lemma to define $c(A)$ and $d(A)$ for each nonterminal A. For each of the finitely many functions d_1 such that $0 \leqq d_1(A) \leqq d(A)$, all $A \in N$, and $d_1(x) = d(x)$, all $x \in \Sigma$, attempt to verify the following facts:

(i) $c(\sigma) = d_1(\sigma) = 0$ for all $\sigma \in \mathfrak{S}$.

(ii) $c(A) = c(\theta)$ for all productions $A \to \theta$.

(iii) $d_1(A) \geqq d_1(\theta)$ for all productions $A \to \theta$ (using formula (3) to calculate $d_1(\theta)$).

If there is some choice of d_1 for which these three conditions hold, then by an induction argument such as in the first part of the proof of lemma 3, $L(\mathfrak{G})$ is balanced. Conversely if $L(\mathfrak{G})$ is balanced, there will exist such a choice of d_1, namely $d_1(A) = \max \{d(\varphi) \mid A \to^* \varphi \in \Sigma^*\}$. For this d_1, condition (iii) must hold, since there exists $\theta' \in \Sigma^*$ for which $\theta \to^* \theta'$ and $d_1(\theta) = d_1(\theta')$.

The result of lemma 4 can be further refined. Let us say a grammar is *completely qualified* if, for each $A \in N$, there are numbers $c(A)$, $d(A)$ for which $c(\theta) = c(A)$, $d(\theta) = d(A)$, whenever $A \to^* \theta \in \Sigma^*$.

LEMMA 5. *Let* $\mathcal{G} = (\Sigma, V, \mathcal{S}, \mathcal{P})$ *be a context-free grammar for which* $L(\mathcal{G})$ *is balanced. It is possible to construct a completely qualified grammar* $\mathcal{G}' = (\Sigma, V', \mathcal{S}', \mathcal{P}')$ *which is equivalent to* $\mathcal{G}$.

Proof. Let $c(A)$, $d(A)$ be defined for all A as in lemma 4. Let $V' = \Sigma \cup \{[A, j] \mid 0 \leqq j \leqq d(A), A \in N\}$. Define $\tau(A) = \{[A, j] \mid 0 \leqq j \leqq d(A)\}$, and $\tau(x) = \{x\}$ for $x \in \Sigma$. Extend τ to V^* by defining $\tau(x_1 x_2 \cdots x_n) = \{y_1 y_2 \cdots y_n \mid y_k \in \tau(x_k), 1 \leqq k \leqq n\} \subseteq V'^*$. Also define $c([A, j]) = c(A)$, $d([A, j]) = j$. Then Eq. (3) can be used to define $c(\theta)$ and $d(\theta)$ for all $\theta \in V'^*$. The grammar $\mathcal{G}'$ is now defined as follows:

$$\mathcal{S}' = \bigcup\{\tau(\sigma) \mid \sigma \text{ in } \mathcal{S}\},$$

$$\mathcal{P}' = \{[A, j] \to \varphi \mid d(\varphi) = j,\ \varphi \in \tau(\theta),\ A \to \theta \in \mathcal{P}\}.$$

It is obvious that $L(\mathcal{G}') \subseteq L(\mathcal{G})$, since any derivation in $\mathcal{G}'$ can be "mapped into" a derivation in $\mathcal{G}$ by replacing $[A, j]$ by A. And it is easy to show, by induction on the length of derivation, that if $\theta \to^* \varphi \in \Sigma^*$ in $\mathcal{G}$ there is some θ' in $\tau(\theta)$ for which $\theta' \to^* \varphi$ in $\mathcal{G}'$. Hence $L(\mathcal{G}) \subseteq L(\mathcal{G}')$. Furthermore any terminal string descended from $[A, j]$ in $\mathcal{G}'$ has deficiency j, so it is clear that $\mathcal{G}'$ is completely qualified.

At this point it is tempting to conjecture that if L is a balanced, context-free language, then L possesses a *balanced grammar*, i.e. a completely qualified grammar such that $c(A) = d(A) = 0$ for all $A \in N$. However, we have the following counterexample:

THEOREM 2. *The balanced language* $L_0 = \{a^n(b^n) \mid n \geqq 0\}$ *cannot be defined by a balanced grammar.*

Proof. Suppose $\mathcal{G} = (\Sigma, V, \mathcal{S}, \mathcal{P})$ is a balanced grammar with $\alpha(\mathcal{G}) = L_0$. Every nonterminal A belongs to one of two disjoint classes:

Class 1. $A \to^* \theta \in \Sigma^*$ implies that θ contains precisely one occurrence of each of (and).

Class 2. $A \to^* \theta \in \Sigma^*$ implies that θ contains no occurrences of parentheses.

This follows from the facts that $c(\theta)$ must equal 0 and that each string of L_0 has just one pair of parentheses. Add a new nonterminal symbol S and add the productions $S \to \sigma$ for all $\sigma \in \mathcal{S}$, then replace $\mathcal{S}$ by $\{S\}$. Then S is of class 1, so we see there must be a production where we switch to class 2, i.e. a production of the form

$$A \to \alpha(\theta)\omega,$$

where A is of class 1 and all nonterminals in $\alpha\theta\omega$ are of class 2, and where $\theta \to^* b^n$ for infinitely many n. This clearly is a contradiction.

3. THE MAIN CONSTRUCTION

The example in theorem 2 shows how difficult it is in general to obtain a balanced grammar for a balanced language. But if we add another hypothesis, such a transformation can always be carried out, as shown in theorem 3 below.

Before considering the general construction of transformations, let us consider first the elementary operations which are involved. It is obvious that the following well-known transformations to a grammar do not change the language defined by that grammar:

TRANSFORMATION 1. *Add a new nonterminal symbol X to the vocabulary; change a production $A \to \alpha\beta\gamma$ to the production $A \to \alpha X\gamma$, for some A, α, β, γ; and add the production $X \to \beta$.*

TRANSFORMATION 2. *Let A be a nonterminal symbol and let $\rho(A) = \{\theta \mid A \to \theta \in \mathcal{P}\}$. Define $\rho(x) = \{x\}$ for all $x \in V - \{A\}$, and $\rho(x_1 \cdots x_n) = \{y_1 \cdots y_n \mid y_k \in \rho(x_k),\ 1 \leqq k \leqq n\}$. Then change $\mathcal{S}$ to $\rho(\mathcal{S}) = \{\sigma' \mid \sigma' \in \rho(\sigma),$ some $\sigma \in \mathcal{S}\}$, and change $\mathcal{P}$ to $\mathcal{P}' = \{B \to \theta' \mid \theta' \in \rho(\theta),$ some θ such that $B \to \theta \in \mathcal{P}\}$.*

In essence, transformation 1 adds one step to a derivation each time the production $A \to \alpha\beta\gamma$ is applied. Transformation 2 takes a shortcut by removing derivation steps when A is involved. Notice that if the nonterminal symbol A does not appear on the righthand side of any production $A \to \theta$ then transformation 2 makes A become "useless"; we will make use of this fact to remove A from the grammar.

