JOURNAL OF COMPUTER AND SYSTEM SCIENCES: 3, 233–247 (1969)

# Scattered Context Grammars*

SHEILA GREIBACH[†] AND JOHN HOPCROFT[‡]

*Air Force Cambridge Research Laboratories, Office of Aerospace Research, USAF, Bedford, Massachusetts and Air Force Office of Scientific Research, Office of Aerospace Research, USAF, Arlington, Virginia*

Received April 19, 1968

ABSTRACT

Scattered context grammars are defined and the closure properties of the family of languages generated are considered. This family of languages is contained in the family of context sensitive languages and contains all languages accepted by linear time nondeterministic Turing machines.

## INTRODUCTION

Recently there have been many attempts to specify both natural and programming languages by means of a syntactic definition. The syntactic definition should be such that the semantic interpretation of the sentences can be attached to the syntactic structure. A major difficulty with this method is the lack of a suitable class of grammars for describing the structures that occur. Context-free grammars are too weak since they are not capable of defining a language such as $\{ww \mid w \text{ in } \{0, 1\}^*\}$ which requires that information be transmitted between widely separated parts of the sentences. Context-sensitive languages are too powerful to be of practical use since they shed little light on the problem of attaching meanings to sentences. In generating a sentence of a language, a context-sensitive grammar may send a nonterminal symbol back and forth through the sentence to transmit information. This process usually has no relation to any desired semantic interpretation. Ideally we would like the capability of transmitting information between widely separated parts of a sentence without the necessity of sending a nonterminal symbol back and forth to transmit the information. This suggests considering grammars in which rewriting a symbol depends on context

* This research was supported in part by the Air Force Cambridge Research Laboratories, Office of Aerospace Research, USAF, under contract F1962867C0008, and by the Air Force Office of Scientific Research, Office of Aerospace Research, USAF, under AFOSR Grant No. AF-AFOSR-1203-67.

[†]Harvard University and Consultant System Development Corporation.

[‡] Cornell University and Consultant System Development Corporation





571/3/3-1

as in a context-sensitive grammar but a symbol can be rewritten even if the context is not adjacent to the symbol. This leads us to the concept of a scattered context grammar about which this paper is concerned.

The paper is divided into three sections. The first section is devoted to definitions and the development of a normal form for scattered context grammers. The second section considers closure properties. The family of scattered context languages, is shown to be an abstract family of languages (i.e., closed under union, product, $+$, $\epsilon$-free homomorphism, inverse homomorphism and intersection with a regular set). Furthermore, the family is shown to be closed under intersection, linear erasing, $\epsilon$-free substitution, permutation and reversal. It is not closed under arbitrary homomorphism or quotient with a regular set.

Section 3 is an attempt to determine the generative power of the scattered context grammars. From the definition of a scattered context grammar, it is obvious that they generate only context-sensitive languages. A corollary in Section 1 shows that the family of languages generated properly contains the family of $\epsilon$-free context-free languages. The main result of Section 3 is that the family of scattered context languages include all languages accepted by quasi-realtime $n$-tape pushdown automata and hence (by the results in Section 2) all languages accepted in linear time by nondeterministic Turing machines. The authors are, however, unable to show proper containment in the family of context-sensitive languages.

## Section 1. Definitions and Basic Lemmas

In Section 1 we define a scattered context grammar and provide the notation necessary to describe the language generated by such a grammar. We then define a restricted type of scattered context grammar called a 2-limited grammar and show that the families of languages generated are the same. The 2-limited grammar is a very useful normal form and is used extensively throughout the remainder of the paper to simplify constructions.

Definition. *A scattered context grammar* (scg) is a quadruple $G = (V, \Sigma, P, S)$ where:

(1) $V$ is a finite set of symbols, $\Sigma$ is a subset of $V$, and $S$ is in $V - \Sigma$,

(2) $P$ is a finite set of productions of the form $(A_1, ..., A_n) \to (w_1, ..., w_n)$, $n \geqslant 1$, each $A_i$ in $V - \Sigma$, and each $w_i$ in $V^+$.[1]

We now introduce notation to describe the language generated by a scg. Let $(A_1, ..., A_n) \to (w_1, ..., w_n)$ be in $P$ and for $1 \leqslant i \leqslant n + 1$, let $x_i$ be in $V^*$. We

[1] $A^+ = \bigcup_{i=1}^{\infty} A^i$ and $A^* = A^+ \bigcup \{\epsilon\}$, where $\epsilon$ is the empty word and $A^{i+1} = A^i A$ for $i \geqslant 1$.

write $x_1A_1x_2A_2 \cdots x_nA_nx_{n+1} \Rightarrow x_1w_1x_2w_2 \cdots x_nw_nx_{n+1}$. Let $\overset{*}{\Rightarrow}$ be the reflexive transitive closure of $\Rightarrow$. We define the language *generated* by $G$ (denoted $L(G)$) to be the set $L(G) = \{w \text{ in } \Sigma^+ \mid S \overset{*}{\Rightarrow} w\}$. The set $L(G)$ is called a *scattered context language* (scl).

We now define a restricted type of scg.

DEFINITION. *A 2-limited grammar* is a scg $G = (V, \Sigma, P, S)$ such that

(1) $(A_1, \ldots, A_n) \to (w_1, \ldots, w_n)$ in $P$ implies $n \leqslant 2$ and for each $i$, $1 \leqslant |w_i| \leqslant 2$ and $w_i$ is in $(V - \{S\})^*$.

(2) $(A) \to (w)$ in $P$ implies $A = S$.[2]

We will now show by means of two lemmas that every scl can be generated by a 2-limited grammar.

LEMMA 1.1. *If $L \subseteq \Sigma^*$ is a language generated by a* scg $G = (V, \Sigma, P, S)$ *and if $c$ is a symbol not in $\Sigma$, then there is a 2-limited grammar $\bar{G}$ with $L(\bar{G}) = Lc$.*[3]

*Proof.* Let $\bar{n}$ be the number of productions in $P$. Number the productions of $P$ from 1 to $\bar{n}$. Let $(A_{i1}, \ldots, A_{in_i}) \to (w_{i1}, \ldots, w_{in_i})$ be the $i$th production. Let $C$ and $\bar{S}$ be new symbols, let $W = \{[i,j] \mid 0 \leqslant i \leqslant \bar{n}, 1 \leqslant j \leqslant n_i\}$ and let

$$\bar{V} = V \cup \{C, \bar{S}\} \cup W \cup (\{C\} \times \{1, \ldots, \bar{n}\}).$$

Let $\hat{G}$ be the scg $(\bar{V}, \Sigma \cup \{c\}, \hat{P}, \bar{S})$ where $\hat{P}$ is defined as follows.

(1) $(\bar{S}) \to (S[C, i])$ is in $\hat{P}$, for $1 \leqslant i \leqslant \bar{n}$.

(2) For each $i$ such that $n_i = 1$ and for each $k$, $1 \leqslant k \leqslant \bar{n}$,

$$(A_{i1}, [C, i]) \to (w_{i1}, [C, k])$$

is in $\hat{P}$.

