BOUNDED CONTEXT PARSABLE GRAMMARS

by

John H. Williams

Technical Report #58

March 1969

TABLE OF CONTENTS

SECTION PAGE

Introduction 1

1. Definitions and Notation 3

2. Bounded Context Parsable Grammars 16

3. Relationship of Bounded Context Parsable Grammars with Other Subsets of Grammars 31

4. Testing for Bounded Context Parsability 41

Bibliography 46

INTRODUCTION

Context-free grammars have been found to be useful in the formal description of the syntax of programming languages [CHM], [FL3]. When designing a language and specifying its syntax, the designer would like to be able to know in advance that all the sentences in his language are unambiguous and that all the sentences in his language can be syntacticly analysed, or parsed, efficiently. Consequently, much work has been done ([FL1], [KNU], [LYN], [ERL], [WW]) to discover subsets of the set of context-free grammars for which membership in the subset is recursively decidable and for which membership in the subset implies that sentences generated by the grammar can be parsed in time linearly proportional to the length of the sentence, i.e. in "real time" [HST]. The goal of these investigations has been to discover subsets that are sufficiently large and unrestrictive so that the language designer may construct grammars that are in the subset without having to alter the desired constructs in his language or to introduce new syntactic types in his grammar in order to comply with the restrictions of the subset.

In this thesis we present a class of such subsets and call them the subsets of Bounded Context Parsable grammars. In section 1 we present the notation and previous work upon which the work of this thesis is built. In section 2 we give a formal definition of the Bounded Context Parsable property and show that any sentence of a Bounded Context Parsable grammar can be parsed in real time. In section 3 we compare the sets of Bounded Context Parsable grammars to some other general sets of grammars, and we show that the set of Bounded Context Parsable languages properly contains the set of deterministic languages [GIN]. In section 4 we show that the problem to determine whether a grammar is in a particular member of the class of subsets of Bounded Context Parsable grammars is effectively decidable but that the problem to determine whether a grammar is in any one of the class is recursively undecidable.

SECTION 1. DEFINITIONS AND NOTATION

By a grammar G, we shall mean a context-free grammar; the class of context-free grammars has been extensively studied, [BPS], [GIN], [CHM], with varying descriptional notations. Given a finite set X of characters, by X^* we shall mean the set of all strings over X including the empty string ϵ, and by X^+ we shall mean the set $X^* - \{\epsilon\}$. The number of elements in X will be denoted by $o(X)$. We shall express a grammar as a 4-tuple, $G = (V, P, V_T, S)$ where:

i) V is a finite set of symbols called the vocabulary of G.

ii) V_T is a subset of V called the terminal vocabulary of G. (we call the complement of V_T with respect to V the non-terminal vocabulary of G and denote it by V_N).

iii) P is a finite set of pairs of strings over V of the form (A, x) where $A \in V_N$ and $x \in V^+$. P is called the set of productions of G.

iv) $S \in V_N$ is called the sentence prototype of G.

We define the relation, $\rightarrow$, on V^* by: $\varphi \rightarrow \psi$ iff

i) $\varphi = x_1 A x_2$

ii) $\psi = x_1 x x_2$

iii) $x_1, x_2 \in V^*$ and $(A, x) \in P$

We will denote the transitive completion of $\rightarrow$ by $\rightarrow^{+}$ and the reflexive and transitive completion of $\rightarrow$ by $\rightarrow^{*}$. For purposes of identification we will order the set P and often write its ith member as $A_i \rightarrow x_i$. Such a string, x_i, will be called a phrase. A production of the form $A \rightarrow B$ where $B \in V_N$ will be called a 1-production. If $\varphi \rightarrow^{+} \psi$, then φ is said to derive ψ and ψ is said to be a φ-derivative. If $A \rightarrow x$ is a production, x is said to be an immediate A-derivative. If $\varphi \in V^{*}$, $\lambda(\varphi)$ will denote the length of φ, i.e. the number of characters in φ, and φ^{R} will denote the reversal or miror image of φ.

By the language of G is meant the set $L(G) = \{\varphi | S \rightarrow^{+} \varphi$ and $\varphi \in V_T^{*}\}$. If $\varphi \in L(G)$, φ is called a sentence of G.

By the sentential forms of G is meant the set $SF(G) = \{\varphi | S \rightarrow^{+} \varphi\}$. Clearly $L(G) \subseteq SF(G)$.

A grammar is said to be reduced if for every $A \in V_N$,

i) $(\exists\varphi)(\exists\psi)\, S \rightarrow^{*} \varphi A \psi;\ \varphi, \psi \in V^{*}$

ii) $(\exists x)\, A \rightarrow^{*} x;\ x \in V_T^{*}$.

We will always assume a grammar to be reduced unless specifically stated otherwise.

A grammar is said to be linear if every phrase contains at most one nonterminal character; clearly if G is linear, then every $\varphi \in SF(G)$ has at most one nonterminal character.

Given a grammar G we define its description grammar by

$$G' = (V', P', V'_T, S') \text{ where}$$

i) $V'_T = V_T \cup \{ [\} \cup \{]_i \mid 1 \le i \le o(P) \}$ where the $]_i$ and $[$ are new symbols not in V .

ii) $V' = V \cup V'_T$

iii) $S' = S$

iv) $P' = \{ A_i \to [\, x_i\,]_i \mid A_i \to x_i \in P \}$.

The language $L(G')$ will be called the description language of G . We define the mapping, $m : V'^* \to V^*$ by

$$m(A) = A, \text{ if } A \in V_N,$$

$$m(a) = a, \text{ if } a \in V_T,$$

$$m([) = \epsilon,$$

$$m(\epsilon) = \epsilon,$$

$$m(]_i) = \epsilon, \ (1 \le i \le o(P)),$$

and $m(\varphi) = m(x_1)\, m(x_2) \cdots m(x_n)$ if $\varphi = x_1 x_2 \cdots x_n$. Thus m is a homomorphism with respect to concatenation. Let $\overline{m}$ be the restriction of m to $SF(G')$. In general for $\varphi \in SF(G)$, $\overline{m}^{-1}(\varphi)$ will be a subset of $SF(G')$.

A grammar will be said to be unambiguous if $\overline{m}$ is $1:1$ and ambiguous otherwise.

If $A_i \rightarrow x_i \in P$, $\varphi \in SF(G)$, and $\varphi = \varphi_1 x_i \varphi_2$, the phrase x_i is said to be a handle of φ if there is a string in $m^{-1}(\varphi)$ of the form $\psi_1 [x_i]_i \psi_2$ such that $m(\psi_1) = \varphi_1$ and $m(\psi_2) = \varphi_2$.

To illustrate these concepts, consider the grammar

$$G_0 = (\{S, A, B, a, b\}, P_0, \{a, b\}, S) \text{ with } P_0 :$$

$$S \rightarrow bAaB$$

$$A \rightarrow aBA$$

$$A \rightarrow ab$$

$$B \rightarrow bA$$

$$B \rightarrow b$$

The phrases of the grammar are bAaB, aBA, ab, bA, and b . The grammar is ambiguous because both

$$\varphi_1 = [b[a[b]_5 \, [ab]_3 \,]_2 \, a \, [b]_5 \,]_1$$

and

$$\varphi_2 = [b[ab]_3 \, a \, [b \, [ab]_3 \,]_4 \,]_1$$

are mapped into bababab under $\overline{m}$.

Given a sentence φ , the process of computing $\overline{m}^{-1}(\varphi)$ is called parsing and a device for calculating $\overline{m}^{-1}(\varphi)$ is often called a parser for G . Parsers are usually constructed so that given an input, φ:

i) if $\varphi \in L(G)$, the parser outputs $\bar{m}^{-1}(\varphi)$

ii) if $\varphi \notin L(G)$, the parser outputs an error message.