THEOREM 3. *Let $\mathcal{G} = (\Sigma, V, \mathcal{S}, \mathcal{P})$ be a context-free grammar for which $L(\mathcal{G})$ is balanced and has bounded associates. Then it is possible to construct an equivalent balanced grammar effectively from $\mathcal{G}$.*

Proof. We may assume from lemma 5 that $\mathcal{G}$ is completely qualified. We may also assume that $\mathcal{G}$ is not "circular", i.e. the relation $A \to^+ A$ does not hold for any nonterminal A; there are well-known methods for removing circularity, basically by defining $A \equiv B$ if $A \to^+ B \to^+ A$ and by replacing each equivalence class by a single symbol.

Consider now a typical production

$$A \to x_1 x_2 \cdots x_n\,, \tag{6}$$

in a completely qualified grammar. We may form a "parenthesis image" of the string $x_1 x_2 \cdots x_n$ by replacing each symbol x_j by a sequence of

$d(x_j)$ right parentheses followed by $c(x_j) + d(x_j)$ left parentheses; the result is a set of parentheses which has $d(A)$ free right parentheses and $c(A) + d(A)$ free left parentheses. (See lemma 1; note that $c(x_j) + d(x_j)$ cannot be negative, since each x_j represents at least one string of Σ^*.)

For example suppose we have the following values of the c and d functions:

x	$c(x)$	$d(x)$
A	-1	4
B	-1	2
C	2	3
(	1	0
)	-1	1
a, b	0	0

If we have the production

$$A \rightarrow) \; b \; B \; (\; a \; C \; A \; B \; b \tag{7}$$

the parenthesis image is

$$))))(((\rightarrow)))(( )))((((( ))))((( ))(,$$

and this has 4 free right parentheses and 3 free left parentheses on both sides of the production. We may now abstract (7), so that only the positions of free parentheses are shown, as follows:

$$\begin{aligned}[A)1][A)2][A)3][A)4][A(1][A(2][A(3] \\ \rightarrow) \; [B)1][B)2][C)3][C(1][A(1][B(1].\end{aligned} \tag{8}$$

Here $[A)1]$ denotes the first free right parenthesis of A, $[A)2]$ is the second, etc. Thus, the second free left parenthesis of any terminal string derived from the righthand side of (7) must be the first free left parenthesis of the string derived from the leftmost B.

Now for any completely qualified grammar $\mathfrak{G}$, consider the directed graph $\mathfrak{D}$ defined in the following way. The vertices of $\mathfrak{D}$ are all the symbols $[A)u]$, $[A(v]$ where A is a nonterminal symbol and $1 \leqq u \leqq d(A)$, $1 \leqq v \leqq c(A) + d(A)$. For each production (6) we include at most $c(A) + 2d(A)$ directed arcs in $\mathfrak{D}$, one for each free parenthesis that does not correspond to an actual parenthesis in the parenthesis image of the right side. From production (7), for example, we would include arcs from $[A)2]$ to $[B)1]$, $[A)3]$ to $[B)2]$, $[A)4]$ to $[C)3]$, $\cdots$, $[A(3]$ to $[B(1]$.

(Compare with (8); no arc is drawn from $[A)1]$ since it corresponds to a real right parenthesis on the right of (8).)

The important property of $\mathfrak{D}$ is that it contains *no oriented cycles* (no paths from a vertex to itself) when $\mathcal{G}$ is not circular and when $L(\mathcal{G})$ has bounded associates. To prove this property, suppose there is a path in $\mathfrak{D}$ from $[A)u]$ to $[A)u]$ for some A and u. By the definition of $\mathfrak{D}$ this is equivalent to saying there are strings α, ω in Σ^* for which $A \to^+ \alpha A \omega$, where the u-th free right parenthesis in the parenthesis image of $\alpha A \omega$ is the u-th free right parenthesis coming from the A. It follows that $c(\alpha) = 0$, since by lemma 1 the u-th free right parenthesis in any string φ is preceded by a string φ' with $c(\varphi') = 1 - u$. We also have $c(A) = c(\alpha) + c(A) + c(\omega)$, hence $c(\omega) = 0$. Now by assumption $\mathcal{G}$ is not circular, so α and ω are not both empty. Also $A \to^+ \alpha^n A \omega^n$ for all $n > 0$, and since A is not useless there are strings α', θ, ω' such that $\alpha' \alpha^n \theta \, \omega^n \omega'$ is in $L(\mathcal{G})$ for all $n > 0$. By lemma 2 this contradicts the assumption that $L(\mathcal{G})$ has bounded associates. A dual argument shows there is no path in $\mathfrak{D}$ from $[A(v]$ to $[A(v]$ for any A and v.

The directed graph $\mathfrak{D}$ is empty (i.e. has no vertices) if and only if $\mathcal{G}$ is a balanced grammar. Therefore the rest of the proof consists of showing, when $\mathfrak{D}$ is not empty, that an equivalent grammar can be constructed whose corresponding graph is empty.

If $\mathfrak{D}$ is not empty, then since there are no cycles there must be at least one "sink" vertex, i.e. a vertex from which no arcs lead outward. Let $[A)u]$ be such a vertex; a dual argument will apply to a vertex $[A(v]$. By the definition of $\mathfrak{D}$, the set of all productions whose left-hand side is A can be written in the form

$$\begin{aligned} A &\to \xi_1)\eta_1 \\ A &\to \xi_2)\eta_2 \\ &\vdots \\ A &\to \xi_n)\eta_n , \end{aligned} \tag{9}$$

where

$$d(\xi_j) = -c(\xi_j) = u - 1, c(\eta_j) = c(A) + u, d(\eta_j) = d(A) - u, \tag{10}$$

for $1 \leqq j \leqq n$.

We can apply transformation 1 to form a new grammar $\mathcal{G}'$ equivalent to $\mathcal{G}$, replacing (9) by the productions

$$A \to X_j)Y_j , \; X_j \to \xi_j , \; Y_j \to \eta_j , \; 1 \leqq j \leqq n \tag{11}$$

where the X_j and Y_j are new nonterminal symbols. By (10), $\mathcal{G}'$ is a fully qualified grammar. Now form another equivalent grammar $\mathcal{G}''$ by applying transformation 2 to the nonterminal A, and deleting A. $\mathcal{G}''$ is now a fully qualified grammar equivalent to $\mathcal{G}$.

