BOUNDED CONTEXT PARSABLE GRAMMARS

John H. Williams

TR 72-127

April 1972

Department of Computer Science
Cornell University
Ithaca, New York 14853

Bounded Context Parsable Grammars

John H. Williams
Cornell University, Ithaca, New York

Abstract

In this paper we extend Floyd's notion of parsing by bounded context to define the Bounded Context Parsable Grammars, a class of recursive subsets of context free grammars for which we can construct linear time parsers. It is shown that the set of languages of the grammars thus defined properly contains the set of deterministic languages without the empty sentence.

Key words and phrases: bounded context grammars, linear time parsers, deterministic context free languages

Introduction

Context-free grammars are well known to be a useful model for the formal description of the syntax of programming languages. 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 syntactically analyzed, or parsed, efficiently. Consequently, much work has been done to discover subsets of the set of context-free gramma 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; e.g. Floyd (1964), Knuth (1965), Lynch (196 Earley (1968), Wirth and Weher (1966) and Deremer (1971). The goal of these investigations has been to discover subsets that are suffi ciently 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 restriction of the subset.

In this paper we present a class of such subsets and call them the class of Bounded Context Parsable grammars. In Section 1 we present the notation and previous work upon which this work 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 linear 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 without the empty sentence. In Section 4 we show that the problem to determine whether a grammar is in a particular member of the class of Bounded Context Parsable grammars is effectively decidable but that the problem to determine whether a grammar is in the union over the class is recursively undecidable. In Section 5 we briefly remark on some possible extensions of these ideas. The ideas presented are intended to be primarily of a theoretical interest, but we suggest a potential practical application in the concluding section.

Section 1: Definitions and Notation

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 $|X|$. We shall express a context-free 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: $\phi \rightarrow \psi$ iff

i) $\phi = \chi_1 A \chi_2$

ii) $\psi = \chi_1 x \chi_2$

iii) $\chi_1, \chi_2 \in V^*$ 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 i^{th} member as $A_i \rightarrow x_i$. Such a string, x_i, will be called a right part. A production of the form $A \rightarrow B$ where

$B \in V_N$ will be called a <u>1-production</u>. If $\phi \rightarrow^{+} \psi$, then ϕ is said to <u>derive</u> ψ and ψ is said to be a <u>ϕ-derivative</u>. If $A \rightarrow x$ is a production, x is said to be an <u>immediate</u> A-derivative. If $\phi \in V^{*}$, $|\phi|$ will denote the length of ϕ, <u>i.e.</u> the number of characters in ϕ, and ϕ^{R} will denote the reversal or mirror image of ϕ.

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

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

A grammar is said to be <u>reduced</u> if for every $A \in V_N$,

i) $(\exists\phi)(\exists\psi) S \rightarrow^{*} \phi A \psi \quad \phi,\psi \in V^{*}$
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 <u>linear</u> if every right part contains at most one nonterminal character; clearly if G is linear, then every $\phi \in SF(G)$ has at most one nonterminal character.

Given a grammar G we define its <u>description grammar</u> by

$G' = (V', P', V_T', S')$ where

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

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

iii) $S' = S$

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

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

$$m(A) = A, \text{ if } A \in V_N$$
$$m(a) = a, \text{ if } a \in V_T ,$$
$$m([) = \epsilon ,$$
$$m(\epsilon) = \epsilon ,$$
$$m(]_i) = \epsilon , \quad (1 \leq i \leq |P|),$$

and $m(\phi) = m(x_1)m(x_2)\ldots m(x_n)$ if $\phi = x_1x_2\ldots x_n$. Thus m is a homomorphism with respect to concatenation. Let $\bar{m}$ be the restriction of m to $SF(G')$. In general for $\phi \in SF(G)$, $\bar{m}^{-1}(\phi)$ will be a subset of $SF(G')$.

A grammar will be said to be <u>unambiguous</u> if $\bar{m}$ is 1:1 and <u>ambiguous</u> otherwise.

If $A_i \rightarrow x_i \in P$, $\phi \in SF(G)$, and $\phi = \phi_1 x_i \phi_2$, the right part x_i is said to be a <u>simple phrase</u> of ϕ if there is a string in $\bar{m}^{-1}(\phi)$ of the form $\psi_1[x_i]_i\psi_2$ such that $m(\psi_1) = \phi_1$ and $m(\psi_2) = \phi_2$.