The essence of the parsing process is to be able to decide which phrases of a sentential form are handles, make the appropriate reductions in the sentential form, and then repeat the process on the new sentential form thus obtained. Younger [YNG] has shown that for any grammar, G, there exists a parser for G that will parse any sentence $\varphi \in L(G)$ in a time proportional to $(\lambda(\varphi))^3$. Earley [ERL] has shown that for any unambiguous grammar G, there is a parser that will parse any $\varphi \in L(G)$ in time proportional to $(\lambda(\varphi))^2$. In computing we are interested in grammars whose sentences can be parsed in time proportional to $\lambda(\varphi)$ or in "real time" [HST]. In particular we wish to discover large subsets C of context-free grammars that possess the following properties:

Property 1) membership in C is recursively decidable,

Property 2) there is an effective procedure for producing a real time parser for any G in C.

One such important class of subsets is the class of Bounded Context (m, n), (BC (m, n)), grammars defined by R. W. Floyd [FL1]. A grammar G is said to be BC (m, n) if in any $\varphi \in SF(G)$ containing an occurrence of a phrase x_i it can be decided whether or not that x_i must be an A_i-derivative by examining only the m characters to the left of x_i in φ and the n characters to the right of x_i in φ .

In order to insure that every phrase of every sentential form will have m characters occurring to the left and n characters occurring to the right we modify the definition slightly and speak of m, n - sentential forms which are the set of strings,

$$\{ \vdash^m \varphi \dashv^n \mid \varphi \in SF(G) \}$$

where $\vdash$ and $\dashv$ are two new characters not in the vocabulary of G and are called the left and right end markers. There are numerous ways of dealing with this formally; for example, given a grammar $G = (V, P, V_T, S)$ and specific values for m and n , say 2 and 3 respectively, we select a new symbol, S_0 , not originally in V , add it to V_N, add $\vdash$ and $\dashv$ to V_T, add a 0th production, $S_0 \rightarrow \vdash \vdash S \dashv \dashv \dashv$, to P , and make S_0 the new sentence prototype of G . Now all the sentential forms of G are m, n - sentential forms (with the exception of the string, $\vdash \vdash S \dashv \dashv \dashv$) and any occurrence of a phrase x_i ($1 \leq i \leq o(P)$) in a sentential form of G always has at least two characters to the left and three characters to the right of it. We do not wish to introduce a cumbersome notation for this type of trivial modification to a grammar; therefore, a grammar that has been so modified will simply be said to include end markers.

Floyd shows that for any given values for m and n , say m_0 and n_0 , the set $BC(m_0, n_0)$ satisfies properties 1 and 2 . Given any grammar with end markers G, the following is a decision procedure to determine if $G \in BC(m_0, n_0)$.

For every production $A_i \rightarrow x_i$ of P and for every pair of strings (w, y) such that $w, y \in V^*$, $\lambda(w) = m_0$, and $\lambda(y) = n_0$, check to see if

*) $$S \xrightarrow{*} \cdots wA_i y \cdots \quad .$$

That *) is decidable has been shown by Bar-Hillel, Perles, and Shamir [BPS]. If *) is false then the phrase x_i occurring in the context, $\cdots wx_i y \cdots$, can never be a handle of a sentential form of G. If *) is true, then if G is to be $BC(m_0, n_0)$, every occurrence of the phrase, x_i, in the context, $\cdots wx_i y \cdots$, must be a handle and in particular an A_i-derivative. We express this formally as,

**) if φ is of the form $vwx_i yz$ and $\delta \in \overline{m}^{-1}(\varphi)$, then δ is of the form $\psi_1\psi_2[x_i]_i\omega_1\omega_2$ where $m(\psi_1) = v$, $m(\psi_2) = w$, $m(\omega_1) = y$, and $m(\omega_2) = z$.

That **) is decidable for given w, y, and i, Floyd shows by the following analysis by cases.

If **) is false, there is a sentential form φ' in which the phrase x_i occurs in context (w, y) but x_i is not an immediate A_i-derivative; i.e. the characters, x_i, have a bracket structure in $\delta' \in \overline{m}^{-1}(\varphi')$ other than $[x_i]_i$. Consider the innermost pair of brackets in δ' that contain at least one character of x_i; if there is more than one such bracket pair, choose the leftmost. Now let s be the substring of characters of φ' that are bracketed in δ' by the pair thus chosen, and let B denote the element of V_N associated with the right bracket; i.e. if the right bracket chosen above is $]_k$, B is the character, A_k, in the kth production $A_k \rightarrow x_k$. The string s will be of one of the following sixteen forms depending on the position of its left and right ends within the string $\varphi' = \ldots wx_iy\ldots$:

s_{11}: $\ldots wx_{i1}$	s_{12}: $\ldots wx_i$	s_{13}: $\ldots wx_iy_1$	s_{14}: $\ldots wx_iy\ldots$
s_{21}: w_2x_{i1}	s_{22}: w_2x_i	s_{23}: $w_2x_iy_1$	s_{24}: $w_2x_iy\ldots$
s_{31}: x_{i1}	s_{32}: x_i	s_{33}: x_iy_1	s_{34}: $x_iy\ldots$
s_{41}: x_{i4}	s_{42}: x_{i2}	s_{43}: $x_{i2}y_1$	s_{44}: $x_{i2}y\ldots$

where,

$$w_1, w_2, y_1, y_2, x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5} \in V^+ ,$$

$$w_1w_2 = w ,$$

$$y_1y_2 = y ,$$

and $x_{i1}x_{i2} = x_i = x_{i3}x_{i4}x_{i5}$.

Thus if **) is false, one of the following sets of relations must be true for appropriate values of the string variables v_1 and v_2 .

R_{11}: $S \rightarrow^{*} \ldots Bx_{i2}y\ldots, \quad B \rightarrow v_1x_{i1}, \quad v_1 \rightarrow^{*} \ldots w$

R_{12}: $S \rightarrow^{*} \ldots By\ldots, \quad B \rightarrow v_1x_i, \quad v_1 \rightarrow^{*} \ldots w$

R_{13}: $S \rightarrow^{*} \ldots By_2\ldots, \quad B \rightarrow v_1x_iv_2, \quad v_1 \rightarrow^{*} \ldots w, \quad v_2 \rightarrow^{*} y_1$

R_{14}: $S \rightarrow^{*} \ldots B\ldots, \quad B \rightarrow v_1x_iv_2, \quad v_1 \rightarrow^{*} \ldots w, \quad v_2 \rightarrow^{*} y\ldots$

R_{21}: $S \rightarrow^{*} \ldots w_1Bx_{i2}y\ldots, \quad B \rightarrow v_1x_{i1}, \quad v_1 \rightarrow^{*} w_2$

R_{22}: $S \rightarrow^{*} \ldots w_1By\ldots, \quad B \rightarrow v_1x_i, \quad v_1 \rightarrow^{*} w_2$

R_{23}: $S \rightarrow^{*} \ldots w_1By_2\ldots, \quad B \rightarrow v_1x_iv_2, \quad v_1 \rightarrow^{*} w_2, \quad v_2 \rightarrow^{*} y_1$

R_{24}: $S \rightarrow^{*} \ldots w_1B\ldots, \quad B \rightarrow v_1x_iv_2, \quad v_1 \rightarrow^{*} w_2, \quad v_2 \rightarrow^{*} y\ldots$

R_{31}: $S \rightarrow^{*} \ldots wBx_{i2}y\ldots, \quad B \rightarrow x_{i1}$

R_{32}: $S \rightarrow^{*} \ldots wBy\ldots, \quad B \rightarrow x_i$, and $A_i \neq B$

R_{33}: $S \rightarrow^{*} \ldots wBy_2\ldots, \quad B \rightarrow x_iv_1, \quad v_1 \rightarrow^{*} y_1$

R_{34}: $S \rightarrow^{*} \ldots wB\ldots, \quad B \rightarrow x_iv_1, \quad v_1 \rightarrow^{*} y\ldots$

R_{41}: $S \rightarrow^{*} \ldots wx_{i3}Bx_{i5}y, \quad B \rightarrow x_{i4}$

R_{42}: $S \rightarrow^{*} \ldots wx_{i1}By\ldots, \quad B \rightarrow x_{i2}$

R_{43}: $S \rightarrow^{*} \ldots wx_{i1}By_2, \quad B \rightarrow x_{i2}v_1, \quad v_1 \rightarrow^{*} y_1$

R_{44}: $S \rightarrow^{*} \ldots wx_{i1}B\ldots, \quad B \rightarrow x_{i2}v_1, \quad v_1 \rightarrow^{*} y\ldots$

Observe that the method for choosing the string s did not require that s be an immediate B-derivative but only that the portion of x_i contained in s be immediately derived from B. It is this consideration that makes necessary the string variables such as v_1 in relation F_{12} for example.