(3) For each $i$ such that $n_i > 1$,

(a) $(A_{i1}, [C, i]) \to ([i, 1], C)$,

(b) $([i, j], A_{i,j+1}) \to (w_{ij}, [i, j+1])$, $1 \leqslant j \leqslant n_i - 1$, and

(c) $([i, n_i], C) \to (w_{in_i}, [C, k])$, $1 \leqslant k \leqslant \bar{n}$, are in $\hat{P}$.

(4) For each $i$ such that $n_i = 1$, $(A_{i1}, [C, i]) \to w_{i1}c$ is in $\hat{P}$ and for each $i$ such that $n_i > 1$, $([i, n_i], C) \to w_{in_i}c$ is in $\hat{P}$.

Clearly $L(\hat{G}) = L(G)$. Since for some $i$ and $j$, $w_{ij}$ may be of length greater than two, $\hat{G}$ may not be a 2-limited grammar. However, by making use of standard techniques one can obtain a 2-limited grammar $\bar{G}$ from $\hat{G}$ such that $L(\bar{G}) = L(\hat{G})$.

[2] $|w|$ denotes the length of $w$.

[3] Although not explicitly stated all constructions in this paper are effective.

LEMMA 1.2. *If $L \subseteq \Sigma^+$, $c$ is a symbol not in $\Sigma$ and $G = (V, \Sigma \cup \{c\}, P, S)$ is a 2-limited grammar with $L(G) = Lc$, then there is a 2-limited grammar $\bar{G}$ with $L(\bar{G}) = L$.*

*Proof.* For each $a$ in $\Sigma \cup \{S\}$, let $\bar{a}$ be a new symbol. Let

$$L_1 = \{A_1A_2A_3 \mid S \overset{*}{\underset{G}{\Rightarrow}} A_1A_2A_3\,,\ A_i \text{ in } V\}$$

and

$$L_2 = \{A_1A_2A_3A_4 \mid S \overset{*}{\Rightarrow} A_1A_2A_3A_4\,,\ A_i \text{ in } V\}.$$

Let $h$ be the homomorphism of $V^*$ defined by $h(a) = \bar{a}$ for each $a$ in $\Sigma$ and $h(A) = A$ for each $A$ in $V - \Sigma$. Let $\bar{V} = h(V) \cup \Sigma \cup \{\bar{S}\} \cup (V \times V)$. Let $\hat{G} = (\bar{V}, \Sigma, \hat{P}, \bar{S})$ where for all $a$ and $b$ in $\Sigma$, $A_1$, $A_2$, $A_3$, $A_4$, $A_5$, and $A_6$ in $V$, and $A$ in $h(V)$, $\hat{P}$ is defined as follows.

(1) $\hat{P}$ contains the productions

(a) $(\bar{S}) \to (a)$ if $a$ is in $L$.

(b) $(\bar{S}) \to (h(A_1)[A_2, A_3])$ if $A_1A_2A_3$ is in $L_1$.

(c) $(\bar{S}) \to (h(A_1A_2)[A_3, A_4])$ if $A_1A_2A_3A_4$ is in $L_2$.

(2) If $P$ contains $(A_1, A_2) \to (w_1, w_2)$, then $\hat{P}$ contains the productions,

(a) $(A_1, A_2) \to (h(w_1), h(w_2))$,

(b) $(A_1, [A_2, A_3]) \to \begin{cases} (h(w_1), [w_2, A_3]) \text{ if } |w_2| = 1 \\ (h(w_1), h(A_4)[A_5, A_3]) \text{ if } w_2 = A_4A_5 \end{cases}$

(c) $(A_1, [A_3, A_2]) \to \begin{cases} (h(w_1), [A_3, w_2]) \text{ if } |w_2| = 1 \\ (h(w_1), A_3[A_4, A_5]) \text{ if } w_2 = A_4A_5 \end{cases}$

(d) $(A, [A_1, A_2]) \to \begin{cases} (A, [w_1, w_2]) \text{ if } |w_1| = |w_2| = 1 \\ (A, h(A_3)[A_4, A_5]) \text{ if } w_1w_2 = A_3A_4A_5 \\ (A, h(A_3A_4)[A_5, A_6]) \text{ if } w_1w_2 = A_3A_4A_5A_6 \end{cases}$

(3) $\hat{P}$ contains $(\bar{a}, [b, c]) \to (a, [b, c])$ and $(\bar{a}, [b, c]) \to (a, b)$.

(Note that the construction simply combines the $c$ with the symbol to its left. The reason for introducing a new symbol $\bar{a}$ for each $a$ in $\Sigma$ is to guarantee that there will always exist a nonterminal $A$ whenever a production of type (2d) is to be applied and a nonterminal $\bar{a}$ which enables $[b, c]$ to be converted to a $b$ by a production of type (3). Clearly $L(\hat{G}) = L$. $\hat{G}$ may not be a 2-limited grammar since in (2d),

$$|\, h(A_3A_4)[A_3, A_6]| = 3.$$

Once again by standard techniques one can obtain a 2-limited grammar $\bar{G}$ from $\hat{G}$ such that $L(\bar{G}) = L(\hat{G})$.

It follows immediately from Lemmas 1.1 and 1.2 that every scl is generated by a 2-limited grammar. Thus we state the following theorem.

THEOREM 1.1. *If $G$ is a* scg, *then there exists a 2-limited grammar $\bar{G}$ with $L(\bar{G}) = L(G)$.*

## SECTION 2. CLOSURE PROPERTIES

In this section we consider the closure properties of the family of scl under various operations and show that the family of scl properly contains the family of $\epsilon$-free[4] context-free languages [2]. The main closure results are that the family of scl is an abstract family of languages [3] (i.e. closed under union, product, $+$, $\epsilon$-free homomorphism,[5] inverse homomorphism, and intersection with a regular set) which is closed under intersection, $\epsilon$-free substitution[6] and linear erasing.[7]

We begin with a preliminary lemma.

LEMMA 2.1. *The family of* scl *is closed under substitution by an $\epsilon$-free context free language, intersection with a regular set and permutations.*

*Proof.* (1) Substitution by $\epsilon$-free context-free languages.

Let $G = (V, \Sigma, P, S)$ be scg. For each $a$ in $\Sigma$ let $\bar{a}$ be a new symbol and let $\Sigma_1$ be the set $\{\bar{a}/a \text{ in } \Sigma\}$. Let $h$ be the homomorphism of $V^*$ defined by $h(a) = \bar{a}$ for each $a$ in $\Sigma$ and $h(A) = A$ for each $A$ in $V - \Sigma$. For each $a$ in $\Sigma$ let $L_a$ be a cfl and let $G_a = (V_a, \Sigma_a, P_a, S_a)$ be a cfg with $L(G_a) = L_a$. Without loss of generality we can assume that for $a \neq b$, $(V_a - \Sigma_a) \cap (V - \Sigma) = (V_b - \Sigma_b) \cap (V_a - \Sigma_a) = \varnothing$, and $P_a$ contains no rule $A \to \epsilon$. Let $\bar{V} = h(V) \cup (\bigcup_a V_a)$ and $\bar{\Sigma} = \bigcup_a \Sigma_a$. Let $\bar{G} = (\bar{V}, \bar{\Sigma}, \bar{P}, S)$ where $\bar{P}$ is defined as follows.