Given a sentence ϕ, the process of computing $\bar{m}^{-1}(\phi)$ is called <u>parsing</u>. Parsers are usually constructed so that given an input, ϕ:

i) if $\phi \in L(G)$, the parser outputs $\bar{m}^{-1}(\phi)$
ii) if $\phi \notin L(G)$, the parser outputs an error message.

The essence of the parsing process is to be able to decide which right parts of a sentential form are simple phrases, make the appropriate reductions in the sentential form, and then repeat the process on the new sentential form thus obtained. Earley(1968) has shown that for any grammar, G, there exists a parser for G that will parse any sentence $\phi \in L(G)$ in a time proportional to $|\phi|^3$. He also has shown that for any unambiguous grammar G, there is a parser that will parse any $\phi \in L(G)$ in time proportional to $|\phi|^2$. When constructing programming languages we are interested in grammars whose sentences can be parsed in time linearly proportional to $|\phi|$. In particular we wish to discover large subsets C of the set 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 linear time parser for any G in C .

There have been several such subsets defined in the past few years. While a comparison of the Bounded Context Parsable grammars with all of these subsets would be beyond the scope of this paper, we will compare them with three notable subsets: the Bounded Context, BC(M,N), grammars of Floyd(1964) upon whose work the present paper is based, the LR(k) grammars of Knuth(1965), and the version of precedence grammars called Simple Mixed Strategy Precedence, SMSP, grammars by Aho, Denning, and Ullman (1972).

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 linear 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 $A_i \to x_i$ if:

i) $w,y \in V^*$, and
ii) $S \to^* \ldots wA_iy\ldots$.

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

i) (w,y) is a derivation context for $A_i \to x_i$, and
ii) if $\phi \in SF(G)$ is of the form $\phi = vwx_iyz$ for some $v, z \in V^*$ and $\delta \in \overline{m}^{-1}(\phi)$, 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 $A_i \to x_i$ if x_i is always an immediate A_i-derivative whenever it appears in a sentential form ϕ in the context (w,y); i.e. $\phi = \ldots wx_iy\ldots$. 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 <u>order</u> <u>$[m,n]$</u> if $|w| = m$ and $|y| = n$. 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^{th} production of order $[m,n]$ or less will be denoted by $DC_i[m,n]$. The set of parsing contexts for the i^{th} production of order $[m,n]$ or less will be denoted by $PC_i[m,n]$ and its j^{th} member will be denoted by (w_{ij},y_{ij}) .

A particular occurrence of a right part x_i in a sentential form ϕ will be said to <u>occur in a parsing context of order $[m,n]$</u> 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 <u>Bounded Context Parsable with left bound m and right bound n</u>, (BCP[m,n]) , if every sentential form of G contains at least one right part occurring in a parsing context of order [m,n] or less.

To illustrate these definitions consider the grammar,

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

$S \to \vdash AEa \dashv$
$S \to \vdash BEb \dashv$
$E \to eE$
$E \to e$
$A \to e$
$B \to e$

For each of the six productions we list the set of derivation contexts or order [2,2] or less and the set of parsing contexts of order [2,2] or less. We will use the Cartesian product $(\{s_1,s_2,s_3\} \times \{s_4,s_5\})$ 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) $S \to \vdash AEa \dashv$
$DC_1[2,2]:(\epsilon,\epsilon)$
$PC_1[2,2]:(\epsilon,\epsilon)$

2) $S \to \vdash BEb \dashv$
$DC_2[2,2]:(\epsilon,\epsilon)$
$PC_2[2,2]:(\epsilon,\epsilon)$

3) $E \to eE$
$DC_3[2,2]:(\{\epsilon,e,ee,\vdash e\} \times \{\epsilon,a,b,a\dashv,b\dashv\})$
$(\{A,\vdash A,Ae\} \times \{\epsilon,a,a\dashv\})$
$(\{B,\vdash B,Be\} \times \{\epsilon,b,b\dashv\})$

$PC_3[2,2]:(\{e,ee,\vdash e\} \times \{\epsilon,a,b,a\dashv,b\dashv\})$
$(\{A,\vdash A,Ae\} \times \{\epsilon,a,a\dashv\})$
$(\{B,\vdash B,Be\} \times \{\epsilon,b,b\dashv\})$

4) $E \to e$
$DC_4[2,2]$:(the same as $DC_3[2,2]$ by the definition of derivation context)

$PC_4[2,2]:(\{\epsilon,e,\vdash e,ee\} \times \{a,b,a\dashv,b\dashv\})$
$(\{A,\vdash A,Ae\} \times \{a,a\dashv\})$
$(\{B,\vdash B,Be\} \times \{b,b\dashv\})$