For fixed w, y, and i, it can be decided if any one of these sixteen sets of relations is true by applying the decision procedures for the general questions, $\varphi \rightarrow^* \ldots\psi\ldots$, $\varphi \rightarrow^* \psi\ldots$, and $\varphi \rightarrow^* \ldots\psi$, presented in Bar-Hillel, Perles, and Shamir [BPS]. Therefore, given values m_0 and n_0 the set $BC(m_0, n_0)$ satisfies property 1. We will prove formally in the next section that the set $BC(m_0, n_0)$ satisfies property 2 as well.

In the same paper Floyd defines another class of subsets of grammars which he calls Bounded Right Context (m, n) grammars. If in the above discussion we can assume that no handle lies to the left of the rightmost character of x_i in the sentential form $\varphi' = \ldots wx_iy\ldots$, then some of the above sixteen sets of relations can never be true and others can be simplified as follows:

Relations R_{11}, R_{21}, R_{31}, and R_{41} cannot occur since they all assume the existence of a handle whose right end falls properly within the phrase x_i and therefore violates the above condition that no handle is to the left of the right end of x_i . The following relations are simplified because now the characters of w that are contained in s must be immediately derived from B as well as the characters of x_i that are contained in s , since there can be no handles to the left of x_i in φ' .

R'_{12}: $S \rightarrow^{*} \ldots By \ldots , \quad B \rightarrow \ldots wx_i$

R'_{13}: $S \rightarrow^{*} \ldots By_2 \ldots , \quad B \rightarrow \ldots wx_iv_1 , \quad v_1 \rightarrow^{*} y_1$

R'_{14}: $S \rightarrow^{*} \ldots B \ldots , \quad B \rightarrow \ldots wx_iv_1 , \quad v_1 \rightarrow^{*} y \ldots$

R'_{22}: $S \rightarrow^{*} \ldots w_1By \ldots , \quad B \rightarrow w_2x_i$

R'_{23}: $S \rightarrow^{*} \ldots w_1By_2 , \quad B \rightarrow w_2x_iv_1 , \quad v_1 \rightarrow^{*} y_1$

R'_{24}: $S \rightarrow^{*} \ldots w_1B \ldots , \quad B \rightarrow w_2x_iv_1 , \quad v_1 \rightarrow^{*} y \ldots$

If for each i and for every pair (w, y) such that $\lambda(w) = m_0$, $\lambda(y) = n_0$, and $S \rightarrow^{*} \ldots wA_iy \ldots$, it is the case that no one of the relations R'_{12}, R'_{13}, R'_{14}, R'_{22}, R'_{23}, R'_{24}, R_{32}, R_{33}, R_{34}, R_{42}, R_{43}, or R_{44} is true, then the grammar is Bounded Right Context (m_0, n_0). A parser that operates from left to right on its input string, thereby reducing all leftmost handles first, justifies the above assumption. Floyd defines the class of Bounded Left Context (m, n) grammars in a similar fashion by removing relations R_{41}, R_{42}, R_{43}, and R_{44}, simplifying the relations

R_{13}, R_{23}, R_{33}, R_{14}, R_{24}, and R_{34}, and requiring the parser to operate from right to left. The above definitions do not exploit the full power afforded by a left to right parsing method as we shall exhibit in section 3.

One final very important class of subsets of grammars satisfying properties 1 and 2 is the class of grammars translatable from left to right with bound k , (LR(k)), as defined by D. E. Knuth [KNU]. Briefly, a grammar is said to be LR(k) if the leftmost handle of any sentential form of G is uniquely determined by the string to its left and the k characters to its right. The powerful feature of the definition of LR(k) grammars is that in parsing from left to right in a sentential form we are permitted to look arbitrarily far to the left to determine if a particular phrase is the leftmost handle. The remarkable thing about the LR(k) property is that Knuth is able to construct a parser that is a deterministic pushdown automaton [GIN]. Given a G and a k_0, Knuth has an effective procedure that will produce a real time parser for G if G is LR(k_0) and will report failure otherwise. Therefore the class LR(k) satisfies properties 1 and 2. In the same paper Knuth shows that while the problem to determine of an arbitrary grammar G whether G is LR(k) is solvable for given k , the problem to determine of an arbitrary grammar G whether thereexists an integer k such that G is LR(k) is recursively unsolvable. Similarly he shows that the problem to determine of an arbitrary grammar G whether there exist integers m and n such that G is $BC(m,n)$ is recursively unsolvable. We will use this latter result in section 4.

SECTION 2. BOUNDED CONTEXT PARSABLE GRAMMARS

In this section we define the class of Bounded Context Parsable grammars and give an effective procedure for producing real time parsers for them.

Given a grammar $G = (V, P, V_T, S)$, a pair of strings (w, y) will be called a derivation context for the production $A_i \rightarrow x_i$ if:

i) $w, y \in V^*$, and

ii) $S \xrightarrow{*} \ldots wA_iy\ldots$.

A pair of strings (w, y) will be called a parsing context for the production $A_i \rightarrow x_i$ if:

i) (w, y) is a derivation context for $A_i \rightarrow x_i$, and

ii) if $\varphi \in SF(G)$ is of the form $\varphi = vwx_iyz$ for some $v, z \in V^*$ and $\delta \in \overline{m}^{-1}(\varphi)$, then δ is of the form $\psi_1\psi_2[x_i]_i\omega_1\omega_2$ where $m(\psi_1) = v$, $m(\psi_2) = w$, $m(\omega_1) = y$, and $m(\omega_2) = z$.

In other words, (w, y) is a parsing context for the production $A_i \rightarrow x_i$ if x_i is always an immediate A_i-derivative whenever it appears in a sentential form φ in the context (w, y); <u>i.e.</u> $\varphi = \ldots wx_iy\ldots$. Observe that if (w, y) is a parsing context for a production and (vw, yz) is a derivation context for that production, then (vw, yz)

must also be a parsing context for that production.

A parsing context (w, y) will be said to be of order $[m, n]$ if $\lambda(w) = m$ and $\lambda(y) = n$. A parsing context of order $[m_1, n_1]$ will be said to have order less than a parsing context of order $[m_2, n_2]$ if $m_1 \leq m_2$ and $n_1 \leq n_2$. The set of derivation contexts for the i<u>th</u> production of order $[m, n]$ or less will be denoted by $DC_i[m, n]$. The set of parsing contexts for the i<u>th</u> production of order $[m, n]$ or less will be denoted by $PC_i[m, n]$ and its j<u>th</u> member will be denoted by (w_{ij}, y_{ij}).

A particular occurrence of a phrase x_i in a sentential form φ will be said to occur in a parsing context of order $[m, n]$ in φ if the pair of strings (ℓ, r) is in $PC_i[m, n]$ when ℓ is the string consisting of the m characters immediately to the left of that particular occurrence of x_i in φ and r is the string consisting of the n characters immediately to the right of that particular occurrence of x_i in φ.

A grammar G will be said to be Bounded Context Parsable with left bound m and right bound n, ($BCP[m, n]$), if every sentential form of G contains at least one phrase occurring in a parsing context of order $[m, n]$ or less.

To illustrate these definitions consider the grammar,

$$G_1 = (\{S, A, B, E, a, b, e, \vdash, \dashv\}, P_1, \{a, b, e, \vdash, \dashv\}, S) \text{ with } P_1 :$$

$S \rightarrow \vdash AEa \dashv$

$S \rightarrow \vdash BEb \dashv$

$E \rightarrow eE$

$E \rightarrow e$

$A \rightarrow e$

$B \rightarrow e$

For each of the six productions we list the set of derivation contexts of order [2, 2] or less and the set of parsing contexts of order [2, 2] or less. As a notational convenience, $(\{s_1, s_2, s_3\}, \{s_4, s_5\})$ will be used to stand for the 6 contexts, (s_1, s_4), (s_2, s_4), (s_3, s_4), (s_1, s_5), (s_2, s_5), and (s_3, s_5).

