Parenthesis Grammars

ROBERT MCNAUGHTON

Rensselaer Polytechnic Institute, Troy, New York*

ABSTRACT. A decision procedure is given which determines whether the languages defined by two parenthesis grammars are equal.

Languages that are rather familiar to students of elementary logic are considered in this paper. They are languages in which ambiguity is avoided by the systematic and tedious use of parentheses, so that a sentence (or terminal string) wears its syntactic structure on its sleeve. In the derivation exactly one pair of parentheses is introduced at every application of every rule. These parentheses, however, are of just one species and have no subscripts. Thus the parentheses do not tell which rule introduced them, or even which nonterminal they came from. In this respect they differ from the "bracketed languages" of Ginsburg and Harrison [1], in which subscripts on brackets go further toward indicating the structure of a terminal string.

The main result of this paper is that the equivalence problem for parenthesis languages (given by their grammars) is solvable. In other words, there is a decision procedure to tell whether two parenthesis grammars generate the same language. Incidentally, this shows that the inclusion problem is also solvable: for to determine whether the language of one grammar is included in the language of the second, one can take the grammar for the union of the two languages (easily obtainable) and test for equality with the second.

The proof of the existence of the decision procedure is rather involved; various attempts to simplify it have failed. Thus the class of parenthesis languages is a class for which the equivalence problem is decidable, but nontrivial. (In contrast, the decidability of the equivalence problem for the bracketed languages of [1] is elementary, because of the subscripts on the brackets.) The proof given in this paper, therefore, should provide an interesting case history for those who are doing research on the equivalence problem for more general languages. The question of the extendibility of the decision procedure to certain broader classes of languages is open.

It is necessary to begin the technical exposition with precise definitions.

A *context-free grammar* consists of a set of *nonterminal symbols*, a set of *terminal symbols*, a set of *initial symbols*, which is a subset of the set of nonterminal symbols, and a set of rules (or productions) of the form

$$A \rightarrow \Omega$$

* Mathematics Department

The work reported herein was supported by Project MAC, an MIT research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Number Nonr-4102(01). Reproduction in whole or in part is permitted for any purpose of the United States Government.

where A is a nonterminal and Ω is a nonempty string of terminals and/or nonterminals. (This definition differs from the usual definition of context-free grammar, in which there must be exactly one initial symbol, which is S. It turns out to be convenient, though not necessary, to allow parenthesis grammars several initial symbols.)

A string Ω of terminal and/or nonterminals is *derivable* in n lines from a nonterminal A in a grammar if there is a sequence of strings $\Omega_1, \cdots, \Omega_n$, where Ω_1 is A, Ω_n is Ω and, for every i, Ω_{i+1} follows from Ω_i by one of the rules; if in addition A is an initial nonterminal of the grammar we say simply that Ω is *derivable* in the grammar. The *language* of a grammar is the set of terminal strings derivable. Two grammars are *equivalent* if their languages are equal.

Capital letters are used in this paper for nonterminals and often also as variables ranging over nonterminals. Thus it is convenient to say "for every nonterminal A," etc. Small letters are used for terminal letters; in parenthesis grammars the open parenthesis (and close parenthesis) are also terminal symbols. Ω is used as a variable over strings.

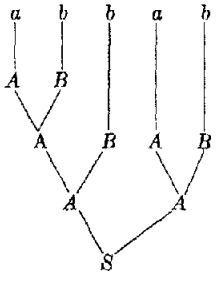
A derivation tree corresponding to a derivation is a labeled tree with a root labeled with the initial nonterminal of the derivation, and where every application of a rule is indicated by branching; thus the application of the rule

$$A \rightarrow \Omega$$

where Ω has m symbols is indicated by m branches from the node labeled A to nodes labeled with the symbols of Ω, in order. For example, a derivation tree for the derivation of *abbab* in the grammar whose initial symbol is S and whose rules are

$$S \rightarrow AA$$
$$A \rightarrow AB$$
$$A \rightarrow a$$
$$B \rightarrow b$$

is:



Although there is no other derivation tree for this terminal string, there are several derivations that result from the tree, since the order of replacing nonterminals currently appearing in a string of the derivation is arbitrary.

The derivation tree is more theoretically important in many respects than the derivation. Thus an *unambiguous* grammar is defined as one in which every terminal string in the language has exactly one derivation tree.