5) $A \rightarrow e$
$DC_5[2,2]:(\{\epsilon,\vdash\} \times \{\epsilon, E, e, Ea, eE, ea, ee\})$
$PC_5[2,2]:(\vdash, Ea), (\vdash, ea)$

6) $B \rightarrow e$
$DC_6[2,2]:(\{\epsilon,\vdash\} \times \{\epsilon, E, e, Eb, eE, eb, ee\})$
$PC_6[2,2]:(\vdash, Eb), (\vdash, eb)$

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, <u>i.e.</u> $S \rightarrow^* \ldots E \ldots$, but (ϵ,ϵ) is not a parsing context for 3; <u>e.g.</u> in the sentential form, $\vdash eEa \dashv$, the occurrence of eE is not an E-derivative.

iii) (e,ϵ) is a parsing context for 3; <u>i.e.</u> in any sentential form, $\ldots eeE \ldots$, the 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; <u>i.e.</u> while $BeE\ldots$ and $AeE\ldots$ may occur in a sentential form, the leftmost occurrence of e in $\vdash eeE\ldots$ 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 right part 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 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 right part occurring in a parsing context of order [2,2] or less may be seen by analyzing the possible cases:

1.) Form 1 contains the right part, $\vdash AEa \dashv$ 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 $(\dashv A,a)$ 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 $A \to e$ of order less than [1,2], if there is any sentential form of G_1 in which the only simple phrase is 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 simple phrase occurring in parsing context, we have by induction on the number of steps in the derivation of a sequential form the

<u>Theorem</u>: <u>Every BCP[m,n] grammar is unambiguous</u>.

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

i) set $\psi_1 = \phi$ and $j = 1$

ii) search ψ_j for a right part 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 simple 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 type of reduction system (Floyd 1961) 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:

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

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

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

Class II consists of the $|V|$ rules,

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

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 simple phrase occurring in parsing context. For example, with the grammar $(\{S,a\}, \{S \rightarrow aS, S \rightarrow a\}, \{a\}, S)$, since the only simple phrase 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 $\frac{|\phi|^2}{2}$.

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 q^{th} character in ψ_j , then no simple 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 immediately to the right 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:

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

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 $|\phi|$ 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 $\phi \in L(G)$ is $K \cdot |\phi|$ or less.

Proof: 1) Let $p = |P|$.

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 $2 \cdot p\ell$ since at least one terminal character or

one additional nonterminal character must be produced after the application of every p rewriting rules in a rightmost (or leftmost) derivation of ϕ.

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 n 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, R_1, 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 $vwx_iyz\Delta \rightarrow vwA_i\Delta yz$ since the simple phrases reduced by the rule $wx_iy\Delta \rightarrow xwA_i\Delta y$ 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 occurrence 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 simple 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), SMSP, LR(n), and RL(m) . We will also show that the set of languages that are BCP[m,n] properly contains the set of deterministic languages without the empty sentence.

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

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

We will let G_{BCP} denote the set, $\bigcup_{m,n \geq 0} G_{BCP[m,n]}$; i.e. a grammar G is in G_{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 satisfy properties 1 and 2 as given in Section 1 above. We present the results of these comparisons as a Venn diagram in figure 1.

Theorem: $G_{BCP} \supsetneq G_{BC}$.

Proof: 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 in Floyd (1964). Since every sentential form has some right part occurring as a simple phrase and therefore occurring in a derivation context of order [m,n], every sentential form has a simple 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 $A \rightarrow e$, <u>i.e.</u> $S \rightarrow^* \cdots \vdash^m A e^n \cdots$ (notice 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, <u>e.g.</u> $\vdash^m e e^n b \dashv$. Therefore G_1 is not in $G_{BC(m,n)}$ for any m or n ; i.e. $G_1 \notin G_{BC}$. Q.E.D.

In Williams(1969) we have presented a definition of Left to Right Bounded Context Parsable (BRC(m,n)) grammars that differs from Floyd's original definition of this notion. Since they are not necessary for the purposes of this paper, we will give only a brief description of them here. Informally, a grammar is BRC(m,n) if in every sentential form of the grammar the leftmost simple phrase of the sentential form occurs in a parsing context of order [m,n] or less. Thus we consider the set of derivation contexts that a nonterminal may have when it is the leftmost nonterminal in the sentential form, and we call that set the left restricted derivation contexts of order [m,n] for the i^{th} production $(LDC_i[m,n])$. Then a grammar is BRC(m,n) if $LDC_i[m,n] \subset PC_i[m,n]$ for each production $A_i \rightarrow x_i$. The set of grammars defined this way is incommensurate with the BRC grammars as defined by Floyd, and in fact neither of these definitions fully captures the notion of grammars whose sentences are parsable using bounded context under a left to right scan.