1) $S \rightarrow \vdash AEa \dashv$

$DC_1[2, 2] : (\epsilon, \epsilon)$

$PC_1[2, 2] : (\epsilon, \epsilon)$

2) $S \rightarrow \vdash BEb \dashv$

$DC_2[2, 2] : (\epsilon, \epsilon)$

$PC_2\ [2, 2]: (\epsilon, \epsilon)$

3) $E \rightarrow eE$

$DC_3[2, 2] : (\{\epsilon, e, ee, \vdash e\}, \{\epsilon, a, b, a\dashv, b\dashv\})$

$(\{A, \vdash A, Ae\}, \{\epsilon, a, a\dashv\})$

$(\{B, \vdash B, Be\}, \{\epsilon, b, b\dashv\})$

$PC_3[2,2] : (\ \{e, ee, \vdash e\},\ \ \{\epsilon, a, b, a\dashv, b\dashv\}\ )$

$(\ \{A, \vdash A, Ae\},\ \ \{\epsilon, a, a\dashv\}\ )$

$(\ \{B, \vdash B, Be\},\ \ \{\epsilon, b, b\dashv\}\ )$

4) $E \rightarrow e$

$DC_4[2,2]$: (the same as $DC_3[2,2]$ by the definition of derivation context)

$PC_4[2,2] : (\ \{\epsilon, e, \vdash e, ee\},\ \ \{a, b, a\dashv, b\dashv\}\ )$

$(\ \{A, \vdash A, Ae\},\ \ \{a, a\dashv\}\ )$

$(\ \{B, \vdash B, Be\},\ \ \{b, b\dashv\}\ )$

5) $A \rightarrow e$

$DC_5[2,2] : (\ \{\epsilon,\ \vdash\},\ \ \{\epsilon, E, e, Ea, eE, ea, ee\}\ )$

$PC_5[2,2] : (\vdash, Ea),\ \ (\vdash, ea)$

6) $B \rightarrow e$

$DC_6[2,2] : (\ \{\epsilon,\ \vdash\},\ \ \{\epsilon, E, e, Eb, eE, eb, ee\}\ )$

$PC_6[2,2] : (\vdash, Eb),\ \ (\vdash, eb)$

Notice that:

i) (ϵ, ϵ) is a parsing context for production 1 ; that is, whenever $\vdash AEa\dashv$ occurs it can be replaced by S "regardless of its context."

ii) (ϵ, ϵ) is a derivation context for production 3, i.e. $S \xrightarrow{*} \ldots E \ldots$, but (ϵ, ϵ) is not a parsing context for 3; e.g. in the sentential form, $\vdash \underline{eE}a \dashv$, the underlined occurrence of eE is not an E-derivative.

iii) (e, ϵ) is a parsing context for 3; i.e. in any sentential form, $\ldots e\underline{eE} \ldots$, the underlined occurrence of eE is an immediate E-derivative.

iv) $(\vdash, eE)$ is a derivation context for both productions 5 and 6, but it is not a parsing context for either; i.e. while $\vdash BeE \cdots$ and $\vdash AeE \cdots$ may occur in a sentential form, the underlined occurrence of e in $\vdash \underline{e}eE \cdots$ may be either an immediate A-derivative or an immediate B-derivative and the context $(\vdash, eE)$ is not sufficient to dictate which. Note that e is also the phrase of production 4, but $(\vdash, eE)$ is not even a derivation context for production 4 since $\vdash EeE$ can never occur in a sentential form.

Now any sentential form of G_1 must be of one of the following twelve forms:

1. $\vdash AEa \dashv$
2. $\vdash Ae^nEa \dashv$, $n \geq 1$
3. $\vdash Ae^na \dashv$, $n \geq 1$
4. $\vdash eEa \dashv$
5. $\vdash e^nEa \dashv$, $n \geq 2$
6. $\vdash e^na \dashv$, $n \geq 2$
7. $\vdash BEb \dashv$
8. $\vdash Be^nEb \dashv$, $n \geq 1$
9. $\vdash Be^nb \dashv$, $n \geq 1$
10. $\vdash eEb \dashv$
11. $\vdash e^nEb \dashv$, $n \geq 2$
12. $\vdash e^nb \dashv$, $n \geq 2$

That each of these forms contains at least one phrase occurring in a parsing context of order $[2, 2]$ or less may be seen by analyzing the possible cases:

1.) Form 1 contains the phrase, $\vdash AEa \dashv$ occurring in context (ϵ, ϵ) which is a parsing context for production 1 , $S \rightarrow \vdash AEa \dashv$.

2.) Form 2 contains an occurrence of eE in context $(\vdash A, a)$ if $n = 1$ or (e, a) if $n > 1$ and both of these are parsing contexts for production 3 , $E \rightarrow eE$.

3.) Forms 3 and 6 contain an occurrence of e in context (ϵ, a) which is a parsing context for production 4, $E \rightarrow e$.

4.) Form 4 contains an occurrence of e in the context $(\vdash, Ea)$ which is a parsing context for production 5, $A \rightarrow e$.

5.) Form 5 contains an occurrence of eE in the context (e, ϵ) which is a parsing context for production 3, $E \rightarrow eE$.

6.) Similarly for forms 7-12.

Therefore the grammar G_1 is BCP[2, 2] since all of the parsing contexts used in the above analysis are of order [2, 2] or less. If at case 2) in the analysis we use the parsing context (A, a) instead of $(\vdash A, a)$ for $n = 1$, then it is seen that G_1 is BCP[1, 2] since all contexts used in the analysis would then be of order [1, 2] or less. It is natural to ask if there is some analysis that will show G_1 to be BCP[1, 1]. Since there is no parsing context for the production $A \rightarrow e$ of order less than [1, 2], if there is any sentential form of G_1 in which the only handle is the phrase e occurring as an immediate A-derivative, G_1 cannot be BCP[1, 1]; $\vdash eEa \dashv$ is such a sentential form.

If a grammar is BCP[m, n], then since every sentential form has a handle occurring in parsing context, the grammar is clearly unambiguous by induction on the number of steps in the derivation of a sentential form.

In the remainder of this section we will show that the class of Bounded Context Parsable grammars satisfies Property 2. Let $G = (V, P, V_T, S)$ be a BCP$[m, n]$ grammar, $p = o(P)$, $PC_i[m, n]$ be the set of k_i parsing contexts for the $i\underline{th}$ production of P $(1 \leq i \leq p)$, and $\varphi \in L(G)$. We can construct the parse of φ by the following procedure:

i) set $\psi_1 = \vdash \varphi \dashv$ and $j = 1$

ii) search ψ_j for a phrase x_i occurring in parsing context and consider the string formed by replacing that occurrence of x_i by A_i; set ψ_{j+1} equal to the string thus formed.

iii) Set $j = j + 1$

iv) If $\psi_j = S$, stop; otherwise, go back to step ii).

Step ii) will always be possible since every sentential form, and therefore every ψ_j, will contain a phrase occurring in a parsing context of order $[m, n]$ or less and for each of the productions in P there are only finitely many such parsing contexts for which we have to look.