A *parenthesis grammar* is a context-free grammar all of whose rules are of the form

$$A \rightarrow (\Omega)$$

where Ω contains no occurrence of (or of). A *backwards-deterministic* parenthesis grammar is one in which no two rules have the same right side. A (*backwards-deterministic*) *parenthesis language* is one that has a (backwards-deterministic) parenthesis grammar.

Note that a backwards-deterministic parenthesis grammar is unambiguous. What is more, such a grammar has an even stronger property concerning parsing; i.e., given any string, we can pick out certain phrases of limited size (namely, the innermost parenthesized parts) and replace them by a nonterminal without any regard to what occurs to the left or to the right, and be confident that we have not made any mistake; this is the reason for the choice of the phrase "backwards-deterministic." At the end of this paper a more general definition is given of "backwards-deterministic" applying to languages without parentheses. (The term "backwards-deterministic" and the concept come from some unpublished work by K. Speierman.)

Before proceeding with the theoretical development that leads to a decision procedure for equivalence, we note that there are several variations on the precise concept of parenthesis languages introduced above that might seem more natural to some. For one thing, the present definition is such that the class of parenthesis languages is not closed under concatenation: if L_1 and L_2 are parenthesis languages L_1L_2 is not. One could, if one desired, allow strings like $(a(ba)b)(ab)$ in a parenthesis language by slightly modifying the technical definition. Then with only minor changes in the technical development the major results of this paper would be intact.

THEOREM 1. *Every parenthesis grammar has an equivalent backwards-deterministic parenthesis grammar effectively obtainable from it.*

PROOF. Let G be a parenthesis grammar for a parenthesis language L. Suppose that $A_1, \cdots, A_n$ are the nonterminals of this grammar. Let the *stencil* of a rule be the right side of that rule with blank spaces in place of the nonterminals. Thus, e.g., the stencil of the rule $A_1 \to (A_2bcA_3A_4)$ is $(\ \_\ bc\_\_\ \_\ )$.

A backwards-deterministic parenthesis grammar G' is now constructed having 2^n-1 nonterminals, one corresponding to each nonempty set of nonterminals of G. Let these be $B_1, \cdots, B_{2^n-1}$, and let $\sigma(B_1), \cdots, \sigma(B_{2^n-1})$ be the corresponding sets.

$$B_{i_1} \to (B_{i_2}bcB_{i_3}B_{i_4})$$

is a rule of G' if and only if $\sigma(B_{i_1})$ is precisely the set of nonterminals A_{j_1} of G such that, for some $A_{j_2} \in \sigma(B_{i_2})$, $A_{j_3} \in \sigma(B_{i_3})$, and $A_{j_4} \in \sigma(B_{i_4})$,

$$A_{j_1} \to (A_{j_2}bcA_{j_3}A_{j_4})$$

is a rule of G. And similarly for other stencils. Thus the set of stencils of rules of G' is exactly the set of stencils of rules of G. The initial nonterminals of G' are all those B_i such that there is at least one initial nonterminal of G in $\sigma(B_i)$.

It is clear that G' is a backwards-deterministic parenthesis grammar. We must show that its language is that of G. The proofs of Lemmas 1 and 2 complete the proof of Theorem 1.

LEMMA 1. *Every derivation D of a terminal string in G can be converted into a derivation D' of the same terminal string in G'.*

PROOF. Rewrite the derivation backwards line by line. The last replacement in

D might be an application of the rule

$$A_1 \rightarrow (ab).$$

The rule in G' to use is the unique rule

$$B_i \rightarrow (ab)$$

with the stencil (ab). This determines the penultimate line of D', and by definition $A_1 \in \sigma(B_i)$. Now suppose the $(x + 1)$-th line of D' has been determined. Suppose, as an inductive hypothesis, that the $(x + 1)$-th line of D' is like the $(x + 1)$-th line of D except that for each occurrence of each nonterminal A_z of G, there is in the corresponding position a nonterminal B_y of G', where $A_z \in \sigma(B_y)$. (Different occurrences of A_z may have different B_y's.) The xth line of D' can be determined from the $(x + 1)$-th line of D' by considering the rule used in going from the xth to the $(x + 1)$-th line in D. Assume, for example, that this rule is