(a) If $(A_1, ..., A_n) \to (w_1, ..., w_n)$ is in $P$, then $(A_1, ..., A_n) \to (h(w_1), ..., h(w_n))$ is in $\bar{P}$.

[4] A language is *$\epsilon$-free* if it does not contain $\epsilon$; a family of languages is *$\epsilon$-free* if all its members are $\epsilon$-free.

[5] A mapping $f$ is said to be *$\epsilon$-free* if $f(x) \neq \epsilon$ for all $x \neq \epsilon$.

[6] Let $L \subseteq \Sigma^*$. For each $a$ in $\Sigma$ let $L_a \subseteq \Sigma_a{}^*$. Let $\tau$ be the function defined by $\tau(\epsilon) = \{\epsilon\}$, $\tau(a) = L_a$ for each $a$ in $\Sigma$, and $\tau(a_1 \cdots a_k) = \tau(a_1) \cdots \tau(a_k)$ for each $k \geqslant 1$ and $a_i$ in $\Sigma$. Then $\tau$ is called a *substitution.* $\tau$ is extended to $2^{\Sigma^*}$ by defining $\tau(X) = \bigcup_{x \text{ in } X} \tau(x)$ for all $X \subseteq \Sigma^*$. A family $\mathscr{L}_1$ of languages is said to be closed under substitution by a family $\mathscr{L}_2$ if $\tau(L)$ is in $\mathscr{L}_1$ for each $L$ in $\mathscr{L}_1$ and each substitution $\tau$ such that $\tau(a)$ is in $\mathscr{L}_2$ for each $a$, or simply *closed under substitution* in the case where $\mathscr{L}_1 = \mathscr{L}_2$.

[7] A family $\mathscr{L}$ of languages is said to be *closed under linear erasing* if for each $L$ in $\mathscr{L}$, homomorphism $h$ and integer $k$ such that $|w| \leqslant k\,|h(w)|$ for each $w$ in $L$, $h(L)$ is in $\mathscr{L}$.

(b) For each $a$ in $\Sigma$, $(\bar{a}) \rightarrow (\sigma_a)$ is in $\bar{P}$.

(c) If $A \rightarrow w$ is in $P_a$, then $(A) \rightarrow (w)$ is in $\bar{P}$.

If $\tau$ is the substitution on $\Sigma$ defined by $\tau(a) = L_a$, then clearly $L(\bar{G}) = \tau(L(G))$.

(2) Intersection with a regular set.

Let $L$ be a scl and $R$ be a regular set. Let $G = (V, \Sigma, P, S)$ be a 2-limited grammar such that $L(G) = L$ and let $A = (K, \Sigma, \delta, q_0, F)$ be a finite automaton with $L(A) = R$. We define a scg $\bar{G} = (\bar{V}, \Sigma, \bar{P}, S)$, where

$$\bar{V} = \Sigma \cup \{S\} \cup (K \times V \times K) \cup (K \times VV \times K)$$

and where $\bar{P}$ is defined as follows.

(a) For each $p$ in $F$, $(S) \rightarrow ([q_0, S, p])$ is in $\bar{P}$.

(b) If $(A, B) \rightarrow (w_1, w_2)$ is in $P$, then for each $p$, $p'$, $q$ and $q'$ in $K$,

$$([p, A, p'], [q, B, q']) \rightarrow ([p, w_1, p'], [q, w_2, q'])$$

is in $\bar{P}$.

(c) For each $p, p'$ and $p''$ in $K$, $A$ and $B$ in $V$, $([p, AB, p']) \rightarrow ([p, A, p''][p'', B, p'])$ is in $\bar{P}$.

(d) For each $p$ in $K$, $a$ in $\Sigma$, $([p, a, \delta(p, a)]) \rightarrow a$ is in $\bar{P}$.

Clearly $L(\bar{G}) = L(G) \cap R$.

(3) Permutations.

Let $L$ be an scl and let $G = (V, \Sigma, P, S)$ be a scg such that $L(G) = L$. For each $a$ in $\Sigma$ let $\bar{a}$ be a new symbol and let $h$ be the homomorphism of $V^*$ defined by $h(a) = \bar{a}$ for each $a$ in $\Sigma$ and $h(A) = A$ for each $A$ in $V - \Sigma$. Now, let $\bar{V} = h(V) \cup \Sigma$,

$$\bar{P} = \{(A_1, \dots, A_n) \rightarrow (h(w_1), \dots, h(w_n)) | (A_1, \dots, A_n) \rightarrow (w_1, \dots, w_n) \text{ in } P\}$$
$$\cup \{(\bar{a}, \bar{b}) \rightarrow (\bar{b}, \bar{a}) |\, a, b \text{ in } \Sigma\} \cup \{(\bar{a}) \rightarrow (a) |\, a \text{ in } \Sigma\}, \text{ and } \bar{G} = (\bar{V}, \Sigma, \bar{P}, S).$$

Clearly $L(\bar{G})$ contains all and only permutations of members of $L(G)$.

From Lemma 2.1 we can immediately obtain the following two corollaries.

Corollary. *The family of* scl *properly contains the family of $\epsilon$-free context-free languages.*

*Proof.* Since $(\{S, a\}, \{a\}, \{(S) \rightarrow (a)\}, S)$ is a scg, closure under substitution of an $\epsilon$-free cfl clearly implies that every $\epsilon$-free cfl is an scl. But $(abc)^+$ is an $\epsilon$-free cfl and the intersection of the set of all permutations of $(abc)^+$ with the regular set $a^*b^*c^*$ is $\{a^nb^nc^n/n \geqslant 1\}$ which is not context-free.

COROLLARY. *The family of* scl *is closed under* $\epsilon$*-free homomorphism.*

In [3] the concept of an abstract family of languages was introduced. In what follows we shall make use of this concept.

DEFINITION. An *abstract family of languages* (AFL) is a pair $(\Sigma, \mathscr{L})$, or $\mathscr{L}$ when $\Sigma$ is understood, where

(1) $\Sigma$ is a countably infinite set of symbols,

(2) for each $L$ in $\mathscr{L}$ there is a finite set $\Sigma_1 \subseteq \Sigma$ such that $L \subseteq \Sigma_1{}^*$,

(3) $L \neq \varnothing$ for some $L$ in $\mathscr{L}$,

(4) $\mathscr{L}$ is closed under the operations of $\bigcup$, $\cdot$, $+$, inverse homomorphism, $\epsilon$-free homomorphism, and intersection with a regular set.

An AFL $\mathscr{L}$ is said to be *full* if $\mathscr{L}$ is closed under arbitrary homomorphism.

Before showing that the family of scl is an AFL we will establish three preliminary lemmas. First we show that the family of scl is closed under reversal,[8] a result which we will use in the proof of Theorem 2.2.

LEMMA 2.2. *If $L$ is an* scl, *then $L^r$ is an* scl.