While the BRC grammars thus defined are not needed here, they are useful in motivating the definition of BCP grammars. The BC grammars ensure that every simple phrase of a sentential form occurs in a parsing context, the BRC grammars ensure only that the leftmost simple phrase of each sentential form occurs in a parsing context, and the BCP grammars ensure only that at least one simple phrase of every sentential form occurs in a parsing context.

Theorem: $\underline{G_{BCP} \supsetneq G_{SMSP}}$

Proof: 1) To show this containment we first recall the definition of SMSP . Given $G = (V,P,V_T,S)$ three auxiliary relations α, λ, and ρ are defined on V as:

$a \; \alpha \; b$	iff	$\exists$	$c \rightarrow \ldots ab \ldots$	in P
$a \; \lambda \; b$	iff	$\exists$	$a \rightarrow b \ldots$	in P
$a \; \rho \; b$	iff	$\exists$	$b \rightarrow \ldots a$	in P

Then G is SMSP if

i) $\alpha\lambda^* \cap \rho^+\alpha\lambda^*$ is empty

ii) if $A \rightarrow \ldots ax$ and $B \rightarrow x$ are both in P where $x \in V^+$ and $a \in V$, then not $a \; \alpha\lambda^* B$.

iii) if $A \to x$ and $B \to x$ are both in P then not $A(\alpha\lambda^*)^T \alpha\lambda^* B$ where $(\alpha\lambda^*)^T$ denotes the transpose of the $\alpha\lambda^*$ relation.

2) Assume G is SMSP but not BCP(1,1) . Then there is a sentential form none of whose simple phrases occurs in a parsing context; in particular the leftmost simple phrase, say $x_i = x_{i1} \cdots x_{i\ell}$, occurs in a context (w,y) not in $PC_i[1,1]$.

3) Since x_i is the leftmost simple phrase and G is SMSP then, $x_{i\ell}\ \rho^+\alpha\lambda^* y$, and the $\alpha\lambda^*$ relation holds between all character pairs to the left of $x_{i\ell}$.

4) Since $(w,y) \notin PC_i[1,1]$ there must be some production other than $A_i \to x_i$ that can produce the leftmost simple phrase in $\ldots wx_i y \ldots$, say $A_j \to x_j$. But the tail of x_j must coincide with the tail of x_i otherwise there would be a precedence conflict. For if the tail of x_j is to the left of the tail of x_i , both $\alpha\lambda^*$ and $\rho^+\alpha\lambda^*$ would be true at that point, and if to the right, both $\alpha\lambda^*$ and $\rho^+\alpha\lambda^*$ would hold between $x_{i\ell}$ and y .

5) Similarly the heads of x_i and x_j must coincide since otherwise one is a proper suffix of the other, <u>i.e.</u> $x_i = \cdots ax_j$, and a $\alpha\lambda^* A_j$ in violation of condition 2) of the definition of SMSP .

6) Therefore $x_i = x_j$, and this implies that $i = j$ since otherwise $A_i(\alpha\lambda^*)^T \alpha\lambda^* A_j$ in violation of condition 3) of the definition of SMSP .

7) Therefore every SMSP grammar is BCP . G_1 shows the containment to be proper.

Q.E.D.

When we compare G_{BCP} with the set of grammars that are LR(n) for some n , G_{LR} , it is seen that neither set contains the other. G_1 is not LR(n) for any value of n since the leftmost simple phrase 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. However, G_1 is RL(1) . For an example of a grammar that is BCP[2,2] but is neither LR(n) nor RL(m) for any values of m and n , consider the grammar G_2 with productions:

$$S \rightarrow \vdash A\ E\ a\ E\ A \dashv$$
$$S \rightarrow \vdash B\ E\ b\ E\ B \dashv$$
$$E \rightarrow eE$$
$$E \rightarrow e$$
$$A \rightarrow e$$
$$B \rightarrow e$$

Knuth (1965) has shown 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) . $L(G_3) = \{a^k b c^k | k \geq 0\}$, 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 simple phrase in any sentential form of G_3 , the sentential form $\vdash a^k b c^k \dashv$, where $k = \max(m,n)$, contains no simple 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 figure 1.