This construction $\mathcal{G} \to \mathcal{G}' \to \mathcal{G}''$ has a corresponding effect on the directed graphs $\mathfrak{D} \to \mathfrak{D}' \to \mathfrak{D}''$. In order to study $\mathfrak{D}'$ and $\mathfrak{D}''$ it is convenient to introduce the following equivalence relation on the vertices of $\mathfrak{D} \cup \mathfrak{D}''$: For $1 \leqq j \leqq n$, let $[A)k] \equiv [X_j)k]$, for $1 \leqq k < u$; $[A)u + k] \equiv [Y_j)k]$, for $1 \leqq k \leqq d(A) - u$; $[A(k] \equiv [Y_j(k]$ for $1 \leqq k \leqq c(A) + d(A)$. This definition can be completed to a definition of equivalence between the vertices of all digraphs later derived from $\mathfrak{D}''$. It is now easy to see that the directed graph $\mathfrak{D}''$ is like $\mathfrak{D}$ with the following changes: (1) The vertices $[X_j)k]$, $[Y_j)k]$, $[Y_j(k]$ just mentioned are added. (2) The arcs which came from a vertex $[A)k]$ or $[A(k]$ in $\mathfrak{D}$ now are shifted so they emanate from an equivalent vertex in $\mathfrak{D}''$. (3) The arcs which were directed into a vertex $[A)k]$, $k \neq u$, or into a vertex $[A(k]$ in $\mathfrak{D}$ now become n arcs directed into the n new equivalent vertices. (4) The arcs which were directed into $[A)u]$ in $\mathfrak{D}$ are deleted. (5) The vertices $[A)k]$ and $[A(k]$ are deleted.

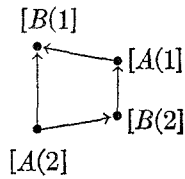
The transition from $\mathcal{G}$ to $\mathcal{G}''$ not only tends to increase the size of the grammar $\mathcal{G}$, it also can increase the number of arcs and vertices in $\mathfrak{D}$. Therefore it is perhaps hard to believe that this process can be iterated until $\mathfrak{D}$ loses all its vertices and arcs. But that is exactly what will happen, if the transformations are applied systematically. For let us consider the equivalence relation between vertices just defined, and let $\mathfrak{D}_0$ be the directed graph whose vertices are equivalence classes of vertices of $\mathfrak{D}$, and whose arcs go from class V to class V' if and only if there is at least one arc in $\mathfrak{D}$ from a vertex of class V to a vertex of class V'. Then $\mathfrak{D}_0''$ is equal to $\mathfrak{D}_0$; except when the class containing $[A)u]$ had only one element in $\mathfrak{D}$, this class and the arcs leading to it are not present in $\mathfrak{D}_0''$. If the class containing $[A)u]$ has more than one element, we can repeat the construction on vertices of this class until it disappears from $\mathfrak{D}_0$. (Note that $[A)u]$ is never equivalent to $[A)k]$ for $k \neq u$, so the construction must decrease the size of the equivalence class we are currently working on.) Therefore it is possible to use induction on the number of vertices in $\mathfrak{D}_0$, and the process ultimately terminates with a balanced grammar.

Since the construction in the proof of theorem 3 is rather involved, it may be of interest to work a nontrivial example here. Consider the

grammar $\mathcal{G}$ with $\Sigma = \{(,), a, b, c, d, e\}$, $V = \Sigma \cup \{A, B\}$, $\mathcal{S} = \{B))\}$, and productions

$$A \to Ba,\ A \to (bB),\ A \to (c(,$$
$$B \to (dA))((),\ B \to (B)))(Ae).$$

Then $c(A) = c(B) = 2$, $d(A) = d(B) = 0$. The directed graph $\mathfrak{D}$ is



First we eliminate the only "sink" vertex, $[B(1]$:

$$\mathcal{S} = \{B_{11}(B_{12})),\ B_{21}(B_{22}))\}$$
$$\mathcal{P} = \{A \to B_{11}(B_{12}a,\ A \to B_{21}(B_{22}a,\ A \to (bB_{11}(B_{12}),\ A \to (bB_{21}(B_{22}),$$
$$A \to (c(,\ B_{11} \to \epsilon,\ B_{12} \to dA))((),$$
$$B_{21} \to (B_{11}(B_{12}))),\ B_{21} \to (B_{21}(B_{22}))),\ B_{22} \to Ae)\}.$$

The construction can be simplified if we use transformation 2 to remove all new nonterminals X_j, Y_j in (11) for which ξ_j or η_j respectively are merely balanced strings on Σ. To reduce the size of the grammar let us consistently use this simplification; we would in this example have omitted B_{11} :

$$\mathcal{S} = \{(B_{12})),\ B_{21}(B_{22}))\}$$
$$\mathcal{P} = \{A \to (B_{12}a,\ A \to B_{21}(B_{22}a,\ A \to (b(B_{12}),\ A \to (bB_{21}(B_{22}),\ A \to (c(,$$
$$B_{12} \to dA))((),\ B_{21} \to ((B_{12}))),\ B_{21} \to (B_{21}(B_{22}))),\ B_{22} \to Ae)\}.$$

The next step is to remove $[A(1]$ which now is a "sink":

$$\mathcal{S} = \{(B_{12})),\ B_{21}(B_{22}))\}$$
$$\mathcal{P} = \{A_{12} \to B_{12}a,\ A_{21} \to B_{21}\ ,\ A_{22} \to B_{22}a,\ A_{32} \to b(B_{12}),$$
$$A_{42} \to bB_{21}(B_{22}),\ A_{52} \to c(,$$
$$B_{12} \to d(A_{12}))((),\ B_{12} \to dA_{21}(A_{22}))((),\ B_{12} \to d(A_{32}))((),$$
$$B_{12} \to d(A_{42}))((),\ B_{12} \to d(A_{52}))((),$$

$$B_{21} \to ((B_{12}))),\ B_{21} \to (B_{21}(B_{22}))),$$
$$B_{22} \to (A_{12}e),\ B_{22} \to A_{21}(A_{22}e),\ B_{22} \to (A_{32}e),\ B_{22} \to (A_{42}e),$$
$$B_{22} \to (A_{52}e)\}.$$