*Proof.* Let $G = (V, \Sigma, P, S)$ be a 2-limited grammar with $L(G) = L$. Let $P^r = \{S \to w \mid S \to w^r \text{ in } P\} \cup \{(A, B) \to (w_1, w_2) \mid (B, A) \to (w_2{}^r, w_1{}^r) \text{ in } P\}$. Let $G^r = (V, \Sigma, P^r, S)$. Clearly $L^r = L(G^r)$.

Next we consider a special type of homomorphism.

DEFINITION. A homomorphism $h$ from $\Sigma_1{}^*$ into $\Sigma_2{}^*$ is *k-restricted* on a subset $L$ of $\Sigma_1{}^*$ if $h(w) = \epsilon$ for $w$ in $L$ implies $w = \epsilon$ and $h(w) \neq \epsilon$ for each subword $w$ of length greater than or equal to $k$ of each word in $L$. A family of languages $\mathscr{L}$ is said to be *closed under restricted homomorphism* if $h(L)$ is in $\mathscr{L}$ whenever $L \subseteq \Sigma_1{}^*$ is in $\mathscr{L}$ and $h$ is a homomorphism of $\Sigma_1{}^*$ which is $k - 1$ restricted on $L$ for some $k$.

We now show that the family of scl is closed under restricted homomorphism.

LEMMA 2.3. *The family of* scl *is closed under restricted homomorphism.*[9]

*Proof.* Let $G = (V, \Sigma, P, S)$ be a 2-limited grammar and $h$ a homomorphism of $\Sigma^*$ which is $k$-restricted on $L(G)$. Let $L_1 = L(G) \cap \{\alpha \mid \alpha \text{ in } \Sigma^+, \mid \alpha \mid < k\}$ and $L_2 = \{\alpha \mid S \stackrel{*}{\Rightarrow} \alpha, k \leqslant \mid \alpha \mid \leqslant 2k + 1\}$. For each $\alpha$ in $V^+$ such that $k \leqslant \mid \alpha \mid \leqslant 2k + 1$, let $[\alpha]$ be a new symbol, $S_1 = \{[\alpha] \mid \alpha \text{ in } V^+, k \leqslant \mid \alpha \mid \leqslant 2k + 1\}$ and

[8] Let $\epsilon^r = \epsilon$ and for each word $a_1 \cdots a_n$, $n \geqslant 1$, each $a_i$ in $\Sigma$, let $(a_1 \cdots a_n)^r = a_n \cdots a_1$. For $L \subseteq \Sigma^*$, let $L^r = \{w \mid w^r \text{ in } L\}$. $L^r$ is called the *reversal* of $L$.

[9] Although it is unsolvable whether a homomorphism $h$ is $k$-restricted on a language generated by a scg $G$, given that $h$ is $k$-restricted on $L(G)$ the construction is effective.

$S_2 = \{[\alpha] \mid k \leqslant |\alpha| \leqslant 2k - 1\}$. Let $\bar{S}$ be a new symbol, $\bar{V} = \Sigma \cup S_1 \cup \{\bar{S}\}$, and $\bar{G} = (\bar{V}, \Sigma, \bar{P}, \bar{S})$ where $\bar{P}$ is defined as follows.

(1) For $\alpha$ in $L_1$, $(\bar{S}) \to (h(\alpha))$ is in $\bar{P}$; for $\alpha$ in $L_2$, $(\bar{S}) \to ([\alpha])$ is in $\bar{P}$.

(2) If $(A, B) \to (w_1, w_2)$ is in $P$, then

(a) for each $[x_1Ay_1]$ and $[x_2By_2]$ in $S_2$, $([x_1Ay_1], [x_2By_2]) \to ([x_1w_1y_1], [x_2w_2y_2])$ is in $\bar{P}$, and

(b) for each $[x_3AzBy_3]$ in $S_2$, $([x_3AzBy_3]) \to ([x_3w_1zw_2y_3])$ is in $\bar{P}$.

(3) If $[\alpha]$, $[\beta]$ and $[\gamma]$ are in $S_1$ and $\alpha = \beta\gamma$, then $([\alpha]) \to ([\beta][\gamma])$ is in $\bar{P}$.

(4) For $[\alpha]$ in $S_2 \cap \Sigma^*$ $([\alpha]) \to (h(\alpha))$ is in $\bar{P}$.

Since $h(\alpha) \neq \epsilon$, for $[\alpha]$ in $S_2 \cap \Sigma^*$, $\bar{G}$ is a scg. Clearly $L(\bar{G}) = h(L(G))$.

COROLLARY. The family of scl is closed under inverse homomorphism.

*Proof.* In [6] it was shown that any $\epsilon$-free family of languages closed under intersection with a regular set, $c$-substitution[10] and $k$-limited erasing[11] is closed under inverse homomorphism. Since $c$-substitution is a special case of context-free substitution and since $k$-limited erasing is a special case of a $k$-restricted homomorphism, the familly of scl is closed under inverse homomorphism.

LEMMA 2.4. *If $L \subseteq \Sigma^+$ is a* scl *and $c$ is a symbol not in $\Sigma$, then $c(Lc)^+$ is a* scl.

*Proof.* Let $G = (V, \Sigma, P, S)$ be a double grammar with $L(G) = L$. Let $c$, $C_1$, $C_2$ and $\bar{S}$ be new symbols and $\bar{V} = V \cup \{c, C_1, C_2, \bar{S}\}$. Let $\bar{G} = (\bar{V}, \Sigma, \bar{P}, \bar{S})$ where $\bar{P}$ is defined as follows.

(1) The production $(\bar{S}) \to (C_1SC_2)$ is in $\bar{P}$. Also if $(S) \to (w)$ is in $P$, then $(C_1, S, C_2) \to (C_1, w, C_2)$ is in $\bar{P}$.

(2) If $(A, B) \to (w_1, w_2)$ is in $P$, then $(C_1, A, B, C_2) \to (C_1, w_1, w_2, C_2)$ is in $\bar{P}$.

(3) $(C_1, C_2) \to (c, C_1SC_2)$ is in $\bar{P}$.

(4) $(C_1, C_2) \to (c, c)$ is in $\bar{P}$.

(Observe that generations in $L(\bar{G})$ proceed in phases: $\bar{S} \stackrel{*}{\Rightarrow} C_1wC_2 \Rightarrow cwC_1SC_2 \Rightarrow \cdots$. Between applications of the production $(C_1, C_2) \to (c, C_1SC_2)$, the generation imitates a generation in $L(G)$. Thus $C_1SC_2 \underset{G}{\stackrel{*}{\Rightarrow}} C_1wC_2$ if and only if $S \underset{G}{\stackrel{*}{\Rightarrow}} w$. If $\bar{G}$ "guesses" too soon that the string of symbols between $C_1$ and $C_2$ is terminal, then

$$C_1SC_2 \stackrel{*}{\Rightarrow} C_1xAyC_2 \Rightarrow cxAyC_1SC_2\,.$$

[10] A *c-substitution* is a substitution on $\Sigma^*$ such that $\tau(a) = c^*ac^*$ for each $a$ in $\Sigma$ where $c$ is a new symbol not in $\Sigma$.