$$A_1 \rightarrow (A_2bcA_3A_4).$$

Suppose there is corresponding to $(A_2bcA_3A_4)$ in the $(x + 1)$-th line of D the phrase $(B_{i_2}bcB_{i_3}B_{i_4})$ in the $(x + 1)$-th line of D' as already constructed, where $A_2 \in \sigma(B_{i_2})$, $A_3 \in \sigma(B_{i_3})$, and $A_4 \in \sigma(B_{i_4})$. There is a unique B_{i_1} such that

$$B_{i_1} \rightarrow (B_{i_2}bcB_{i_3}B_{i_4})$$

is a rule of G', where $A_1 \in \sigma(B_{i_1})$. We thus make the xth line of D' like the $(x + 1)$-th line of D' except that B_{i_1} replaces $(B_{i_2}bcB_{i_3}B_{i_4})$ in its occurrence corresponding to the replacement in the $(x + 1)$-th line of D. Clearly the xth line of D' will have the required properties.

If D' is constructed backwards in this manner, the first line will be B_y where, if A_z is the first line of D, $A_z \in \sigma(B_y)$. B_y will be initial in G' since A_z is initial in G. Thus D' is a derivation of the same string in the grammar G'.

LEMMA 2. *Every derivation D' of a terminal string in G' can be converted into a derivation D of the same terminal string in G.*

PROOF. This time the derivation is to be rewritten forward. If the first line of D' is the initial nonterminal B_y there exists an A_z such that $A_z \in \sigma(B_y)$ and A_z is an initial nonterminal of G. (If there are several such A_z's it does not matter which one is selected.) As an inductive hypothesis, suppose that the xth line of D is like the xth line of D' except for containing in place of each B_y an occurrence of some A_z in the corresponding place, where $A_z \in \sigma(B_y)$. (Different occurrences of B_y may have corresponding to them different A_z's.) Suppose, for example, that the rule

$$B_{i_1} \rightarrow (B_{i_2}bcB_{i_3}B_{i_4})$$

is the rule by means of which the $(x + 1)$-th line of D' is obtained from the xth line of D', and suppose that A_{j_1} occurs in the xth line of D corresponding to that occurrence of B_{i_1} in the xth line of D'. Then $A_{j_1} \in \sigma(B_{i_1})$ and hence there exist nonterminals $A_{j_2} \in \sigma(B_{i_2})$, $A_{j_3} \in \sigma(B_{i_3})$, and $A_{j_4} \in \sigma(B_{i_4})$ such that

$$A_{j_1} \rightarrow (A_{j_2}bcA_{j_3}A_{j_4})$$

is the rule of G that gives us the $(x + 1)$-th line of D satisfying the inductive hypothesis. In this manner D is constructed so that it is a derivation of G and its

last line (since there are no nonterminals) is exactly the same as the last line of D'. This concludes the proof of Theorem 1.

Where A is a nonterminal of a grammar G, let $\Delta_G(A)$ be the set of all strings derivable from A (including those with nonterminals).

THEOREM 2. *If A and B are distinct nonterminals of a backwards-deterministic parenthesis grammar G, then $\Delta_G(A)$ and $\Delta_G(B)$ are disjoint.*

PROOF. A string in a backwards-deterministic parenthesis grammar has at most one derivation tree. Hence no string could have both A at the root of one derivation tree and B at the root of another.

A backwards-deterministic parenthesis grammar may have two nonterminals whose combined role can be played by a single nonterminal. If such is the case we say that the grammar is not "reduced," in a sense to be defined more precisely below. Theorem 4 states that there is an algorithm to reduce such a grammar. Toward that objective some auxiliary concepts are now defined.

A *context* β is defined to be a string with one blank, and we write $\beta[\Omega]$ for the context β with its blank replaced by the string Ω. Let Σ be a set of nonterminals of a grammar G. We say Σ is *n-distinguishable* in G if there is a context β such that $\beta[A]$ is derivable in n lines or less in G from an initial nonterminal if and only if $A \in \Sigma$. Σ is *distinguishable in G* if it is n-distinguishable, for some n. Note that, in a parenthesis grammar, the length of a derivation is one more than the number of parenthesis pairs in the string. Thus all derivations of a string of the form $\beta[A]$ must be of the same length. The following theorem is the first step in establishing the existence of an algorithm for reducing a backwards-deterministic parenthesis grammar.