Now we remove the vertices equivalent to $[B(2]$, namely $[B_{12}(1]$ and $[B_{22}(1]$:

$$\mathfrak{S} = \{(B_{1211}(())),\ (B_{1221}(())),\ (B_{1231}(())),\ (B_{1241}(())),\ (B_{1251}(())),\ B_{21}((B_{2212})),$$
$$B_{21}(B_{2221}(B_{2222})),\ B_{21}((B_{2232})),\ B_{21}((B_{2242})),\ B_{21}((B_{2252}))\}$$

$$\mathfrak{P} = \{A_{12} \to B_{1211}()a,\ A_{12} \to B_{1221}()a,\ A_{12} \to B_{1231}()a,\ A_{12} \to B_{1241}()a,$$
$$A_{12} \to B_{1251}()a,$$
$$A_{21} \to B_{21}\,,$$
$$A_{22} \to (B_{2212}a,\ A_{22} \to B_{2221}(B_{2222}a,\ A_{22} \to (B_{2232}a,\ A_{22} \to (B_{2242}a,$$
$$A_{22} \to (B_{2252}a,$$
$$A_{32} \to b(B_{1211}(()),\ A_{32} \to b(B_{1221}(()),\ A_{32} \to b(B_{1231}(()),$$
$$A_{32} \to b(B_{1241}(()),\ A_{42} \to b(B_{1251}(()),$$
$$A_{42} \to bB_{21}((B_{2212}),\ bB_{21}(B_{2221}(B_{2222}),\ A_{42} \to bB_{21}((B_{2232}),$$
$$A_{42} \to bB_{21}((B_{2242}),\ A_{42} \to bB_{21}((B_{2252}),$$
$$A_{52} \to c(,$$
$$B_{1211} \to d(A_{12})),\ B_{1221} \to dA_{21}(A_{22})),\ B_{1231} \to d(A_{32})),$$
$$B_{1241} \to d(A_{42})),\ B_{1251} \to d(A_{52})),$$
$$B_{21} \to ((B_{1211}(()))),\ B_{21} \to ((B_{1221}(()))),\ B_{21} \to ((B_{1231}(()))),$$
$$B_{21} \to ((B_{1241}(()))),\ B_{21} \to ((B_{1251}(()))),$$
$$B_{21} \to (B_{21}((B_{2212}))),\ B_{21} \to (B_{21}(B_{2221}(B_{2222}))),\ B_{21} \to (B_{21}((B_{2232}))),$$
$$B_{21} \to (B_{21}((B_{2242}))),\ B_{21} \to (B_{21}((B_{2252}))),$$
$$B_{2212} \to A_{12}e),\ B_{2221} \to A_{21}\,,\ B_{2222} \to A_{22}e),\ B_{2232} \to A_{32}e),$$
$$B_{2242} \to A_{42}e),\ B_{2252} \to A_{52}e)\}.$$

It is clear that the vertices equivalent to $[A(2]$, namely $[A_{12}(1]$, $[A_{22}(1]$, $[A_{32}(1]$, $[A_{42}(1]$, and $[A_{52}(1]$, may now be removed in the same fashion,

and we obtain a long, balanced grammar whose nonterminals are $\{A_{1211}, A_{1221}, A_{1231}, A_{1241}, A_{1251}, A_{21}, A_{2212}, A_{2221}, A_{2222}, A_{2232}, A_{2242}, A_{2252}, A_{3212}, A_{3222}, A_{3232}, A_{3242}, A_{3252}, A_{4211}, A_{4212}, A_{4221}, A_{4222}, A_{4231}, A_{4232}, A_{4241}, A_{4242}, A_{4251}, A_{4252}, B_{1211}, B_{1221}, B_{1231}, B_{1241}, B_{1251}, B_{21}, B_{2212}, B_{2221}, B_{2222}, B_{2232}, B_{2242}, B_{2252}\}$. It can be shown that $L(\mathcal{G})$ is a parenthesis language.

4. THE MAIN THEOREM

The construction in the previous section allows us to work with balanced grammars, but there obviously are grammars (e.g. those with no parentheses at all) which are balanced but do not correspond to a language with bounded associates.

The next result is the final link in the chain needed to characterize parenthesis languages:

LEMMA 6. *Let* $\mathcal{G} = (\Sigma, V, \mathcal{S}, \mathcal{P})$ *be a balanced grammar for which* $L(\mathcal{G})$ *is balanced and has bounded associates. Then it is possible to construct a parenthesis grammar* $\mathcal{G}'$ *effectively from* $\mathcal{G}$, *where* $L(\mathcal{G}') = L(\mathcal{G})$.

Proof. As in the proof of theorem 3, we may assume $\mathcal{G}$ is not circular. Let us also add a new nonterminal symbol S, new productions $S \to \sigma$ for all $\sigma \in \mathcal{S}$, and change $\mathcal{S}$ to $\{S\}$. The resulting equivalent grammar $\mathcal{G}_1$ is balanced since $L(\mathcal{G})$ and $\mathcal{G}$ are balanced.

Now we can modify $\mathcal{G}_1$ by successively applying "transformation 1" of the previous section, until we obtain a grammar $\mathcal{G}_2$ in which the right-hand sides of all productions have one of the forms

$$\theta \text{ or } (\theta)$$

where $\theta \in U^*$, i.e. θ is a string with no parentheses. For example, the production

$$A \to a((Bc)d(e)f)$$

can be replaced by the set

$$A \to aX$$
$$X \to (YdZf)$$
$$Y \to (Bc)$$
$$Z \to (e)$$

where X, Y, Z are new nonterminals.

Now in $\mathcal{G}_2$ let us define the relation $A \prec B$ for nonterminals A, B if

there exists a production $A \to \alpha B \omega$ such that α, ω are in U^*. This relation generates a transitive completion relation $A \prec^+ B$ as before.

It is impossible to have $A \prec^+ A$ for any nonterminal A; for this would imply $A \to^+ \alpha A \omega$ for some balanced strings $\alpha, \omega \in \Sigma^*$, and since $\mathcal{G}$ is not circular we would have $\alpha\omega \neq \epsilon$ and (as in the proof of theorem 3) $L(\mathcal{G})$ would not have bounded associates.

Therefore the relation $\prec^+$ is a partial ordering, and it is possible to arrange the nonterminals of $\mathcal{G}_2$ into a sequence $A_1, A_2, \cdots, A_n$ such that

$$A_j \prec^+ A_k \text{ implies } j < k.$$

Let us add new nonterminals $X_1, X_2, \cdots, X_n$ to $\mathcal{G}_2$, add the productions $A_j \to X_j$ for $1 \leqq j \leqq n$, and replace every production of the form $A_j \to (\theta)$ by the production $X_j \to (\theta)$. Finally let us remove the nonterminals $A_1, \cdots, A_n$ as follows: Assume all A_k have been removed for $k > j$, and apply "transformation 2" of the previous section to A_j; this removes A_j. If this process is performed for $j = n, \cdots, 2, 1$, we clearly obtain a parenthesis grammar, since all productions not involving parentheses have been removed and no new ones have been created.