[11] A family $\mathscr{L}$ is said to be *closed under k-limited erasing*, $k$ an integer, if whenever $c$ is not in $\Sigma$ and $L$ is in $\mathscr{L}$ with $L \subseteq (\Sigma\{\epsilon, c, ..., c^k\})^*$, $h(c) = \epsilon$ and $h(a) = a$ for each $a$ in $\Sigma$, then $h(L)$ is in $\mathscr{L}$.

Now the symbol $A$ can never be rewritten and thus a terminal string cannot be derived). Clearly $L(\bar{G}) = c(Lc)^+$.

We now prove one of our main results.

THEOREM 2.1. *The family of* scl *is an* AFL.

*Proof.* In [6] it was shown that a family of languages is an AFL if it is closed under union, $+$, $\epsilon$-free homomorphism, inverse homomorphism and intersection with a regular set. We have already shown that the family of scl is closed under $\epsilon$-free homomorphism, inverse homomorphism and intersection with a regular set. Thus we need only closure under union and $+$.

(1) Union.

For $i = 1$ and 2, let $G_i = (V_i, \Sigma_i, P_i, S_i)$ be scg. Without loss of generality let $(V_1 - \Sigma_1) \cap (V_2 - \Sigma_2) = \varnothing$. Let $S$ be a new symbol, $V = V_1 \cup V_2 \cup \{S\}$, $\Sigma = \Sigma_1 \cup \Sigma_2$, $P = \{(S) \to (S_1), (S) \to (S_2)\} \cup P_1 \cup P_2$, and $G = (V, \Sigma, P, S)$. Clearly $L(G) = L(G_1) \cup L(G_2)$.

(2) $+$

Let $L \subseteq \Sigma^*$ be a scl and $c$ be a symbol not in $\Sigma$. By Lemma 2.4, $(cL)^+ c$ is a scl and, since $\epsilon$ is not in $L$, by Lemma 2.3, $L^+$ is an scl.

COROLLARY. *The family of* scl *is closed under product.*

COROLLARY. *The family of* scl *is closed under $\epsilon$-free* gsm[12] *and inverse* gsm *mappings.*

We shall now prove that the family of scl is closed under linear erasing, intersection and substitution. We do this by first proving a special case of intersection and a special case of linear erasing which together yield the general cases.

LEMMA 2.5. *If $L \subseteq \Sigma^+ c \Sigma^+$, $c$ a symbol not in $\Sigma$, is a* scl, *then $L \cap \{wcw \mid w$ in $\Sigma^+\}$ is a* scl.

*Proof.* Let $G = (V, \Sigma \cup \{c\}, P, S)$ be a 2-limited grammar such that $L(G) = L$. Let $\bar{c}$, $d$, $\bar{S}$, $C_1$, $C_2$, and $C_3$ be new symbols and for each $a$ in $\Sigma$, let $\bar{a}$ and $\hat{a}$ be new

[12] A *generalized sequential machine* (gsm) is a 6-tuple $G = (K, \Sigma, \Delta, \delta, \lambda, p_0)$, where (i) $K$, $\Sigma$ and $\Delta$ are finite sets (of *states*, *inputs* and *outputs*, resp.), (ii) $\delta$ is a mapping of $K \times \Sigma$ into $K$ (*next state* function), (iii) $\lambda$ is a mapping of $K \times \Sigma$ into $\Delta^*$ (*output function*), and (iv) $p_0$ is in $K$ (*start* state). The mapping $\delta$ is extended to $K \times \Sigma^*$ by letting $\delta(q, \epsilon) = q$ and $\delta(q_1, xa) = \delta(\delta(q, x), a)$ for all $q$ in $K$, $x$ in $\Sigma^*$ and $a$ in $\Sigma$. The function $\lambda$ is extended to $K \times \Sigma^*$ by letting $\lambda(q, \epsilon) = \epsilon$ and $\lambda(q, xa) = \lambda(q, x)\lambda(\delta(q, x), a)$ for all $q$ in $K$, $x$ in $\Sigma^*$ and $a$ in $\Sigma$. The mapping $G$ defined by $G(L) = \{\lambda(p_0, x) \mid x$ in $L\}$ is called a gsm *mapping*. The mapping $G^{-1}(L) = \{x \mid \lambda(p_0, x)$ in $L\}$ is called an *inverse* gsm *mapping*.

symbols. Let $h$ be the homomorphism on $V^*$ defined by $h(a) = \bar{a}$ for each $a$ in $\Sigma$, $h(c) = \bar{c}$, and $h(A) = A$ for each $A$ in $V - (\Sigma \cup \{c\})$. Let

$$\bar{V} = V \cup \{\bar{c}, d, \bar{S}, C_1, C_2, C_3\} \cup \{\bar{a}, \hat{a} \mid a \text{ in } \Sigma\}.$$

Let $\bar{G} = (\bar{V}, \Sigma \cup \{c, d\}, \bar{P}, \bar{S})$ where $\bar{P}$ is defined as follows.

(1) The production $(\bar{S}) \to (C_1 S C_2)$ is in $\bar{P}$.

(2) (a) If $(S) \to (w)$ is in $P$, then $(C_1, S, C_2) \to (C_1, h(w), C_2)$ is in $\bar{P}$.

(b) If $(A, B) \to (w_1, w_2)$ is in $P$, then $(C_1, A, B, C_2) \to (C_1, h(w_1), h(w_2), C_2)$ is in $\bar{P}$.

(3) (a) $(C_1, \bar{a}, \bar{c}, \bar{a}, C_2) \to (d, \hat{a}, C_3, \hat{a}, C_2)$ is in $\bar{P}$.

(b) $(\hat{a}, \bar{b}, C_3, \hat{a}, \bar{b}, C_2) \to (a, \hat{b}, C_3, a, \hat{b}, C_2)$ is in $P$.

(c) $(\hat{a}, C_3, \hat{a}, C_2) \to (a, c, a, d)$ is in $\bar{P}$.

(The productions of (1) and (2) generate strings of the form $C_1 \bar{a}_1 \cdots \bar{a}_n \bar{c} \bar{b}_1 \cdots \bar{b}_m C_2$ where $a_1 \cdots a_n c b_1 \cdots b_m$ is $L$. Production (3a) initiates "signals" which start at $C_1$ and $\bar{c}$. Note that (3a) can be applied only once. Production (3b) causes the "signals" to propagate left to right in synchronism, comparing symbols. As symbols are compared they are converted to terminals. If any symbol is "skipped" it can never become a terminal. Finally application of (3c) causes the "signal" to disappear).

It is easily shown that $L(\bar{G}) = d(L \cap \{wcw \mid w \text{ in } \Sigma^+\})\, d$. The details are omitted. It follows from Lemma 2.3 that $L \cap \{wcw \mid w \text{ in } \Sigma^+\}$ is a scl.

The construction in Lemma 2.5 involves sending two "signals" through a string, checking if symbols match up, and propagating in one direction such that if a symbol is "skipped" then the generation is ultimately blocked and will never yield a terminal string. We use the same technique to show closure under a very restricted type of linear erasing.

LEMMA 2.6. *Let $c$ and $d$ be symbols not in $\Sigma$. If $L \subseteq \{d^k cw \mid |w| = k \geqslant 1, w \text{ in } \Sigma^+\}$ is an* scl, *then $L_1 = \{w \mid \exists k, d^k cw \text{ is in } L\}$ is an* scl.