THEOREM 3. *There is an algorithm to determine, for any set Σ of nonterminals of a backwards-deterministic parenthesis grammar, whether or not Σ is distinguishable.*

PROOF. Clearly, for any n, there is an algorithm to determine whether a set Σ is n-distinguishable: one has simply to list all derivations of length n or less. Thus Theorem 3 will follow once we establish the proposition that, if r is the number of nonterminals of the grammar, then a set of nonterminals is distinguishable if and only if it is 2^r-distinguishable. But an elementary mathematical argument shows that this proposition follows in turn from the following lemma, whose proof concludes the proof of Theorem 3.

LEMMA 3. *For any n, if there is a Σ that is $(n + 2)$-distinguishable but not $(n + 1)$-distinguishable, then there is a Σ_0 that is $(n + 1)$-distinguishable but not n-distinguishable.*

PROOF. Let β_{n+2} be a context with $n + 1$ or fewer parenthesis pairs such that Σ is precisely the set of all nonterminals A such that $\beta_{n+2}[A]$ is derivable in $n + 2$ lines or less in G. Since Σ is not $(n + 1)$-distinguishable, β_{n+2} must contain exactly $n + 1$ parenthesis pairs.

We establish now that the blank in β_{n+2} must be inside a parenthesis phrase that does not have another parenthesis phrase within it. Suppose, on the contrary, that the blank is inside a parenthesis phrase of β_{n+2} of the form

$$(\Omega_1(\Omega_2)\Omega_3 \_\_ \Omega_4)$$

or

$$(\Omega_1 \quad \Omega_3(\Omega_2)\Omega_4)$$

where Ω_2 is nonnull. (Ω_2) can come from only one possible nonterminal, say C, since the grammar is backwards-deterministic. If (Ω_2) in β_{n+2} is replaced by C, the result is another context with fewer parenthesis pairs that distinguishes Σ, contradicting the assumption that Σ is not $(n + 1)$-distinguishable.

Having established that the parenthesis pair containing the blank contains no other parenthesis pair, we assume for the sake of exposition that the parenthesis phrase containing the blank has the form $(A_1bc \_\_ A_3)$. Let β_{n+1} be the context which is like β_{n+2} except for having a single blank in place of this whole parenthesis phrase. Since β_{n+2} has $n + 1$ parenthesis pairs, β_{n+1} has n parenthesis pairs. Without loss of generality, we can confine our attention to derivations of $\beta_{n+2}[A]$, for any A, in which this parenthesized phrase is the last to appear. Thus the line before will be $\beta_{n+1}[B]$, for some B.

Let Σ_0 be the set of all nonterminals B such that $\beta_{n+1}[B]$ is derivable. The proof is complete if we can show that Σ_0 is $(n + 1)$-distinguishable (which is obvious), but not n-distinguishable.

Suppose now that Σ_0 is n-distinguishable. There would be a context β_n' with $n - 1$ or fewer parenthesis pairs such that Σ_0 is the set of all B such that $\beta_n'[B]$ is derivable in G. Recall that Σ is the set of all nonterminals A such that

$$B \rightarrow (A_1bcAA_3)$$

is a rule of G for some $B \in \Sigma_0$. If we take β_{n+1}' to be the context $\beta_n'[(A_1bc \_\_ A_3)]$, Σ is the set of all nonterminals A such that $\beta_{n+1}'[A]$ is derivable. From the fact that there are n or fewer parenthesis pairs in β_{n+1}', it would follow that Σ is $(n + 1)$-distinguishable, a contradiction, concluding the proof of Lemma 3 and the proof of Theorem 3.

Two nonterminals A_1 and A_2 of a parenthesis grammar G are *equivalent* if, for every context β, either both $\beta[A_1]$ and $\beta[A_2]$ are derivable in G or neither are. (Clearly, this is an equivalence relation.) Thus A_1 and A_2 are equivalent if and only if there is no distinguishable Σ having one of these without the other. An equivalence class of nonterminals is a nonempty set having all and only all the nonterminals equivalent to some given nonterminal. A grammar is *reduced* if no two distinct nonterminals are equivalent, and if it has no useless nonterminals; a nonterminal A is *useless* in G if there is no context β such that $\beta[A]$ is derivable or if there is no string of terminals derivable from A. It is well known (and, in fact, obvious) that (1) any useless nonterminal, and every rule containing that nonterminal, can be discarded from the grammar without changing the language of the grammar, and that (2) there is an algorithm to determine the useless nonterminals.