As an example of the construction in lemma 6, consider the grammar $\mathcal{G}$ with $\Sigma = \{(,), a, b\}$, $V = \Sigma \cup \{A, B\}$, $\mathcal{S} = \{A(B), c\}$, $\mathcal{P} = \{A \to aBB, A \to (A), B \to \epsilon, B \to (B)(Ab)\}$. The grammar $\mathcal{G}_1$ is $(\Sigma, V_1, \mathcal{S}_1, \mathcal{P}_1)$ where $V_1 = V \cup \{S\}$, $\mathcal{S}_1 = \{S\}$, $\mathcal{P}_1 = \mathcal{P} \cup \{S \to A(B), S \to c\}$. The grammar $\mathcal{G}_2$ is $(\Sigma, V_1 \cup \{C, D\}, \{S\}, \mathcal{P}_2)$ where

$$\mathcal{P}_2 = \{A \to aBB, A \to (A), B \to \epsilon, B \to CD, C \to (B), D \to (Ab), S \to AC, S \to c\}.$$

We have $S \prec A \prec B \prec C, B \prec D$. Adding new nonterminals X_S, X_A, X_B, X_C, X_D, these productions are changed to

$$\{A \to X_A, B \to X_B, C \to X_C, D \to X_D, S \to X_S,$$
$$A \to aBB, X_A \to (A), B \to \epsilon, B \to CD, X_C \to (B), X_D \to (Ab),$$
$$S \to AC, S \to c\}.$$

Eliminating D, C, and then B, and noting that X_B, X_S are useless, we get

$$\{A \to X_A,$$
$$A \to a, A \to aX_CX_D, A \to aX_CX_DX_CX_D, X_A \to (A),$$
$$X_C \to (), X_C \to (X_CX_D), X_D \to (Ab), S \to AX_C, S \to c\}.$$

Finally eliminate A and then S to get the parenthesis grammar $\mathcal{G}' = \{\Sigma, V', \mathcal{S}', \mathcal{P}'\}$ where $V' = \Sigma \cup \{X_A, X_C, X_D\}$, $\mathcal{S}' = \{X_AX_C, aX_C, aX_CX_DX_C, aX_CX_DX_CX_DX_C, c\}$,

$$\mathcal{P}' = \{X_A \to (X_A), X_A \to (a), X_A \to (aX_CX_D), X_A \to (aX_CX_DX_CX_D),$$
$$X_C \to (), X_C \to (X_CX_D), X_D \to (X_Ab), X_D \to (ab),$$
$$X_D \to (aX_CX_Db), X_D \to (aX_CX_DX_CX_Db)\}.$$

THEOREM 4. *A context-free language is a parenthesis language if and only if it is balanced and has bounded associates. If* $\mathcal{G} = (\Sigma, V, \mathcal{S}, \mathcal{P})$ *is a context-free grammar, there is an effective algorithm which determines whether or not* $L(\mathcal{G})$ *is a parenthesis language; and if* $L(\mathcal{G})$ *is a parenthesis language, a parenthesis grammar* $\mathcal{G}' = (\Sigma, V', \mathcal{S}', \mathcal{P}')$ *can be effectively constructed from* $\mathcal{G}$.

Proof. If $\mathcal{G}$ is a parenthesis grammar, $L(\mathcal{G})$ is balanced and has bounded associates by lemma 3. Conversely if $L(\mathcal{G})$ is balanced and has bounded associates, we may apply theorem 3 and then lemma 6 to construct an equivalent parenthesis grammar.

To solve the stated decision problem, we can first decide if $L(\mathcal{G})$ is balanced, using the method of theorem 1. If it is balanced, we may continue by finding a completely qualified grammar as in lemma 5. Now the construction in the proof of theorem 3 can be carried out unless the directed graph $\mathfrak{D}$ defined there has oriented cycles; $\mathfrak{D}$ can be effectively constructed and examined for cycles. If we get through the construction in theorem 3, we still cannot be sure that $L(\mathcal{G})$ has bounded associates, but the construction in the proof of lemma 6 can be carried out unless the relation $A \prec^+ B$ is not a partial ordering; the latter condition is also equivalent to the existence of an oriented cycle in an appropriate directed graph (namely a directed graph with arcs from A to B if $A \prec B$). Hence we have an algorithm which either constructs the desired grammar $\mathcal{G}'$ or which determines that it would be impossible to do so.

COROLLARY. *If* $\mathcal{G} = (\Sigma, V, \mathcal{S}, \mathcal{P})$ *is any context-free grammar and if* $\mathcal{G}' = (\Sigma', V', \mathcal{S}', \mathcal{P}')$ *is a parenthesis grammar, there is an effective algorithm to decide if* $L(\mathcal{G}) = L(\mathcal{G}')$.

Proof. First decide if $L(\mathcal{G})$ is a parenthesis language; and if it is, find a parenthesis grammar $\mathcal{G}''$ equivalent to $\mathcal{G}$. Then use the procedure of McNaughton (1967) or the procedure described in the next section, to decide if $L(\mathcal{G}'') = L(\mathcal{G}')$.

Theorem 4 and its corollary can be extended to grammars in which

several kinds of parenthesis are used (e.g. both parenthesis and brackets), as in Ginsburg and Harrison (1967), where considerably more restrictive conditions are required. One may replace each type of parenthesis pair by two symbols ($(a$ and $a)$ where a identifies the kind of parenthesis being used. It follows that we can solve the three open problems stated by Ginsburg and Harrison (1967, p. 20) by considering all possible choices for the parenthesis pairs, using rather simple arguments; details are omitted.

The corollary to theorem 4 cannot be extended to the case that $\mathcal{G}'$ is a balanced grammar, since even the superficially simple problem of deciding whether $L(\mathcal{G}) = T^*$ is well-known to be recursively unsolvable (see Bar-Hillel, Perles, and Shamir (1961)). It follows also that we cannot extend the corollary to "one-sided" parenthesis grammars, since we cannot decide if $L(\mathcal{G})$ equals $L((\{(, a, b\}, \{(, a, b, S\}, \{S\}, \{S \to \epsilon, S \to (aS, S \to (bS\}))$.