We can describe this procedure as a kind of Floyd reduction system [FL2] containing two classes of reduction rules, I and II, where the reduction process always attempts to reduce by a rule of type I before attempting to reduce by a rule of type II. Class I consists of the rules:

$$\begin{array}{rcl}
w_{11}x_1y_{11}\Delta & \rightarrow & w_{11}A_1y_{11}\Delta \\
w_{12}x_1y_{12}\Delta & \rightarrow & w_{12}A_1y_{12}\Delta \\
 & \vdots & \\
w_{1k_1}x_1y_{1k_1}\Delta & \rightarrow & w_{1k_1}A_1y_{1k_1}\Delta \\
w_{21}x_2y_{21}\Delta & \rightarrow & w_{21}A_2y_{21}\Delta \\
w_{22}x_2y_{22}\Delta & \rightarrow & w_{22}A_2y_{22}\Delta \\
 & \vdots & \\
w_{2k_2}x_2y_{2k_2}\Delta & \rightarrow & w_{2k_2}A_2y_{2k_2}\Delta \\
 & \vdots & \\
w_{p1}x_py_{p1}\Delta & \rightarrow & w_{p1}A_py_{p1}\Delta \\
 & \vdots & \\
w_{pk_p}x_py_{pk_p}\Delta & \rightarrow & w_{pk_p}A_py_{pk_p}\Delta
\end{array}$$

I.e., class I consists of the $\sum_{i=1}^{p} k_i$ rules,

$$w_{ij}x_iy_{ij}\Delta \quad \rightarrow \quad w_{ij}A_iy_{ij}\Delta\,, \quad 1 \le j \le k_i, \quad 1 \le i \le p.$$

Class II consists of the $o(V)$ rules,

$$\Delta v_i \rightarrow v_i\Delta\,, \quad \text{for all} \quad v_i \in V\,.$$

The reduction process operates as described by Floyd for the rules of class II. However, after application of a class I rule, the scanning marker Δ is moved back to the beginning of the string before the next scan for a reduction rule. This is because step ii) in the procedure searches each string from the beginning and not from the point of the last reduction. Using this reduction system a large amount of time may be wasted in scanning each sentential form for a phrase occurring in parsing context. For example, with the grammar $(\{S, a\}, \{S \rightarrow aS, S \rightarrow a\}, \{a\}, S)$ since the only handle in any sentential form is at the right hand end, the entire string must be scanned each time and the number of reduction rules applied will be about $\frac{\lambda(\varphi)^2}{2}$.

The parsing method can be improved by observing that it is not necessary to start scanning at the beginning of ψ_j in step ii) for $j > 1$. If the A_i introduced in the application of step ii) to ψ_{j-1} is the qth character in ψ_j, then no phrase in ψ_j whose right-most character is to the left of the $(q - n)$th character of ψ_j can occur in a parsing context since it would have been discovered and reduced by an earlier application of step ii). Therefore we may resume scanning n characters to the left of the most recently introduced symbol in ψ_j rather than going back to the beginning. We can implement this modification in the reduction system by removing the special treatment of type I reductions, i.e. by not moving the marker Δ back to the beginning of the string, and by rewriting each type I rule in the form:

$$(\mathrm{I}') \qquad w_{ij}x_iy_{ij}\Delta \quad \rightarrow \quad w_{ij}A_i\Delta y_{ij} \, , \quad 1 \leq j \leq k_i, \quad 1 \leq i \leq p.$$

With this improvement the reduction system will parse in a time linearly proportional to the length of the sentence φ since the number of reductions applied with the modified reduction system is bounded above by a linear function of φ as is demonstrated in the following:

Theorem: If $G = (V, P, V_T, S)$ is BCP[m, n], there exists an integer K such that the number of reductions used by the modified reduction system described above in parsing any $\varphi \in L(G)$ is $K \cdot \lambda(\varphi)$ or less.

Proof:

1) Let $p = o(P)$.

2) Since G is BCP, it is unambiguous and no infinite cycling of 1-productions can occur in the derivation of a sentence of length ℓ. Therefore the number of applications of productions in P used in deriving a sentence of length ℓ can be at most $2p \cdot \ell$ since at least one terminal character or one additional nonterminal character must be produced after the application of every 2p rewriting rules.

3) Therefore from 2), the number of applications of reduction rules of type I' can be at most $2p\ell$ for a sentence of length ℓ.

4) The number of applications of type II rules can be at most ℓ (to scan the entire sentence) plus the number of characters that must be rescanned due to moving the pointer back in applications of type I' rules. Since the pointer backs up at most n characters at each application of a type I' rule and the number of such applications is at most $2p\ell$ by 2) above, the number of applications of reductions of type II can be at most $\ell + 2p\ell n$.

5) Therefore the total number of rules applied in reducing a sentence of length ℓ will be at most $2p\ell + \ell + 2p\ell n$.

6) Thus $K = 2p(n+1) + 1$, and since p and m are constants of the grammar independent of ℓ, the theorem is proved.

Q.E.D.

To illustrate the parsing method we again consider the grammar G_1 defined earlier in this section. We have shown that G_1 is BCP[2, 2]. We will construct a reduction system for G_1 and use it to parse the sentence, $\vdash eeeeea \dashv$. Whereas each of the productions may have a large number of parsing contexts of order [2, 2] or less, it often will not be necessary to include them all in the reduction system constructed for parsing; indeed, if (w, y) and (vw, yz) are both parsing contexts for production i, it will not be necessary to include the reduction rule $vwx_i yz\Delta \rightarrow vwA_i\Delta yz$ since the phrases reduced by the rule $wx_i y\Delta \rightarrow wA_i\Delta y$ will include all those of the former. For example in G_1 the production $E \rightarrow eE$ has thirty-three parsing contexts of order [2, 2] or less but only three, (e, ϵ), (A, ϵ), and (B, ϵ), need to be included in the reduction system. While the number of rules in the system R_1 has been greatly reduced by eliminating unnecessary parsing contexts, there are still some redundant rules in the system. We see that rules 17, 18, and 20 are useless since A, B, and S are nonterminals and are never introduced to the right of Δ in any reduction. Therefore these rules can never be applicable to any sentential form. A more subtle redundancy occurs in rules 9 and 11. Rule 9 is in the system because $(\vdash, ea)$ is a parsing context for the production $A \rightarrow e$. However, any sentential form containing the phrase e in that context must also contain an occurence of e in the context (ϵ, a) which is a parsing context for $E \rightarrow e$. If the latter reduction is made first, i.e. rule 6 is applied first, then rule 9 will never be applicable, since the phrase e in question will now occur in the context $(\vdash, Ea)$ and be reduced by rule 8. Similarly rules 7 and 10 make rule 11 redundant.

The following twenty rule reduction system, R_1 , will parse sentences in $L(G_1)$:

1 $\vdash A\ E\ a \dashv \Delta \rightarrow S\ \Delta$

2 $\vdash B\ E\ b \dashv \Delta \rightarrow S\ \Delta$

3 $e\ e\ E\ \Delta \rightarrow e\ E\ \Delta$

4 $A\ e\ E\ \Delta \rightarrow A\ E\ \Delta$

5 $B\ e\ E\ \Delta \rightarrow B\ E\ \Delta$

6 $e\ a\ \Delta \rightarrow E\ \Delta\ a$

7 $e\ b\ \Delta \rightarrow E\ \Delta\ b$

8 $\vdash e\ E\ a\ \Delta \rightarrow \vdash A\ \Delta\ E\ a$

9 $\vdash e\ e\ a\ \Delta \rightarrow \vdash A\ \Delta\ e\ a$

10 $\vdash e\ E\ b\ \Delta \rightarrow \vdash B\ \Delta\ E\ b$

11 $\vdash e\ e\ b\ \Delta \rightarrow \vdash B\ \Delta\ e\ b$

12 $\Delta\ a \rightarrow a\ \Delta$

13 $\Delta\ b \rightarrow b\ \Delta$

14 $\Delta\ e \rightarrow e\ \Delta$

15 $\Delta\ \vdash \rightarrow \vdash \Delta$

16 $\Delta\ \dashv \rightarrow \dashv\ \Delta$

17 $\Delta\ A \rightarrow A\ \Delta$

18 $\Delta\ B \rightarrow B\ \Delta$

19 $\Delta\ E \rightarrow E\ \Delta$

20 $\Delta\ S \rightarrow S\ \Delta$

Using the reduction system R_1 the sentence, $\vdash eeeeea \dashv$, would be parsed as follows:

Sentential form	First rule applicable
$\Delta \vdash e\ e\ e\ e\ e\ a \dashv$	15
$\vdash \Delta\ e\ e\ e\ e\ e\ a \dashv$	14
$\vdash e\ \Delta\ e\ e\ e\ e\ a \dashv$	14
$\vdash e\ e\ \Delta\ e\ e\ e\ a \dashv$	14
$\vdash e\ e\ e\ \Delta\ e\ e\ a \dashv$	14
$\vdash e\ e\ e\ e\ \Delta\ e\ a \dashv$	14
$\vdash e\ e\ e\ e\ e\ \Delta\ a \dashv$	12
$\vdash e\ e\ e\ e\ e\ a\ \Delta \dashv$	6
$\vdash e\ e\ e\ e\ E\ \Delta\ a \dashv$	3
$\vdash e\ e\ e\ E\ \Delta\ a \dashv$	3
$\vdash e\ e\ E\ \Delta\ a \dashv$	3
$\vdash e\ E\ \Delta\ a \dashv$	12
$\vdash e\ E\ a\ \Delta \dashv$	8
$\vdash A\ \Delta\ E\ a \dashv$	19
$\vdash A\ E\ \Delta\ a \dashv$	12
$\vdash A\ E\ a\ \Delta \dashv$	16
$\vdash A\ E\ a \dashv \Delta$	1
$S\ \Delta$	done

SECTION 3. RELATIONSHIP OF BOUNDED CONTEXT PARSABLE GRAMMARS WITH OTHER SUBSETS OF GRAMMARS

In this section we will compare the set of BCP[m, n] grammars with the sets of grammars that are BC(m, n), BRC(m, n), BLC(m, n), LR(n), or RL(m). We will also show that the set of languages that are BCP[m, n] properly contains the set of deterministic languages defined by Ginsburg and Greibach [GIN].

We will let $S_{BCP[m,n]}$ denote the set of BCP[m, n] grammars. The definition of Bounded Context Parsable induces the partial ordering on the sets $S_{BCP[n,m]}$ defined by:

$$S_{BCP[m_1,n_1]} \supset S_{BCP[m_2,n_2]} \text{ if } m_1 \geq m_2 \text{ and } n_1 \geq n_2 .$$

We will let S_{BCP} denote the set, $\bigcup_{m,n \geq 0} S_{BCP[m,n]}$; *i.e.* a grammar G is in S_{BCP} if $\exists m \; \exists n$ such that G is BCP[m, n]. It is interesting to compare the sets of Bounded Context Parsable grammars with other sets of grammars that satisfý properties 1 and 2 as given in Section 1 above. We present the results of these comparisons as a Venn diagram in figure 1 on page 37.

$$S_{BCP} \underset{\neq}{\supset} S_{BC} .$$

If a grammar is Bounded Context (m, n), then every derivation context of order [m, n] must be a parsing context by the definition of Bounded

Context grammars. Since every sentential form has some phrase occurring as a handle and therefore occurring in a derivation context of order $[m, n]$, every sentential form has a phrase occurring in a parsing context of order $[m, n]$ or less; therefore the grammar must be $BCP[m, n]$. That the inclusion is proper is demonstrated by the existence of G_1 which we have shown is $BCP[1, 2]$. G_1 is not $BC(1, 2)$ nor is it $BC(m, n)$ for any m and n. This is seen in that $(\vdash^m, e^n)$ is a derivation context for the production $A \rightarrow e$, i.e. $S \overset{*}{\rightarrow} \cdots \vdash^m A e^n \cdots$ (recall our implicit assumption about having the requisite number of end markers present to allow all phrases to occur in a context of order $[m, n]$.) But the context $(\vdash^m, e^n)$ is not a parsing context for $A \rightarrow e$ since there are sentential forms in which the occurrence of e in the context $(\vdash^m, e^n)$ is not an A-derivative, e.g. $\vdash^m e\, e^n b \dashv$. Therefore G_1 is not in $S_{BC(m, n)}$ for any m or n; i.e. $G_1 \not\in S_{BC}$.

Whereas we have used $S_{BC(m, n)}$ to denote the set of Bounded Context (m, n) grammars exactly as defined by Floyd, we will modify his definitions of bounded right context and bounded left context grammars somewhat. As we mentioned in Section 1, his definitions do not fully exploit the power of left to right or right to left syntactic analysis. To demonstrate this, consider G^R_1 the reversal of the grammar G_1. G^R_1 has productions:

$$S \rightarrow \vdash a\ E\ A \dashv$$

$$S \rightarrow \vdash b\ E\ B \dashv$$

$$E \rightarrow E\ e$$

$$E \rightarrow e$$

$$A \rightarrow e$$

$$B \rightarrow e$$

Now by Floyd's definition, G^R_1 is not Bounded Right Context (2, 1); this is because $S \rightarrow^* \cdots e\ e\ A \dashv$, but the set of relations R_{32} is true, i.e. $S \rightarrow^* \cdots e\ e\ B \dashv$, $B \rightarrow e$, and $B \neq A$. That is, one of the derivation contexts for $A \rightarrow e$, $(ee, \dashv)$, is not a parsing context even under the modified tests for parsing context. But a parser operating from left to right would never need to consider the context $(e\ e, \dashv)$ for the phrase e . By the time it reached an e occurring next to the right end marker, it would be working on either the sentential form $\vdash a\ E\ e \dashv$ or the sentential form $\vdash b\ E\ e \dashv$, and both of these derivation contexts for the phrase e , $(a\ E, \dashv)$ and $(b\ E, \dashv)$, are parsing contexts, the former for the production $A \rightarrow e$ and the latter for the production $B \rightarrow e$. Therefore, for a parser operating from left to right, the only derivation contexts that need to be parsing contexts are those that occur leftmost in a sentenial form. That is, only those derivation contexts for a production whose phrase occurs as the leftmost handle of a sentential form need to be parsing contexts for that production.

We define the set of left-restricted derivation contexts of a production as follows:

The pair (w, y) will be said to be a left-restricted derivation context of order $[m, n]$ for the ith production $A_i \rightarrow x_i$, ($LDC_i[m, n]$), iff

1) $\lambda(w) = m$

2) $\lambda(y) = n$

3) $\exists v \exists z$ such that

   i) $v, z \in V^*$

   ii) $S \rightarrow^* vwA_i yz$

   iii) in the string $\delta = \overline{m}^{-1}(vwA_i yz) = \varphi_1 A_i \varphi_2$ where $m(\varphi_1) = vw$ and $m(\varphi_2) = yz$, there are no right brackets in φ_1.

The set of right-restricted derivation contexts is defined similarly. Clearly, $LDC_i[m, n] \subset DC_i[m, n]$ and $RDC_i[m, n] \subset DC_i[m, n]$.

We now define our modified versions of bounded right and left context grammars. A grammar G will be said to be $BRC(m, n)$ if for all productions $A_i \rightarrow x_i$, $(w, y) \in LDC_i[m, n]$ implies that $(w, y) \in PC_i[m, n]$; i.e. G is $BRC(m, n)$ if every left-restricted derivation context for the ith production is a parsing context for the ith production. Similarly, a grammar will be said to be $BLC(m, n)$ if for all productions $A_i \rightarrow x_i$, $(w, y) \in RDC_i[m, n]$ implies $(w, y) \in PC_i[m, n]$.

With this definition, G_1^R is seen to be BRC(2, 1).

Now if a grammar G is BRC(m, n), then since every sentential form has a phrase occurring in a left-restricted derivation context of order [m, n], in particular the leftmost handle of the sentential form, every sentential form has a phrase occurring in a parsing context of order [m, n] and thus G is BCP[m, n]. Thus we have the following inclusion relationship:

$$S_{BCP} \supsetneqq S_{BRC}$$

To see that this inclusion is proper, recall the grammar G_1. G_1 is BCP[1, 2], but it is not BRC(m, n) for any m and n; $(\vdash^m, e^n)$ is a left-restricted derivation context for $A \rightarrow e$, but it is not a parsing context for $A \rightarrow e$ as was shown above. Similarly we have:

$$S_{BCP} \supsetneqq S_{BLC}$$

This inclusion is proper since G_1^R is BCP[2, 1] but not BLC(m, n) for any m and n. G_1^R is BCP[2, 1] since G_1 is BCP[1, 2]. G_1^R is not BLC(n, m) for any m and n since G_1 is not BRC(m, n) for any m and n.