(The reader who is familiar with switching theory will recognize the concept of a reduced grammar as resembling the concept of a reduced state graph. The reduction procedure of Theorem 4 is like the reduction procedure for state graphs. Lemma 3 plays a role in reducing a grammar that corresponds, in the reduction procedure for sequential machines, to the lemma that if two states are distinguishable by an experiment of length $n + 2$, but not by an experiment of length $n + 1$, then there are two states that are distinguishable by an experiment of length $n + 1$ but not by an experiment of length n.)

THEOREM 4. *There is an algorithm to obtain a reduced backwards-deterministic parenthesis grammar with the same language as a given backwards-deterministic parenthesis grammar.*

PROOF. Let G be a backwards-deterministic parenthesis grammar. Without loss of generality, assume G has n nonterminals none of which is useless. By use of Theorem 3, the list of all distinguishable sets of nonterminals can be listed, since there are only finitely many sets of nonterminals. But this information also gives us the equivalence classes of nonterminals: let these be, without repetition, $\Sigma_1, \cdots, \Sigma_m$. Form the grammar G' with the nonterminals $B_1, \cdots, B_m$. For every rule of G

$$A_1 \rightarrow (A_2 bc A_3 A_4)$$

take for G' the rule

$$B_{i_1} \rightarrow (B_{i_2} bc B_{i_3} B_{i_4})$$

where $A_1 \in \Sigma_{i_1}$, $A_2 \in \Sigma_{i_2}$, $A_3 \in \Sigma_{i_3}$, and $A_4 \in \Sigma_{i_4}$. B_i is an initial nonterminal if and only if there is an $A \in \Sigma_i$ which is an initial nonterminal in G. Note that if $A \in \Sigma_i$ is an initial nonterminal of G then every $A' \in \Sigma_i$ is initial. We now need two lemmas. Let us say that a string Ω of the vocabulary of G and a string Ω' of the vocabulary of G' are *corresponding strings* if they are the same length, and, for every x, (1) the xth symbol of Ω and the xth symbol of Ω' are the same terminal symbol, or (2) the xth symbol of Ω is a nonterminal A_i and the xth symbol of Ω' is a nonterminal B_j, where $A_i \in \Sigma_j$.

LEMMA 4. *If $\Omega_1, \cdots, \Omega_n$ is a derivation in G and if $\Omega_1', \cdots, \Omega_n'$ are respective corresponding strings of G', then $\Omega_1', \cdots, \Omega_n'$ is a derivation in G'.*

PROOF. Ω_1' must be an initial nonterminal of G', since Ω_1 is an initial nonterminal of G. Ω_{i+1}' must be the result of the application of a rule of G' to Ω_i', since Ω_{i+1} is the result of the application of a rule of G to Ω_i.

LEMMA 5. *If Ω' is a string derivable in G' and if Ω and Ω' are corresponding strings, then Ω is derivable in G.*

The proof is by induction on the length of the derivation of Ω' in G'. If the length is 1, Ω' must be an initial nonterminal B_j, and every $A_i \in \Sigma_j$ must be initial, by a remark above. Assume that Lemma 5 is true for all strings derivable in x lines, and let Ω_{x+1}' be derivable in $x + 1$ lines. Suppose Ω_{x+1}' results from Ω_x' in a derivation in G' by means of the rule

$$B_{i_1} \rightarrow (B_{i_2} bc B_{i_3} B_{i_4}).$$

By the inductive hypothesis, every Ω_x corresponding to Ω_x' is derivable in G. We know that, for some $A_1 \in \Sigma_{i_1}$, $A_2 \in \Sigma_{i_2}$, $A_3 \in \Sigma_{i_3}$, $A_4 \in \Sigma_{i_4}$,

$$A_1 \rightarrow (A_2 bc A_3 A_4)$$

is a rule of G. For this A_1, consider the class K_x of all Ω_x such that (1) Ω_x corresponds to Ω_x', and (2) A_1 in Ω_x takes the place of B_{i_1} in the noted occurrence in Ω_x'. From each $\Omega_x \in K_x$ we can obtain an Ω_{x+1} by the above rule from Ω_x so that Ω_{x+1} will correspond to Ω_{x+1}' and will have $(A_2 bc A_3 A_4)$ in place of $(B_{i_2} bc B_{i_3} B_{i_4})$ in the noted occurrence.