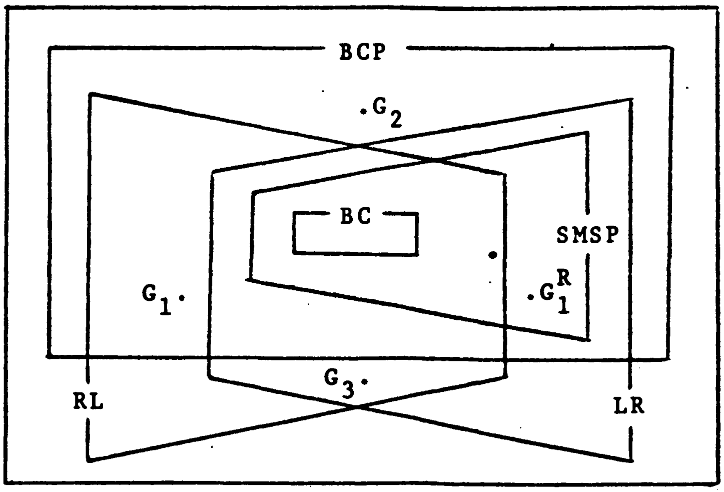

Figure 1. Venn diagram of subsets of context-free grammars.

As is obvious from the definition as 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.

Since the definitions of the bounded context condition do not allow rules of the form $A \rightarrow \epsilon$, we will restrict our comparisons to the universe of context free languages not containing ϵ . Accordingly we will let D denote the set of deterministic languages that do not contain the empty sentence. 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, SMSP, LR or RL if there exists a G such that L = L(G) and G is BC(m,n), SMSP, LR(n) or RL(m) for some values of m and n. Knuth has shown that every language in D is LR(1) , i.e. there exists an LR(1) grammar for it; moreover, every LR(k) language is deterministic. He also shows 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 . 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 $B \to b$; (ϵ,c) and (ϵ,D) are parsing contexts for $D \to b$. Any sentential form of G_4 has either a right part 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 simple phrase occurring in a parsing context of order $[0,1]$ or less and the grammar is seen to be BCP[0,1] .

Aho, Denning, and Ullman (1972) have shown that if $L \in D$ then there is a SMSP G such that $L(G) = L$. Since every SMSP grammar is BCP we have shown that

$$L_{BCP} \supsetneq D$$

That the inclusion is proper is demonstrated by the existence of G_4 .

We will let D^R denote the set of languages not containing ϵ whose reversals are deterministic. $L(G_4) \in D^R$ since G_4^R 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) $S \to \vdash S_1 d S_2 \dashv$

2) $S_1 \to U_1$	3) $S_2 \to U_2$
4) $S_1 \to V_1 c$	5) $S_2 \to cV_2$
6) $U_1 \to aU_1B_1$	7) $U_2 \to B_2U_2a$

8) $U_1 \to aB_1$

9) $U_2 \to B_2a$

10) $V_1 \to aV_1D_1D_1$

11) $V_2 \to D_2D_2V_2a$

12) $V_1 \to aD_1D_1$

13) $V_2 \to D_2D_2a$

14) $B_1 \to b$

15) $B_2 \to b$

16) $D_1 \to b$

17) $D_2 \to b$

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 figure 2.

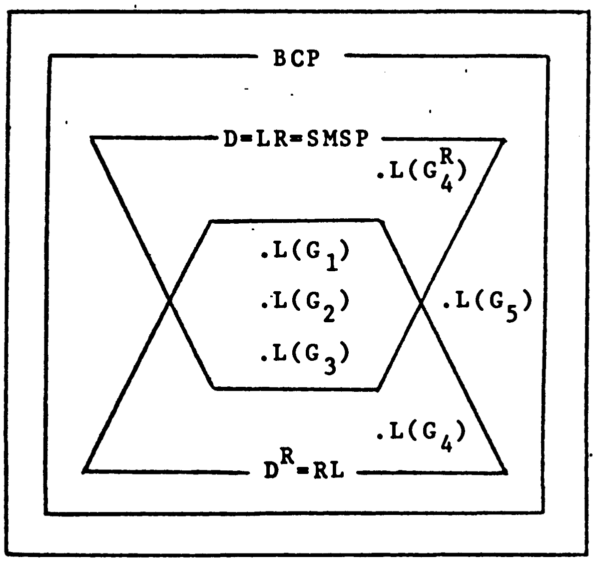

Figure 2. Venn diagram of subsets of context-free languages not containing ϵ .