For an example of a grammar that is BCP[2, 2] but is neither BRC(m, n) nor BLC(m, n) for any values of m and n, consider the grammar G_2 with productions:

$S \rightarrow \vdash A\ E\ a\ E\ A \dashv$

$S \rightarrow \vdash B\ E\ b\ E\ B \dashv$

$E \rightarrow e\ E$

$E \rightarrow e$

$A \rightarrow e$

$B \rightarrow e$

When we compare S_{BCP} with the set of grammars that are LR(n) for some n, (S_{LR}), it is seen that neither set contains the other. G_1 is not LR(n) for any value of n since the leftmost handle of the sentential form, $\vdash e\ e^n\ a \dashv$, cannot be parsed by looking only at the characters to the left, $\vdash$, and the n characters to the right, e^n. The unbounded left context available to the LR analysis does no good in this particular situation; the information needed to determine how to parse the first e lies arbitrarily far to the right. As above, G_1 is RL(1), but G_2 is neither LR(n) nor RL(n) for any value of n. Knuth has shown [KNU] that the grammar G_3 with productions:

$S \rightarrow \vdash T \dashv$

$T \rightarrow a\ U\ c$

$T \rightarrow b$

$U \rightarrow a\ T\ c$

$U \rightarrow b$

is LR(0). $L(G_3) = \{a^k b c^k \mid k \geq 0\}$, and the b must be reduced to T or U according as k is even or odd. For any values of m and n, (a^m, c^n) is a derivation context for both of the productions $T \rightarrow b$ and $U \rightarrow b$, but it is a parsing context for neither. Since there is only one handle in any sentential form of G_3, the sentential form $\vdash a^k b c^k \dashv$, where $k = \max(m, n)$, contains no phrase occurring in a parsing context of order $[m, n]$ or less. Therefore G_3 is not BCP[m, n] for any values of m and n. We summarize the results of these comparisons as a Venn diagram in the following figure:

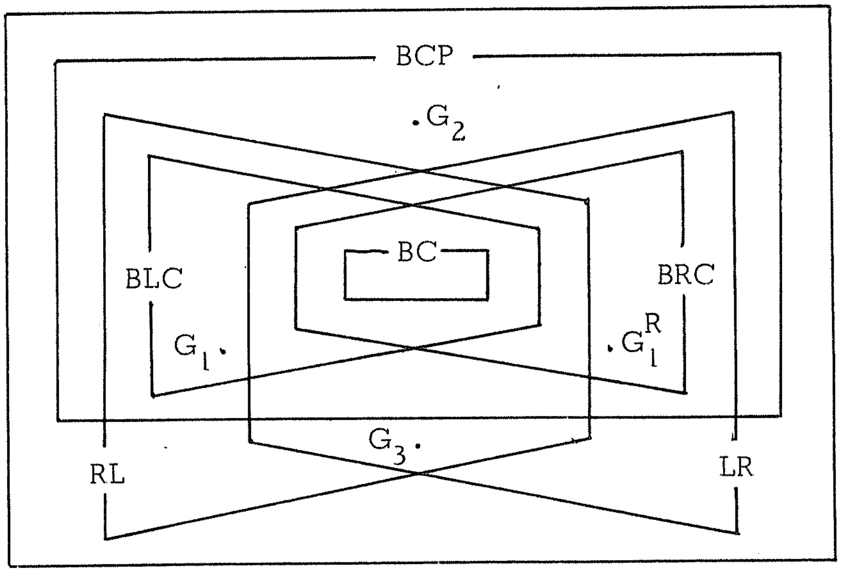

Figure 1. Venn diagram of subsets of context-free grammars.

As is obvious from the definition and was noted above, the Bounded Context Parsable property is symmetric in the following sense. If G is BCP[m, n], then G^R is BCP[n, m] where G^R, the reversal of G, is obtained by reflecting the right hand sides of all the productions of G.

This observation leads to an interesting comparison of subsets of languages.

A language L will be said to be Bounded Context Parsable (BCP) if there exists a context-free grammar G such that G is BCP[m, n] for some values of m and n and L = L(G). Similarly L will be said to be BC, BRC, BLC, LR or RL if there exists a G such that L = L(G) and G is BC(m, n), BRC(m, n), BLC(m, n), LR(n) or RL(m) for some values of m and n. Knuth shows [KNU] that the set of languages that are LR, which we will denote L_{LR}, is exactly the set D of deterministic languages defined by Ginsburg and Greibach [GIN]. Knuth also shows that $L_{BRC} = L_{LR}$ and that the language $\{\vdash a^n b^n \dashv \mid n \geq 1\} \cup \{\vdash a^n b^{2n} c \dashv \mid n \geq 1\}$ is not deterministic and therefore not LR or BRC. We will construct a grammar G_4 for this language and show that it is BCP.

Let G_4 be the grammar with productions:

1) $S \rightarrow \vdash U \dashv$

2) $S \rightarrow \vdash V c \dashv$

3) $U \rightarrow a U B$

4) $U \rightarrow a B$

5) $V \rightarrow a V D D$

6) $V \rightarrow a D D$

7) $B \rightarrow b$

8) $D \rightarrow b$

For each of productions 1) - 6), (ϵ, ϵ) is a parsing context. $(\epsilon, \dashv)$ and (ϵ, B) are parsing contexts for $B \rightarrow b$; (ϵ, c) and (ϵ, D) are parsing contexts for $D \rightarrow b$. Any sentential form of G_4 has either a phrase of one of productions 1) - 6) occurring in it or an occurrence of b with one of the characters, B, D, c, or $\dashv$ on its right. Therefore, every sentential form of G_4 has a phrase occurring in a parsing context of order [0, 1] or less and the grammar is seen to be BCP[0, 1]. Thus we have that:

$$L_{BCP} \supsetneq D .$$

If L is deterministic, the construction of Knuth produces a BRC grammar for L, and that grammar is also BCP [KNU]. That the inclusion is proper is demonstrated by the existence of G_4 .

We will let D^R denote the set of languages whose reversals are deterministic. $L(G_4) \in D^R$ since G^R_4 is LR(0). Using the same approach as above we can construct a grammar G_5 such that G_5 is BCP but $L(G_5) \notin D \cup D^R$. Let G_5 be the grammar with productions:

1) $S \rightarrow \vdash S_1 d S_2 \dashv$

2) $S_1 \rightarrow U_1$	3) $S_2 \rightarrow U_2$
4) $S_1 \rightarrow V_1 c$	5) $S_2 \rightarrow c V_2$
6) $U_1 \rightarrow a U_1 B_1$	7) $U_2 \rightarrow B_2 U_2 a$

8) $U_1 \rightarrow a\ B_1$

9) $U_2 \rightarrow B_2\ a$

10) $V_1 \rightarrow a\ V_1 D_1 D_1$

11) $V_2 \rightarrow D_2 D_2 V_2\ a$

12) $V_1 \rightarrow a\ D_1 D_1$

13) $V_2 \rightarrow D_2 D_2\ a$

14) $B_1 \rightarrow b$

15) $B_2 \rightarrow b$

16) $D_1 \rightarrow b$

17) $D_2 \rightarrow b$

Clearly G_5 is BCP[1, 1], but $L(G_5)$ cannot be the language accepted by a deterministic push-down automaton. We illustrate these comparisons of sets of languages in the following figure:

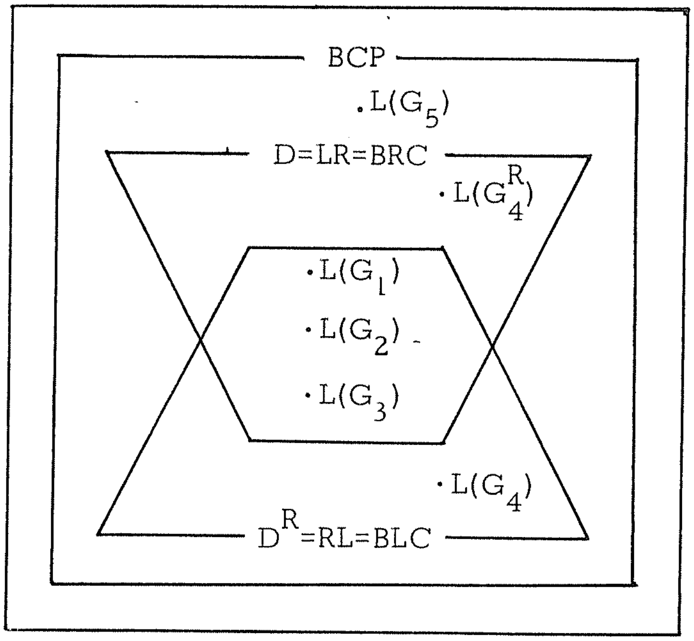

Figure 2. Venn diagram of subsets of context-free languages.