Now consider an arbitrary string corresponding to Ω_{x+1}'; we may write this string as $\Omega_x[(A_2' bc A_3' A_4')]$, where $\Omega_x[A_1]$ is in K_x. We know from what is proved in the above paragraph that $\Omega_x[(A_2 bc A_3 A_4)]$ is derivable in G. But then it follows that $\Omega_x[(A_2' bc A_3 A_4)]$ is derivable, since A_2' and A_2 are equivalent in G. From this it follows that $\Omega_x[(A_2' bc A_3' A_4)]$ and finally $\Omega_x[(A_2' bc A_3' A_4')]$ are derivable. Q.E.D.

It follows easily from Lemmas 4 and 5 that G' is equivalent to G. Clearly G' is a

parenthesis grammar. It remains to show that G' is (1) backwards-deterministic and (2) reduced.

(1) Suppose G' is not backwards-deterministic. Then G' has two rules,

$$B_{i_1} \rightarrow (B_{i_2}bcB_{i_3}B_{i_4}), \qquad B_{i_1'} \rightarrow (B_{i_2}bcB_{i_3}B_{i_4})$$

where $i_1 \neq i_1'$. Then G must have two rules,

$$A_{j_1} \rightarrow (A_{j_2}bcA_{j_3}A_{j_4}), \qquad A_{j_1'} \rightarrow (A_{j_2'}bcA_{j_3'}A_{j_4'})$$

where $A_{j_1} \in \Sigma_{i_1}$, $A_{j_1'} \in \Sigma_{i_1'}$, A_{j_2} and $A_{j_2'} \in \Sigma_{i_2}$, A_{j_3} and $A_{j_3'} \in \Sigma_{i_3}$, and A_{j_4} and $A_{j_4'} \in \Sigma_{i_4}$. Since A_{j_2}, A_{j_3}, and A_{j_4} are, respectively, equivalent to $A_{j_2'}$, $A_{j_3'}$, and $A_{j_4'}$, for any context β,

$$\beta[(A_{j_2}bcA_{j_3}A_{j_4})]$$

is derivable if and only if

$$\beta[(A_{j_2'}bcA_{j_3'}A_{j_4'})]$$

is derivable. But the phrase $(A_{j_2}bcA_{j_3}A_{j_4})$ must come from A_{j_1} and $(A_{j_2'}bcA_{j_3'}A_{j_4'})$ from $A_{j_1'}$, since G is backwards-deterministic, and we can suppose that these rules were used last in any derivation. So $\beta[A_{j_1}]$ is derivable if and only if $\beta[A_{j_1'}]$ is, for any context β; from which we conclude that A_{j_1} is equivalent to $A_{j_1'}$ and $\Sigma_{i_1'} = \Sigma_{i_1}$, contradicting the stipulation that $i_1 \neq i_1'$. Thus G' is backwards-deterministic.

(2) Suppose G' is not reduced. It is clear that, since G has no useless nonterminals, G' has no useless nonterminals. Thus, for some i and j, $i \neq j$ and, for every context β', $\beta'[B_i]$ is derivable in G' if and only if $\beta'[B_j]$ is derivable. Suppose $A_1 \in \Sigma_i$, $A_2 \in \Sigma_j$. Then, by Lemmas 4 and 5, it follows that, for every context β in G, $\beta[A_1]$ is derivable if and only if $\beta[A_2]$ is derivable; hence A_1 and A_2 are equivalent, and $\Sigma_i = \Sigma_j$, contradicting the stipulation that $i \neq j$. Thus G' is reduced, which concludes the proof of Theorem 4.

Two grammars are *isomorphic* if (1) they have the same set of terminals, (2) there is a one-to-one correspondence between the sets of nonterminals such that initial nonterminals in one grammar correspond to initial nonterminals in the other grammar, and (3) there is a one-to-one correspondence between rules of one and rules of the other, where a rule of one grammar can be obtained from the corresponding rule of the other by replacing each nonterminal in the second grammar by the corresponding nonterminal in the first.