5. BOOLEAN PROPERTIES OF PARENTHESIS LANGUAGES

Now let us study the relationships satisfied by parenthesis languages with respect to set operations. It is obvious from the definition that the set of parenthesis languages is closed under union and concatenation; for if we are given parenthesis grammars $\mathcal{G}_1 = (\Sigma, V_1, \mathcal{S}_1, \mathcal{P}_1)$ and $\mathcal{G}_2 = (\Sigma, V_2, \mathcal{S}_2, \mathcal{P}_2)$, where we may assume $V_1 - \Sigma$ and $V_2 - \Sigma$ are disjoint, then $\mathcal{G}_3 = (\Sigma, V_1 \cup V_2, \mathcal{S}_1 \cup \mathcal{S}_2, \mathcal{P}_1 \cup \mathcal{P}_2)$ and $\mathcal{G}_4 = (\Sigma, V_1 \cup V_2, \mathcal{S}_1\mathcal{S}_2, \mathcal{P}_1 \cup \mathcal{P}_2)$ generate $L(\mathcal{G}_1) \cup L(\mathcal{G}_2)$ and $L(\mathcal{G}_1)L(\mathcal{G}_2)$, respectively. Much more is true:

THEOREM 5. *The set of parenthesis languages is closed under relative complementation. (Thus, if L_1 and L_2 are parenthesis languages, so is $L_1 - L_2$.)*

Proof. Let $\mathcal{G}_1 = (\Sigma, V_1, \mathcal{S}_1, \mathcal{P}_1)$ and $\mathcal{G}_2 = (\Sigma, V_2, \mathcal{S}_2, \mathcal{P}_2)$ be parenthesis grammars, and assume the nonterminal alphabets $N_1 = V_1 - \Sigma$ and $N_2 = V_2 - \Sigma$ are disjoint. Let V be Σ plus the set of all pairs $[A, \mathfrak{B}]$ where A is in N_1 and $\mathfrak{B}$ is a subset of N_2. For any pair of strings θ_1, θ_2 over $(V_1 \cup V_2 \cup V)^*$, let

$$\theta_2 \sim \theta_2$$

mean θ_1 and θ_2 are of the same length and agree at all terminals, i.e. $\theta_1 = x_1x_2 \cdots x_n$ and $\theta_2 = y_1y_2 \cdots y_n$ where $x_j = y_j$ whenever either x_j or y_j is in Σ.

Now define the set $\alpha \doteq \beta$ for strings $\alpha \in V^*, \beta \in V_2^*$ as follows, when

$\alpha = x_1 \cdots x_n$ and $\beta = y_1 \cdots y_m$:

$$\alpha \dot{-} \beta = \begin{cases} \{x_1 \cdots x_{k-1}[A, \mathfrak{B} \cup \{y_k\}]x_{k+1} \cdots x_n \mid 1 \leqq k \leqq n \text{ and } x_k = [A, \mathfrak{B}]\}, & \text{if } \alpha \sim \beta; \\ \{\alpha\}, & \text{if } \alpha \nsim \beta. \end{cases}$$

This "difference" operation may be extended to sets as follows:

$$\{\alpha_1, \cdots, \alpha_m\} \dot{-} \beta = \bigcup_{1 \leqq j \leqq m} (\alpha_j \dot{-} \beta), \; m \geqq 0;$$

$$\{\alpha_1, \cdots, \alpha_m\} \dot{-} \emptyset = \{\alpha_1, \cdots, \alpha_m\};$$

$$\{\alpha_1, \cdots, \alpha_m\} \dot{-} \{\beta_1, \cdots, \beta_n\} = (\{\alpha_1, \cdots, \alpha_m\} \dot{-} \{\beta_1, \cdots, \beta_{n-1}\}) \dot{-} \beta_n, \quad n \geqq 1.$$

It is not difficult to verify that $(\alpha \dot{-} \beta) \dot{-} \gamma = (\alpha \dot{-} \gamma) \dot{-} \beta$, so the definition of $\{\alpha_1, \cdots, \alpha_m\} \dot{-} \{\beta_1, \cdots, \beta_n\}$ does not depend on the order of the β's. Furthermore $\{\alpha_1, \cdots, \alpha_m\} \dot{-} \{\beta_1, \cdots, \beta_n\} = \bigcup_{1 \leqq j \leqq m} (\alpha_j \dot{-} \{\beta_1, \cdots, \beta_n\})$.

Let us now proceed to construct a grammar $\mathcal{G}$ for $L(\mathcal{G}_1) - L(\mathcal{G}_2)$. We define $\tau(a) = a$ for $a \in \Sigma$, $\tau(A) = [A, \emptyset]$ for $A \in N_1$, and $\tau(x_1x_2 \cdots x_n) = \tau(x_1)\tau(x_2)\cdots\tau(x_n)$ for $x_1x_2 \cdots x_n$ in V_1^*. This embeds V_1^* into V^*. Now $\mathcal{G} = (\Sigma, V, \mathcal{S}, \mathcal{P})$ where

$$\mathcal{S} = \tau(\mathcal{S}_1) \dot{-} \mathcal{S}_2;$$

$$\mathcal{P} = \{[A, \mathfrak{B}] \to \theta \mid \theta \in \tau(\rho(A)) \dot{-} \rho(\mathfrak{B})\},$$

where $\rho(A) = \{\theta \mid A \to \theta\}$, $\rho(\mathfrak{B}) = \{\theta \mid B \to \theta, \text{ some } B \in \mathfrak{B}\}$.

Let $\to_1$, $\to_2$, $\to$ denote respectively the production relations in $\mathcal{G}_1$, $\mathcal{G}_2$, and $\mathcal{G}$. In order to show that $L(\mathcal{G}) = L(\mathcal{G}_1) - L(\mathcal{G}_2)$, we will prove that for every terminal string $\theta \in \Sigma^*$, $[A, \mathfrak{B}] \to^* \theta$ *if and only if* $A \to_1^* \theta$ *and* $B \nrightarrow_2^* \theta$ *for all* $B \in \mathfrak{B}$. This fact is almost obvious, yet it is almost impossible to explain in a few words; it is hoped that the reader will see (after the long explanation which follows) why it is obvious. First assume $[A, \mathfrak{B}] \to^* \theta$. By replacing all pairs $[A', \mathfrak{B}']$ by A' in this derivation, we obtain a derivation $A \to_1^* \theta$ in $\mathcal{G}_1$. Let this derivation be $A \to_1 \varphi \to_1^* \theta$. Now since we have a parenthesis grammar, $\varphi = (\alpha_1 A_1 \alpha_2 A_2 \cdots \alpha_k A_k \alpha_{k+1})$ where $k \geqq 0$ and $\alpha_1\alpha_2 \cdots \alpha_{k+1} \in T^*$ and $A_1A_2 \cdots A_k \in N_1^*$; furthermore $\theta = (\alpha_1\theta_1\alpha_2\theta_2 \cdots \alpha_k\theta_k\alpha_{k+1})$ where $A_j \to_1^* \theta_j$, $1 \leqq j \leqq k$. Assume there is some $B \in \mathfrak{B}$ such that $B \to_2^* \theta$; then by the