SECTION 4. TESTING FOR BOUNDED CONTEXT PARSABILITY

In this section we will show that the set $S_{BCP[m, n]}$ satisfies our property 1, but that S_{BCP} does not.

So far we have given no general method for determining of an aribtrary grammar G and integers m and n, whether G is $BCP[m, n]$. G_1 was shown to be $BCP[1, 2]$ but not $BCP[1, 1]$ by a special analysis by cases of the different sentential forms possible. It was shown that G_3 fails to be $BCP[m, n]$ for any values of m and n by showing that one particular sentential form of G_3 had no phrase occurring in a parsing context of order $[m, n]$ or less. We demonstrate the existence of a general decision procedure in the following:

Theorem: There is an algorithm to determine of an arbitrary reduced context-free grammar G and arbitrary integers m and n, whether G is $BCP[m, n]$.

Proof: 1) Let $G = (V, P, V_T, S)$ be a reduced context-free grammar, $p = o(P)$, and $m, n \geq 0$.

2) For each production $A_i \rightarrow x_i$ in P, compute $DC_i[m, n]$, the set of derivation contexts of order $[m, n]$ or less for $A_i \rightarrow x_i$. This can be effectively computed by deciding for each of the pairs (w, y) such that $\lambda(w) \leq m$ and $\lambda(y) \leq n$, whether $S \rightarrow^* \ldots wA_iy \ldots$ [BPS].

3) Compute the subsets $PC_i[m, n] \subset DC_i[m, n]$ for each of the p productions in P . That is, for each $(w, y) \in DC_i[m, n]$, determine if (w, y) is a parsing context for the ith production. Floyd's method of analysing the sixteen sets of relations which we described in section 1 is an effective method for determining whether an occurrence of x_i in the context (w, y) is necessarily an A_i-derivative.

4) Let $C_i[m, n] = \{w_{ij}x_iy_{ij} \mid (w_{ij}, y_{ij}) \in PC_i[m, n]\}$ for all $1 \leq i \leq p$.

Let $C[m, n] = \bigcup_{i=1}^{p} C_i[m, n]$

Since $C[m, n]$ is a finite set, it is regular, and therefore the set $R = V^* \cdot C[m, n] \cdot V^*$ is regular where we use $\cdot$ to indicate the complex product as usual. Notice that R is the set of all strings in V^* that contain at least one phrase occurring in a parsing context of order $[m, n]$ or less.

5) Construct G' such that $L(G') = SF(G)$. Let

$G' = (V', P', V'_T, S'')$ where:

$V' = V \cup \{A' \mid A \in V_N\} \cup S''$

$V'_T = V$

P' is the set of productions obtained as follows:

i) If $A \rightarrow x$ is a production of P, then $A' \rightarrow x'$ is a production of P' where x' is the string over V' obtained from x by priming all the non-terminal characters in x.

ii) $A' \rightarrow A$ is in P' for all $A \in V_N$.

iii) If $S \rightarrow x$ is in P, then $S'' \rightarrow x'$ is in P' where again x' is obtained by priming the non-terminals in x. The reason for treating S differently from the other non-terminals of G is that we do not wish to consider $\vdash S \dashv$ to be a sentential form of G (unless, of course, $S \rightarrow^{+} S$ in which case G is ambiguous). That is, we want every sentential form of G to have a handle.

6) Since R is regular, $\overline{R} = V^{*} - R$ is regular [RS].

7) Since $\overline{R}$ is regular and G' is context-free, there exists a context-free grammar $\overline{G}$ such that $L(\overline{G}) = L(G') \cap \overline{R}$. [BPS].

8) Now, G is BCP[m, n]

iff every sentential form of G has a phrase occurring in a parsing context of order [m, n] or less

iff $L(G') \subset R$

iff $L(G') \cap \overline{R}$ is empty

iff $L(\overline{G})$ is empty.

9) There is an effective procedure for determining whether the language of an arbitrary context-free grammar is empty [BPS].

Therefore, we can effectively determine whether G is $BCP[m, n]$.

Q.E.D.

Notice that when the above decision procedure responds affirmatively, we can immediately construct a reduction system for the sets $PC_i[m, n]$ to parse sentences of G as was shown in section 2. Thus for each pair of values for m and n, $S_{BCP[m, n]}$ satisfies properties 1 and 2.

The above decision procedure will tell us whether a grammar is $BCP[m, n]$ only for given m and n. Therefore given a grammar G, we can first determine whether G is $BCP[1, 1]$, and if not, we can then determine whether G is $BCP[2, 2]$, and so forth. Before beginning this sequence of tests, we would like to be assured that at some point the decision procedure will respond affirmatively. That is, we would like to be able to decide the more general question, do there exist integers m and n such that G is $BCP[m, n]$. We show that this question is recursively undecidable for context-free grammars with the help of the following:

Lemma: If G is a linear context-free grammar, then G is BC(m, n) iff G is BCP[m, n].

Proof: 1) If G is BC(m, n), it is BCP[m, n] since $S_{BC(m,n)} \subset S_{BCP[m,n]}$ as was shown in section 3.

2) If G is BCP[m, n], i.e. if every sentential form of G has a phrase occurring in a parsing context of order [m, n] or less, then every derivation context of order [m, n] is a parsing context since every sentential form has only one handle. Therefore G is BC(m, n).

Knuth has shown that the problem to determine of an arbitrary linear context-free grammar G whether there exist integers m and n such that G is BC(m, n) is recursively unsolvable [KNU]. Since G is BC(m, n) iff G is BCP[m, n] for linear G, we have immediately the following:

Theorem: The problem to determine of an arbitrary context-free grammar G whether there exist integers m and n such that G is BCP[m, n] is recursively unsolvable.

BIBLIOGRAPHY

[BPS] Y. Bar-Hillel, M. Perles, and E. Shamir, On formal properties of simple phrase structure grammars, in "Language and Information," Addison-Wesley, Reading, Mass., 1964.

[CHM] N. Chomsky, Formal Properties of Grammars, in D. Luce, R. Bush, and E. Galanter (eds.), "Handbook of Mathematical Phychology," Wiley, New York, 1963.

[ERL] J. C.Earley, An efficient context-free parsing algorithm, Ph.D. dissertation, Carnegie-Mellon University, 1968.

[FL1] R. W. Floyd, Bounded context syntactic analysis, C. ACM, V7, 1964, pp. 64-67.

[FL2] R. W. Floyd, A descriptive language for symbol manipulation, J. ACM, V8, 1961, pp. 579-584.

[FL3] R. W. Floyd, On the non-existence of a phrase structure grammar for ALGOL 60, C. ACM, V5, 1962, pp. 483-484.

[GIN] S. Ginsburg, "The Mathematical Theory of Context-free Languages," McGraw Hill, New York, 1966.

[HST] J. Hartmanis and R. E. Stearns, On the computational complexity of algorithms, Trans. A.M.S., 117, No. 5, May, 1965.

[KNU] D. E. Knuth, On the translation of language from left to right, Info. and Cont., V8, 1965, pp. 607-639.

[LYN] W. C. Lynch, Ambiguities in Backus normal form languages, Ph.D. dissertation, University of Wisconsin, January, 1963.

[RS] M. O. Rabin and D. Scott, Finite automata and their decision problems, IBM J. Res. Develop., V3, 1959, pp. 114-125.

[WW] N. Wirth and H. Weber, EULER, C. ACM, V9, 1966, pp. 89-99.

[YNG] D. H. Younger, Recognition and parsing of context-free languages in time n^3, Inf. and Cont., V10, Feb., 1967.