THEOREM 5. *Two equivalent reduced backwards-deterministic parenthesis grammars are isomorphic.*

PROOF. Let G, G' be grammars satisfying the hypothesis of Theorem 5. Let A_1 be a nonterminal of G and let Δ be the set of terminal strings derivable from A_1. Δ is not empty (otherwise A_1 would be useless); let $\Omega \in \Delta$. Ω must be a part of a word of the language (again, since A_1 is not useless) beginning and ending with a pair of mated parentheses and hence must be derivable from some A_1' in G'. Let Δ' be the set of terminal strings derivable from A_1' in G'.

Now suppose $\Delta \neq \Delta'$. Then there would be a terminal string, say Ω_0, in one but not in the other. Suppose $\Omega_0 \in \Delta - \Delta'$. Then Ω_0 must be derivable from some A_2' in G', where $A_2' \neq A_1'$. Now Ω_0 and Ω are not derivable from any other nonterminal in G other than A_1, since G is backwards-deterministic. Hence, for any terminal

context β, $\beta[\Omega]$ is in the language if and only if $\beta[\Omega_0]$ is. From this fact it follows that, for every context β of G', terminal or otherwise, $\beta[A_1']$ is derivable in G' if and only if $\beta[A_2']$ is derivable; for, since G' has no useless nonterminals, a terminal string must be derivable from every derivable string. But this result contradicts the assumption that G' is reduced. Similarly, from the supposition that $\Omega_0 \in \Delta' - \Delta$ we can conclude that G is not reduced. Thus we infer that $\Delta = \Delta'$.

From this argument we conclude that each nonterminal in G has a corresponding nonterminal in G' from which the same set of terminal strings can be derived. This shows that there is a one-to-one correspondence between nonterminals. Since the languages of the two grammars are the same, initial nonterminals must correspond to initial nonterminals. It remains to show that there is the appropriate one-to-one correspondence between rules of G and rules of G'. Henceforth in the proof, let A_i' be the nonterminal in G' corresponding to A_i in G, for each i.

Suppose $A_1 \rightarrow (A_2bcA_3A_4)$ is a rule of G. Let Ω_2, Ω_3, and Ω_4 be terminal strings derivable from A_2, A_3, and A_4, respectively. $(\Omega_2bc\Omega_3\Omega_4)$ must be derivable from A_1. By what has been proved, Ω_2, Ω_3, Ω_4, and $(\Omega_2bc\Omega_3\Omega_4)$ are terminal strings derivable in G' from A_2', A_3', A_4', and A_1', respectively. But, because G' is a backwards-deterministic parenthesis grammar, the second line in a derivation of $(\Omega_2bc\Omega_3\Omega_4)$ from A_1' in G' must be $(A_2'bcA_3'A_4')$. It follows then that

$$A_1' \rightarrow (A_2'bcA_3'A_4')$$

is a rule of G'. This argument goes both ways, and suffices to show that there is the appropriate one-to-one correspondence between the two sets of rules. The two grammars are therefore isomorphic, Q.E.D.

The objective of this paper has now been achieved, namely, the proof of the Main Theorem, which clearly follows from Theorems 1 through 5. The corollary following it is justified in a remark at the beginning of this paper.

MAIN THEOREM. *There is a decision procedure to determine whether the languages of two given parenthesis grammars are equal.*

COROLLARY. *There is a decision procedure to determine whether one of two such languages is included in the other.*

A *concealed parenthesis grammar* is a grammar that is not a parenthesis grammar but generates a parenthesis language. The problem of how to recognize whether a given context-free grammar is a concealed parenthesis grammar and the problem of converting one into a parenthesis grammar are open. That these are not simple problems is illustrated in the following examples. A grammar whose language is not a parenthesis language, although it might seem so at first glance, is:

$$\begin{aligned} S &\rightarrow (A) \\ A &\rightarrow AA \\ A &\rightarrow (a) \end{aligned}$$

An example of a concealed parenthesis grammar, but not obviously so, is:

$$\begin{aligned} S &\rightarrow AB \\ A &\rightarrow ((a)A) \\ A &\rightarrow (c \\ B &\rightarrow (B(b)) \\ B &\rightarrow d) \end{aligned}$$