*Proof.* Let $\bar{c}$, $\bar{d}$ be new symbols and for each $a$ in $\Sigma$, let $\bar{a}$ be a new symbol. Let $\Sigma_1 = \Sigma \cup \{\bar{c}, \bar{d}\} \cup \{\bar{a} \mid a \text{ in } \Sigma\}$ and let $h$ be the homomorphism defined on $\Sigma \cup \{c, d\}$ by $h(c) = \bar{c}$, $h(d) = \bar{d}$ and $h(a) = \bar{a}$ for $a$ in $\Sigma$. Then $L_2 = h(L)$ is an scl and let $G = (V, \Sigma_1, P, S)$ be a 2-limited grammar with $L(G) = L_2$. Let $C$, $\hat{d}$, $S'$ be new symbols and let $\hat{a}$ be a new symbol for each $a$ in $\Sigma$. Let $\bar{G} = (\bar{V}, \Sigma \cup \{c, d\}, P', S')$ be a scg where $\bar{V} = V \cup \{C, c, d, \hat{d}, S'\} \cup \{\hat{a} \mid a \text{ in } \Sigma\}$ and $P'$ is defined as follows:

(1) (a) If $dca$ is in $L$, $(S') \to (acdc)$ is in $P'$.

(b) $(S') \to (SC)$ is in $P'$.

(2) (a) If $(S) \to (w)$ is in $P$, $(S, C) \to (w, C)$ is in $P'$.

(b) If $(A, B) \to (w_1, w_2)$ is in $P$, $(A, B, C) \to (w_1, w_2, C)$ is in $P'$.

(3) (a) For each $a$ in $\Sigma$, $(\bar{d}, \bar{c}, \bar{a}, C) \rightarrow (\hat{d}, c, \hat{a}, c)$ is in $P'$.

(b) For each $a$ and $b$ in $\Sigma$, $(\hat{d}, \bar{d}, \bar{d}, \hat{a}, \bar{b}) \rightarrow (a, d, \hat{d}, \bar{d}, \hat{b})$ is in $P'$.

(c) For each $a$ and $b$ in $\Sigma$, $(\hat{d}, \bar{d}, \hat{a}, \bar{b}) \rightarrow (a, d, b, d)$ is in $P'$.

(Rules (1a) generate the "small strings", while (1b) starts off the generation. The rules in (2) simulate the generations of $G$ so that $S' \overset{*}{\Rightarrow} wC$ in $P'$ corresponds to $S \overset{*}{\Rightarrow} w$ in $P$. Rule (3a) tags the first $\bar{d}$ and the first $\bar{a}$ after $\bar{c}$ and ensures that only (3b) and (3c) can be applied in the future. If the wrong symbol is tagged, the generation blocks as usual. Rules in (3b) in effect send two "heads" down, permuting every other $d$-symbol with the leftmost unpermuted $\Sigma$ symbol. In a correct generation (3b) is applied:

$$wa_l d\hat{d}\bar{d}\bar{d}x\hat{a}_{l+1}\bar{a}_{l+2}y \Rightarrow wa_l da_{l+1}d\hat{d}x\bar{d}\hat{a}_{l+2}y.$$

Again, if a symbol is "skipped" it can never become terminal and the generation blocks. The generation must end with a rule of (3c) which performs a permutation as it switches the heads off:

$$wa_l d\hat{d}\bar{d}\hat{a}_{l+1}\bar{a}_{l+2}c \Rightarrow wa_l da_{l+1}da_{l+2}dc).$$

It can be shown that

$$\begin{aligned} L_3 = L(\bar{G}) &= \{a_1 d \cdots a_k dca_{k+1} d \cdots a_{2k} dc \mid d^{2k}ca_1 \cdots a_{2k} \text{ in } L\} \\ &\cup \{a_1 d \cdots a_k cda_{k+1} \cdots a_{2k+1} dc \mid d^{2k+1}ca_1 \cdots a_{2k+1} \text{ in } L\} \\ &\cup \{acdc \mid dca \text{ in } L\}. \end{aligned}$$

Then by Lemma 2.3, with $k = 3$, we can "erase" the $c$'s and $d$'s so that $L_1$ is an scl.

We use one more preliminary lemma. Actually the lemma concerns AFL in general rather than just the family of scl.

Lemma 2.7. *If an* AFL *is closed under intersection, it is closed under $\epsilon$-free substitution. If a full* AFL *is closed under intersection, it is closed under substitution.*

*Proof.* Let $\mathscr{L}$ be an AFL closed under intersection. Let $L \subseteq \Sigma_0{}^+$ be in $\mathscr{L}$ and for each $a$ in $\Sigma_0$ let $L_a \subseteq \Sigma_a{}^+$ be in $\mathscr{L}$. Let $\tau$ be the $\epsilon$-free substitution on $\Sigma_0{}^*$ defined by $\tau(a) = L_a$ for each $a$ in $\Sigma_0$. For each $a$ in $\Sigma_0$ let $\bar{a}$ be a new symbol and $\Sigma_1 = \{\bar{a}/a \text{ in } \Sigma_0\} \cup (\bigcup_a \Sigma_a)$. Let $h_1$ be the homomorphism on $\Sigma_1{}^*$ defined by $h_1(\bar{a}) = a$ for each $a$ in $\Sigma_0$ and $h_1(b) = \epsilon$ for each $b$ in $\bigcup_a \Sigma_a$. Now

$$h_1^{-1}(L) = \left\{w_0\bar{a}_1w_1\bar{a}_2w_2 \cdots \bar{a}_nw_n/a_1 \cdots a_n \text{ in } L, w_i \text{ in } \left(\bigcup_a \Sigma_a\right)^*\right\}.$$

Let $L_1 = (\bigcup_a \bar{a}L_a)^+$ and $L_2 = h_1^{-1}(L) \cap L_1$. In [3] it was shown that each AFL contains all $\epsilon$-free regular sets and is closed under restricted homomorphism.

Since by definition $\mathscr{L}$ is closed under inverse homomorphism, intersection, $L_2$ is in $\mathscr{L}$. Let $h_2$ be the homomorphism on $\Sigma_1{}^*$ defined by $h_2(\bar{a}) = \epsilon$ for each $a$ in $\Sigma_0$ and $h_2(b) = b$ for each $b$ in $\bigcup_a \Sigma_a$. Now $\tau(L) = h_2(L_2)$. Observe that $h_2$ is $k$-restricted on $L_2$ and hence $\tau(L)$ is in $\mathscr{L}$. If $L \subseteq \Sigma_0{}^*$, then observe that $\tau(L) = \tau(L - \{\epsilon\}) \cup \{\epsilon\}$ is in $\mathscr{L}$ since $\mathscr{L}$ is closed under $\bigcup\{\epsilon\}$ whenever any set in $\mathscr{L}$ contains $\epsilon$. If $\mathscr{L}$ is a full AFL, then $\mathscr{L}$ is closed under arbitrary homomorphism and $\tau$ need not be $\epsilon$-free.

We now prove another main result.

THEOREM 2.2. *The family of* scl *is closed under intersection, $\epsilon$-free substitution, and linear erasing.*

*Proof.* (1) Intersection.