same reasoning $B \to_2 \chi = (\alpha_1 B_1 \alpha_2 B_2 \cdots \alpha_k B_k \alpha_{k+1})$ and $B_j \to_2^* \theta_j$, $1 \leqq j \leqq k$. Similarly $[A, \mathfrak{B}] \to \psi = (\alpha_1 [A_1, \mathfrak{B}_1] \cdots [A_k, \mathfrak{B}_k] \alpha_{k+1})$ and $[A_j, \mathfrak{B}_j] \to^* \theta_j$, $1 \leqq j \leqq k$. In particular $\varphi \sim \chi \sim \psi$, and by the construction of $\mathcal{G}$ we know that $\chi \in \rho(\mathfrak{B})$ and $\psi \in \tau(\rho(A)) \dot{-} \rho(\mathfrak{B}) = \bigcup \{\tau(\sigma) \dot{-} \rho(\mathfrak{B}) \mid \sigma \in \rho(A)\}$. Since $A_1, \cdots, A_k$ appear as the first components of the nonterminal symbols of ψ, and the $\dot{-}$ operation does not affect these components, we must have $\psi \in \tau(\varphi) \dot{-} \rho(\mathfrak{B})$. Hence by the definition of $\tau(\varphi) \dot{-} \rho(\mathfrak{B})$, we must have $B_j \in \mathfrak{B}_j$ for some j. But by induction on the length of derivation, $[A_j, \mathfrak{B}_j] \to^* \theta_j$ implies $B_j \twoheadrightarrow_2^* \theta_j$, and this is a contradiction.

Conversely assume $A \to_1^* \theta$, and $B \twoheadrightarrow_2^* \theta$ for all $B \in \mathfrak{B}$. Then as above there is some $\varphi \in \rho(A)$ having the form $(\alpha_1 A_1 \cdots A_k \alpha_{k+1})$, where $A_j \to_1^* \theta_j$ and $\theta = (\alpha_1 \theta_1 \cdots \theta_k \alpha_{k+1})$. Let $\rho(\mathfrak{B}) = \{\chi_1, \chi_2, \cdots, \chi_r\}$ and consider the set $\mathfrak{I}_q = \tau(\varphi) \dot{-} \{\chi_1, \cdots, \chi_q\}$, $0 \leqq q \leqq r$. It suffices to show by induction on q that there is some element $\psi \in \mathfrak{I}_q$ such that $\psi \to^* \theta$, for then we have $[A, \mathfrak{B}] \to \psi \to^* \theta$ for some ψ in $\mathfrak{I}_r$. The condition is obvious when $q = 0$ because $\mathcal{G}_1$ is actually embedded in $\mathcal{G}$; i.e., $\sigma \to_1^* \sigma'$ certainly implies $\tau(\sigma) \to^* \tau(\sigma')$. When $q > 0$, suppose $\chi_q \not\sim \varphi$; then $\mathfrak{I}_q = \mathfrak{I}_{q-1}$. If $\chi_q \sim \varphi$ then we know by hypothesis that $\chi_q = (\alpha_1 B_1 \cdots B_k \alpha_{k+1})$ where $B_j \twoheadrightarrow_2^* \theta_j$ for some j. Take $\psi \in \mathfrak{I}_{q-1}$ such that $\psi \to^* \theta$; then if $\psi = (\alpha_1 [A_1, \mathfrak{B}_1] \cdots [A_k, \mathfrak{B}_k] \alpha_{k+1})$ the element ψ' obtained from ψ by replacing $\mathfrak{B}_j$ by $(\mathfrak{B}_j \cup \{B_j\})$ is in $\mathfrak{I}_q$. By induction on the length of derivation, $[A_j, \mathfrak{B}_j \cup \{B_j\}] \to^* \theta_j$ so $\psi' \to^* \theta$.

To complete the proof that $L(\mathcal{G}) = L(\mathcal{G}_1) - L(\mathcal{G}_2)$, note that the arguments just stated amount to a proof that $\alpha \to \theta \in \Sigma^*$ for some α in $\tau(\{\sigma_1, \cdots, \sigma_m\}) \dot{-} \{\rho_1, \cdots, \rho_n\}$ if and only if there is some $t \leqq m$ such that $\sigma_t \to_1^* \theta$, yet $\rho_s \twoheadrightarrow_2^* \theta$ for $1 \leqq s \leqq n$. Therefore the construction of $\mathcal{S}$ leads to precisely the strings of $L(\mathcal{G}_1) - L(\mathcal{G}_2)$.

As an example of this construction, let

$$
\begin{aligned}
T &= \{a, b, c, d\},\ V_1 = \{X, Y\},\ V_2 = \{A, B\},\\
\mathcal{S}_1 &= \{aXY, a\},\ \mathcal{S}_2 = \{aAB, aBB, b\},\\
\mathcal{P}_1 &= \{X \to (YaX),\ X \to (b),\\
&\qquad Y \to (XcX),\ Y \to (d)\},\\
\mathcal{P}_2 &= \{A \to (BaA),\ A \to (b),\\
&\qquad B \to (AcB),\ B \to (BcA),\\
&\qquad B \to (b),\ B \to (d)\}.
\end{aligned}
$$

Using the notation $[X]$ to stand for $[X, \emptyset]$ and $[X; B_1, \cdots, B_n]$ to stand for $[X, \{B_1, \cdots, B_n\}]$, we have the following grammar $\mathcal{G}$ for $L(\mathcal{G}_1) - L(\mathcal{G}_2)$:

$$
\begin{aligned}
V &= \{[X], [X;A], [X;B], [X;A,B], [Y], [Y;B]\},\\
\mathcal{S} &= \{a[X;A,B][Y], a[X;A][Y;B], a[X;B][Y;B], a[X][Y;B], a\},\\
\mathcal{P} &= \{[X] \to ([Y]a[X]), [X] \to (b),\\
&\quad [X;A] \to ([Y;B]a[X]), [X;A] \to ([Y]a[X;A]),\\
&\quad [X;B] \to ([Y]a[X]),\\
&\quad [X;A,B] \to ([Y;B]a[X]), [X;A,B] \to ([Y]a[X;A]),\\
&\quad [Y] \to ([X]c[X]), [Y] \to (d),\\
&\quad [Y;B] \to ([X;A]c[X]), [Y;B] \to ([X]c[X;B]),\\
&\quad [Y;B] \to ([X;B]c[X]), [Y;B] \to ([X]c[X;A])\}.
\end{aligned}
$$

The nonterminal symbols $[Y;A]$ and $[Y;A,B]$ are useless so they have been omitted.