A parenthesis grammar equivalent to it is:

$$
\begin{aligned}
S &\rightarrow (cd) \\
S &\rightarrow (cZW) \\
S &\rightarrow (XYd) \\
S &\rightarrow (XYZW) \\
X &\rightarrow (a) \\
Y &\rightarrow (XY) \\
Y &\rightarrow (c) \\
Z &\rightarrow (ZW) \\
Z &\rightarrow (d) \\
W &\rightarrow (b)
\end{aligned}
$$

In all of these grammars the only initial nonterminal is S. (*Note added in proof:* In December, 1966, Donald E. Knuth announced the solution of these problems.)

Finally, it should be remarked that the concepts of this paper apply to some grammars that do not have parentheses at all. The well-known parenthesis-free notation is an example. Consider, for example, the following grammar for a fragment of Łukasiewicz' propositional calculus (in which S is the only nonterminal and is initial):

$$
\begin{aligned}
S &\rightarrow cSS \\
S &\rightarrow nS \\
S &\rightarrow p \\
S &\rightarrow q \\
S &\rightarrow r
\end{aligned}
$$

A terminal string in this language can be parsed in only one way, and in fact there is a method of parsing that assures us that we can proceed without making any false starts. The reason is that the grammar is backwards-deterministic, in a sense that can be defined precisely as follows: A context-free grammar is *backwards-deterministic* (in the general sense) if (1) it has no two distinct rules that have $\Omega_1\Omega_2$ and $\Omega_2\Omega_3$ as the respective right sides, for any strings Ω_1, Ω_2, Ω_3 where Ω_2 is nonnull, (2) it has no two distinct rules having Ω_2 and $\Omega_1\Omega_2\Omega_3$ as the respective right sides for any Ω_1, Ω_2, Ω_3, and (3) it has no rule with $\Omega_1\Omega_2\Omega_1$ as its right side where Ω_1 is nonempty. Thus in any string of terminals and nonterminals, there are no two parsing possibilities that interfere with each other. It is easy to see that for parenthesis grammars, this definition of "backwards-deterministic" is equivalent to the one used above.

A backwards-deterministic grammar is similar to what Johnston [2] calls a "fully nested computer language," and is identical to what Floyd [3] calls a "bounded context grammar of bound 0."

The grammar given above for the fragment of Lukasiewicz' propositional calculus is backwards-deterministic. Indeed it is possible to say that the essence of Łukasiewicz' contribution is the construction of an approximately backwards-deterministic language without parentheses. (There are problems converting the whole of a parenthesis-free propositional calculus into a backwards-deterministic language, for example, the problem of providing for an infinite number of propositional variables. These problems are not insurmountable, but it does not seem worthwhile to discuss them here. Parenthesis-free languages have bounded-context grammars with a small bound, in the sense of Floyd [3], and so it is not useful to convert them into backwards-deterministic languages.)

The proof in this paper for parenthesis grammars does not go over for general backwards-deterministic grammars. Although Theorem 2 is valid, Theorems 3 and 4 are doubtful, and Theorem 5 is false, for backwards-deterministic grammars. A counterexample to Theorem 5 is the above grammar for the fragment of Lukasiewicz' propositional calculus. It is backwards-deterministic and reduced, but the following equivalent reduced backwards-deterministic grammar is not isomorphic to it:

$$S \to XS$$
$$X \to cS$$
$$S \to nS$$
$$S \to p$$
$$S \to q$$
$$S \to r$$

Therefore the equivalence problem for general backwards-deterministic context-free languages is open, as it is for the more general class of bounded-context languages of [3].

ACKNOWLEDGMENT. The author is grateful to Professor Donald E. Knuth for his detailed criticism of an early draft of this paper.

REFERENCES

1. GINSBURG, S., AND HARRISON, M. A. Bracketed context-free languages. *J. Computer and System Sciences 1* (1967), 1-23.
2. JOHNSTON, JOHN B. A class of unambiguous computer languages. *Comm. ACM 8* (1965), 147-149.
3. FLOYD, ROBERT W. Bounded context syntactic analysis. *Comm. ACM 7* (1964), 62-67.

RECEIVED APRIL, 1966; REVISED AUGUST, 1966