Let $L_1$ and $L_2$ be scl with $L_1 \cup L_2 \subseteq \Sigma^*$ and let $c$ be a new symbol not in $\Sigma$. Then $L_3 = L_1cL_2 \cap \{wcw/w \text{ in } \Sigma^+\}$ is a scl. Observe that $L_3 = \{wcw/w \text{ in } L_1 \cap L_2\}$. Let $d$ be a new symbol and let $\tau$ be the $\epsilon$-free substituion on $(\Sigma \cup \{c\})^*$ defined by $\tau(a) = \{a, d\}$ for each $a$ in $\Sigma$ and $\tau(c) = c$. Let $L_4 = \tau(L_3) \cap d^+c\Sigma^+$. By Lemma 2.1, $L_4$ is a scl. But $L_4 = \{d^kcw/|\, w\, | = k, w \text{ in } L_1 \cap L_2\}$ and hence by Lemma 2.6, $L_1 \cap L_2$ is a scl.

(2) $\epsilon$-free substitution.

This follows immediately from (1) and Lemma 2.7.

(3) Linear erasing.

Let $L \subseteq \Sigma^*$ be a scl and let $h$ be a homomorphism of $\Sigma^*$ into $\Sigma_1{}^*$ for which there exists a $k$ such that $|\, w\, | \leqslant k\, |\, h(w)|$ for each $w$ in $L$. Without loss of generality we assume there exists a symbol $c$ in $\Sigma$ such that $h(c) = \epsilon$ and $h(a) = a$ for each $a$ in $\Sigma$-$\{c\}$. (Otherwise, let $c$ be a new symbol and $h_1$ the homomorphism of $\Sigma^*$ defined by $h_1(a) = h(a)$ if $h(a) \neq \epsilon$, and $h_1(a) = c$ if $h(a) = \epsilon$. Let $h_2$ be the homomorphism on $(\Sigma_1 \cup \{c\})^*$ defined by $h_2(c) = \epsilon$ and $h_2(a) = a$ for all $a$ in $\Sigma_1$. Clearly $h(L) = h_2(h_1(L))$. Obviously, $h_1(L)$ is a scl and if $k_1 = \max\{|\, h(a)|/a \text{ in } \Sigma\}$ then $|\, h_1(w)| \leqslant k_1\, k\, |\, h_2(h_1(w))|$).

Let $L_1$ be the intersection of the set of all permutations of strings in $L$ with the regular set $(\Sigma\{\epsilon, c,..., c^k\})^*$. Let $h_3(c) = c$ and $h_3(a) = \epsilon$ for each $a$ in $\Sigma$. Let $\bar{c}$ be a new symbol. Now $L_2 = \{w_1\bar{c}w_2/w_i \text{ in } (\Sigma \cup \{c\})^*, h(w_1) = h(w_2{}^r), h_3(w_1) = h_3(w_2)\}$ is clearly a scl since $L_2 = (h^{-1}(\{w\bar{c}w^r \mid w \text{ in } \Sigma^*\})) \cap \{w_1\bar{c}w_2 \mid w_i \text{ in } (\Sigma \cup \{c\})^*\ h_3(w_1) = h_3(w_2)\}$, where both languages in the intersection are context-free. Let $L_3 = (L^r\, \bar{c}L_1) \cap L_2$. By Lemma 2.2, Theorem 2.1 and part (1) above, $L_3$ is a scl. Let $d$ be a new symbol and let $\tau$ be the substitution on $(\Sigma \cup \{c, \bar{c}\})^*$ defined by $\tau(\bar{c}) = \bar{c}$ and $\tau(a) = \{a, d\}$ for each $a$ in $\Sigma \cup \{c\}$. Let $h_4$ be the homomorphism on $(\Sigma \cup \{c, \bar{c}, d\})^*$ defined by $h_4(d) = \epsilon$, $h_4(\bar{c}) = \epsilon$ and $h_4(a) = a$ for each $a$ in $\Sigma \cup \{c\}$. Then

$$h(L) = h(h_4(\tau(L_3) \cap d^*\bar{c}(\Sigma \cup \{c\})^*)).$$

By Lemma 2.1, $\tau(L_3) \cap d^*\bar{c}(\Sigma \cup \{c\})^* \subseteq \{d^lcw/|\, w\, | = l\}$ is a scl and by Lemma 2.6, $h_4(\tau(L_3) \cap d^*\bar{c}(\Sigma \cup \{c\})^*)$ is a scl. Thus by Lemma 2.3, $h(L)$ is a scl.

COROLLARY. *If $L$ is a recursively enumerable set, then there exists a* scl $L'$ *and a homomorphism $h$ such that $h(L') = L$.*

*Proof.* In [4] it was shown that for any recursively enumerable set $L$ there exist $\epsilon$-free deterministic cfl $L_1$ and $L_2$ and a homomorphism $h$ such that $L = h(L_1 \cap L_2)$. But the family of scl contains all $\epsilon$-free cfl and is closed under intersection. Thus $L' = L_1 \cap L_2$ is an scl such that $h(L') = L$.

COROLLARY. The family of scl is not closed under arbitrary homomorphism or quotient by a regular set.

*Proof.* $L_1$ and $L_2$ above can be found so that there exists a regular set $R$ such that $L = h(L_1 \cap L_2) = (L_1 \cap L_2)/R$.

COROLLARY. *The emptiness problem is recursively unsolvable for* scg.

## SECTION 3. RELATION OF SCL TO CSL.

In this section we will show that the family of scl contains all $\epsilon$-free languages that can be accepted by a "quasi-realtime" $n$-tape pda. As a corollary we will show that any language accepted by a nondeterministic Turing machine in linear time is a scl.

DEFINITION. *A quasi-realtime n-pushdown tape* pda (qr $n$-pda for short), $n \geqslant 1$ is an 8-tuple $M = (K, \Sigma, \Gamma, \delta, q_0, F, Z_0, n)$ such that

(1) $K$, $\Sigma$, and $\Gamma$ are finite sets (of *states*, *inputs*, and *tape symbols*, respectively), $q_0$ in $K$ (the *start* state), $Z_0$ in $\Gamma$, $F \subseteq K$ (the set of *final* states) and $n$ a positive integer.

(2) $\delta$ is a mapping of $K \times \Sigma \times \Gamma^n$ into finite subsets of $K \times (\Gamma^* \times \cdots \times \Gamma^*)$.