COROLLARY 1. *The set of parenthesis languages is closed under intersection.*

Proof. $L_1 \cap L_2 = L_1 \cup L_2 - (((L_1 \cup L_2) - L_1) \cup ((L_1 \cup L_2) - L_2))$. Alternatively, it is possible to construct $L_1 \cap L_2$ in a natural way: If $x_1 \cdots x_n \sim y_1 \cdots y_n$, define $x_1 \cdots x_n \cap y_1 \cdots y_n$ to be $(x_1 \cap y_1) \cdots (x_n \cap y_n)$ where $a \cap a = a$ for $a \in \Sigma$; $A \cap B = [A,B]$ for $A \in V_1$, $B \in V_2$. The details are straightforward.

COROLLARY 2. *If $\mathcal{G}_1$ and $\mathcal{G}_2$ are parenthesis grammars, there is an algorithm to decide if $L(\mathcal{G}_1) = L(\mathcal{G}_2)$.*

Proof. The construction in the proof of theorem 5 yields grammars for $L(\mathcal{G}_1) - L(\mathcal{G}_2)$ and $L(\mathcal{G}_2) - L(\mathcal{G}_1)$, and it is easy to test if these languages are empty. (For typical languages, however, the construction of McNaughton (1967) may lead to a simpler way to test this special condition.)

COROLLARY 3. *The complement of a parenthesis language is a context-free language.*

Proof. By theorem 4, every parenthesis language over Σ is contained in some language $P_n(\Sigma)$ given by the grammar

$$\mathcal{G} = (\Sigma, V, \mathcal{S}, \mathcal{P}),$$

$$V = \Sigma \cup \{A\}$$
$$S = \{\sigma \mid \sigma \in (T \cup \{A\})^* \quad \text{and} \quad |\sigma| \leqq n\}$$
$$\mathcal{P} = \{A \to (\theta) \mid \theta \in S\}$$

This grammar defines the set of all parenthesized formulas over Σ with associates bounded by n. If $L \subseteq P_n(\Sigma)$, then $\Sigma^* - L = (\Sigma^* - P_n(\Sigma)) \cup (P_n(\Sigma) - L)$, hence by theorem 5 it suffices to show $\Sigma^* - P_n(\Sigma)$ is context-free. A grammar for the latter is readily constructed, e.g.

$$(\Sigma, \Sigma \cup \{A, B, C, X, Z\}, \{A)X, X(A, B\}, \mathcal{P})$$

where $\mathcal{P}$ consists of the productions

$$A \to Y; A \to Y(A)A;$$
$$B \to C^{n+1}A; B \to A(B)A$$
$$C \to Z; C \to Y(A)A;$$
$$X \to \epsilon; X \to aX, \quad \text{all} \quad a \in \Sigma;$$
$$Y \to \epsilon; Y \to aY, \quad \text{all} \quad a \in T;$$
$$Z \to aY, \quad \text{all} \quad a \in T.$$

Here $L(A) = \{\text{balanced strings}\}$, $L(B) = \{\text{balanced strings with } > n \text{ associates}\}$, $L(C) = L(A) - \epsilon$, $L(X) = \Sigma^*$, $L(Y) = T^*$, $L(Z) = T^+$.

COROLLARY 4. *If $\mathcal{G}_1$ is a context-free grammar and $\mathcal{G}_2$ is a parenthesis grammar, there is an algorithm to decide whether or not $L(\mathcal{G}_1) \subseteq L(\mathcal{G}_2)$.*

Proof. If $L(\mathcal{G}_1) \subseteq L(\mathcal{G}_2)$ then $L(\mathcal{G}_1)$ is balanced and has bounded associates, so $L(\mathcal{G}_1)$ must be a parenthesis language. Therefore we may test whether $L(\mathcal{G}_1)$ is a parenthesis language, and then whether $L(\mathcal{G}_1) - L(\mathcal{G}_2)$ is empty.

The related problem, whether $L(\mathcal{G}_2) \subseteq L(\mathcal{G}_1)$, is *unsolvable* in general, is shown by the following construction. Let $\alpha_1, \cdots, \alpha_n, \beta_1, \cdots, \beta_n$ be nonempty strings on the alphabet $\{a, b\}$. Let L_0 be the language over the alphabet $\{(,), a, b, c_1, \cdots, c_n\}$ derivable from S in the grammar

$$\left.\begin{array}{l} S \to S), \\ S \to (x_{j1}(x_{j2} \cdots (x_{jk_j}Sc_j \\ S \to (x_{j1}(x_{j2} \cdots (x_{jk_j}c_j \end{array}\right\} \text{if } \alpha_j = x_{j1} \cdots x_{jk_j},\ 1 \leqq j \leqq n.$$

Let L_2 be the parenthesis language over the same alphabet derivable

from S in the grammar

$$\left.\begin{array}{l} S \to (x_{j1}(x_{j2} \cdots (x_{jt_j} S c_j)^{t_j} \\ S \to (x_{j1}(x_{j2} \cdots (x_{jt_j} c_j)^{t_j} \end{array}\right\} \text{ if } \beta_j = x_{j1} \cdots x_{jt_j},\ 1 \leqq j \leqq n.$$

Clearly the Post correspondence problem for $\alpha_1, \cdots, \alpha_n, \beta_1, \cdots, \beta_n$ has a solution if and only if $L_0 \cap L_2 \neq \emptyset$. It is also easy to verify that L_1, the complement of L_0, is context-free. Therefore the Post correspondence problem has a solution if and only if $L_2 \nsubseteq L_1$.

ACKNOWLEDGEMENT

The author wishes to express deep appreciation to the referee of this paper for his careful reading of the manuscript and his many helpful suggestions about the style of presentation.

RECEIVED: January 15, 1966

REFERENCES

BAR HILLEL, PERLES, AND SHAMIR (1961), On Formal Properties of Simple Phrase Structure Grammars, *Z. Phonetik, Sprachwiss. Kommunikationsforsch.* **14**, 143–172.

GINSBURG, S. AND HARRISON, M. A. (1967), Bracketed Context-Free Languages, *J. Computer and System Sci.* **1**, 1–23.

MCNAUGHTON, R. (1967), Parenthesis Grammars, *J. Assoc. Computing Mach.* **14**, 490–500.