Section 4: Testing for Bounded Context Parsability

In this section we will show that for every m and n the set $G_{BCP[m,n]}$ is decidable (property 1) but that G_{BCP} is not.

So far we have given no general method for determining of an arbitrary 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, $p = |P|$, 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 (Bar Hillel,1964) by deciding f each of the pairs (w,y) such that $|w| \leq m$ and $|y| \leq n$, whether $S \overset{*}{\rightarrow} \ldots wA_iy \ldots$.
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 i^{th}

production. Floyd's (1964) method of analyzing the sixteen sets of relations 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 simple 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 $A \to x$ is a production of P , then $A' \to x'$ 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) $A' \to A$ is in P' for all $A \in V_N$.
   iii) If $S \to x$ is in P , then $S'' \to x'$ 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 simple phrase.

6) Now, G is BCP[m,n]
 iff every sentential form of G has a simple phrase occurring in a parsing context of order [m,n] or less
 iff $L(G') \subset R$

7) There is an effective procedure for determining whether the language of an arbitrary context-free grammar is contained in a regular set (Hopcroft & Ullman, 1969).
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 , $G_{BCP[m,n]}$ 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 $G_{BC(m,n)} \subset G_{BCP[m,n]}$ as was shown in Section 3.

2) If G is BCP[m,n], i.e. if every sentential form of G has a simple 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 simple phrase. Therefore G is BC(m,n) .

Knuth(1965) 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. 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.

Section 5: Conclusion

We have presented a large class of grammars satisfying the two properties described in Section 1. While these properties are certainly desirable ones for a class of grammars to have if we are to base a formal method of syntactic analysis on them, they don't necessarily imply that the resulting method will be efficient. The constant of proportionality and the space required for tables to drive the parser must also be reasonably small if the method is to be used in actual compilers. We do not present this class as one well suited to practical use but rather as one of interest because it contains some nondeterministic languages as well as all the deterministic languages that do not contain the empty sentence. Of course it is not impossible that these notions might be adopted in a working compiler, possibly on one of the new "parallel" machines where one does not restrict oneself to a strictly left to right scan. An interesting example of a method that initially appeared to be too large and unwieldy for practical use although theoretically very powerful is the LR(k) grammars of Knuth which Deremer(1971) has modified to produce one of the most powerful and efficient methods currently available.

The definition could be modified to include productions with empty right parts as in the definition of LR(k) grammars so that languages containing the empty sentence could be given BCP grammars. A more interesting continuation of this work, however, would be a characterization of the BCP languages. They are known to transcend the deterministic languages and lie within the unambiguous languages, but it is unknown just how large the class is.

REFERENCES

Aho, A.V., P.J. Denning, and J.D. Ullman (1972), Weak and Mixed Strategy Precedence Parsing, Journal ACM, 19, 225-243.

Bar-Hillel, Y., M. Perles, and E. Shamir (1964), On Formal Properties of Simple Phrase Structure Grammars, in Language and Information, Addison-Wesley, Reading, Mass.

Colmerauer, A. (1970), Total Precedence Relations, Journal ACM, 17, 14-30.

Deremer, F.L. (1971), Simple LR(k) Grammars, CACM, 14, 7, 453-460

Earley, J.C. (1968), An Efficient Context-Free Parsing Algorithm, Ph.D. Dissertation, Carnegie-Mellon University.

Floyd, R.W. (1964), Bounded Context Syntactic Analysis, CACM, 7, 64-

Floyd, R.W. (1961), A Descriptive Language for Symbol Manipulation, Journal ACM, 8, 579-584.

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

Hopcroft, J.E., and J.D. Ullman (1969), Formal Languages and their Relation to Automata, Addison-Wesley, Reading, Mass.

Knuth, D.E. (1965), On the Translation of Language from Left to Right, Information and Control 8, 607-639.

Lynch, W.D. (1963), Ambiguities in Backus Normal Form Languages. Ph.D. Dissertation, University of Wisconsin, January.

Rabin, M.O. and D. Scott (1959), Finite Automata and their Decision Problems, IBM J. Res. Develop., 3, 114-125.

Williams, J.H. (1969), Bounded Context Parsable Grammars, Technical Report No. 58, Computer Science Department, University of Wisconsin, March.

Wirth, N. and H. Weber (1966), EULER, CACM 9, 89-99.