We now introduce notation to describe the language accepted by a qr $n$-pda. For each $q$ and $q'$ in $K$, $a$ in $\Sigma$, $w$ in $\Sigma^*$, $A_i$ in $\Gamma$, $y_i$ and $z_i$ in $\Gamma^*$, $1 \leqslant i \leqslant n$, we write $(q, aw, y_1A_1, \ldots, y_nA_n) \vdash (q', w, y_1z_1, \ldots, y_nz_n)$ if and only if $(q', z_1, \ldots, z_n)$ is in $\delta(q, a, A_1, \ldots, A_n)$. We let $\overset{*}{\vdash}$ denote the reflexive transitive closure of $\vdash$. The language $L(M)$ accepted by $M$ by final state is

$$\{w \text{ in } \Sigma^* \mid \exists f \text{ in } F, y_1, \ldots, y_n \text{ in } \Gamma^*, (q_0, w, Z_0, \ldots, Z_0) \overset{*}{\vdash} (f, \epsilon, y_1, \ldots, y_n)\}.$$

The language Null $(M)$ accepted by $M$ by empty stack is

$$\{w \text{ in } \Sigma^* \mid \exists q \text{ in } K, (q_0, w, Z_0, \ldots, Z_0) \overset{*}{\vdash} (q, \epsilon, Z_0, \ldots, Z_0)\}.$$

We say that $L$ is a qr $n$-pda language if and only if there exists a qr $n$-pda $M$ such that $L = L(M)$. $\mathscr{L}_n$ denotes the family of all qr $n$-pda languages.

We state the following result without proof.

LEMMA 3.1. (1) *Every* qr $n$-pda *language can be accepted by a one state* qr $n$-pda *by empty stack.*

(2) *For each* $n$, $\mathscr{L}_n$ *is an* AFL.

We now give the following characterization of the quasi-realtime $n$-pda languages.

LEMMA 3.2. *$L$ is in $\mathscr{L}_n$ if and only if there exist $n$ realtime deterministic context-free languages $L_1, \ldots, L_n$ and a length preserving homomorphism $h$ such that*

$$L = h(L_1 \cap \cdots \cap L_n).$$

*Proof.* (if) Since every cfl can be recognized by a qr 1-pda [5, 7], $\mathscr{L}_n$ clearly contains the image of the intersection of $n$ cfl under a $\epsilon$-free homomorphism. (only if) Let $M = (K, \Sigma, \Gamma, \delta, q_1, F, Z_0, n)$ be a qr $n$-pda and let $L = L(M)$ For each $q$ in $K$, $a$ in $\Sigma$ and $n$-tuple $(A_1, \ldots, A_n)$ in $\Gamma \times \cdots \times \Gamma$, order the elements of $\delta(q, a, A_1, \ldots, A_n)$ and let $m(q, a, A_1, \ldots, A_n)$ be the number of elements in $\delta(q, a, A_1, \ldots, A_n)$.

For each $i$, $1 \leqslant i \leqslant m(q, a, A_1, \ldots, A_n)$, let $[q, a, A_1, \ldots, A_n, i]$ be a new symbol and let $\bar{\Sigma}$ be the set of all such symbols. Let $h$ be the length-preserving homomorphism on $\bar{\Sigma}^*$ defined by $h([q, a, A_1, \ldots, A_n, i]) = a$. (Note that each symbol in $\bar{\Sigma}$ contains a state, an input symbol from $\Sigma$, $n$ tape symbols from $\Gamma$, and the number of a move from $M$. We will construct $n$ deterministic pda which accept strings of symbols from $\bar{\Sigma}$. The $j$th pda will simulate the $j$th pushdown store of $M$ and determine if the sequence of encoded moves is legitimate and ends in a final state. If so, the $j$th pda accepts the input). Let $D$ be a new symbol and let $K' = K \cup \{D\}$ For each $j$, $1 \leqslant j \leqslant n$, we define a deterministic pda $M_j = (K', \bar{\Sigma}, \Gamma, \delta_j, \bar{q}_0, F, Z_0)$ by

(1) for each $q$ in $K$ and

$$[q, a, A_1, \ldots, A_n, i] \text{ in } \bar{\Sigma},\ \delta_j(q, [q, a, A_1, \ldots, A_n, i], A_j) = (q', z_j)$$

where the $i$th member of $\delta(q, a, A_1, \ldots, A_n)$ is $(q', z_1, \ldots, z_j, \ldots, z_n)$,

(2) for each $q$ in $K'$, $a$ in $\bar{\Sigma}$ and $Z$ in $\Gamma$ such that $\delta_j(q, a, Z)$ is not defined in [*1*], $\delta_j(q, a, Z) = (D, Z)$.

Let $L_j = L(M_j)$. Let $w = b_1 \cdots b_k$ where for $1 \leqslant l \leqslant k$,

$$b_l = [q_l, a_l, A_{l1}, \ldots, A_{ln}, i_l].$$

Note that $w$ is in $L_j$ if and only if $q_1 = q_0$, $A_{1j} = Z_0$, and there exist $y_{1j}, \ldots, y_{kj}$, $z_{1j}, \ldots, z_{kj}$ in $\Gamma^*$ and $q_{k+1}$ in $F$ such that

$$(q_1, b_1, A_{1j}) \vdash (q_2, b_2, y_{2j}A_{2j}) \vdash \cdots \vdash (q_{k+1}, b_k, y_{kj}A_{kj}) \vdash (q_{k+1}, \epsilon, y_{kj}Z_{kj})$$

where $(q_{l+1}, z_{l+1}, ..., z_{ln})$ is the $i_l$th member of $\delta(q_l, a_l, A_{l1}, ..., A_{ln})$, $1 \leqslant l \leqslant k$, and $y_{l+1,i}A_{l+1,i} = y_{li}z_{li}$, for $1 \leqslant l < k$. Hence $w$ is in $L_1 \cap L_2 \cap \cdots \cap L_n$ if and only if $q_1 = q_0$, for each $j$, $A_{1j} = z_0$ and there exist $y_{1j}, ..., y_{kj}, z_{1j}, ..., z_{kj}$ in $\Gamma^*$, and $q_{k+1} \epsilon F$, such that $(q_{l+1}, z_{l+1}, ..., z_{ln})$ is the $i$th member of $\delta(q_l, a_l, A_{l1}, ..., A_{ln})$, $1 \leqslant l \leqslant k$, and $y_{l+1,i}A_{l+1,i} = y_{li}z_{li}$, for $1 \leqslant l \leqslant k$. But this later condition implies $h(w)$ is in $L(M)$. Further, if $y$ is in $L(M)$, then there exists $w$ in $L_1 \cap \cdots \cap L_n$ such that $y = h(w)$. Therefore $L = L(M) = h(L_1 \cap \cdots \cap L_n)$.

THEOREM 3.1. *If language $L$ is accepted by* qr $n$-pda *then* $L - \{\epsilon\}$ *is a* scl.

*Proof.* Follows immediately from Lemma 3.2 and the fact that the family of scl contains all context-free languages and is closed under intersection and length preserving homomorphism.

COROLLARY. *Any $\epsilon$-free language definable in linear time by a nondeterministic $n$-tape Turing machine is a* scl.

*Proof.* Any such language is a qr 3-pda language [7].

## REFERENCES


1. R. V. BOOK AND S. GREIBACH. Quasi-realtime Languages, (submitted for publication).
2. N. CHOMSKY. "On Certain Formal Properties of Grammars." *Inform. Control* **2**, 137–167 (1959).
3. S. GINSBURG AND S. GREIBACH. Abstract Families of Languages, SDC TM-738/031/00, April (1967).
4. S. GINSBURG, S. GREIBACH, AND M. HARRISON. One-Way Stack Automata, *JACM* 14, No. 2, April 1967, p. 389–418.
5. S. GREIBACH. Inverses of Phrase Structure Generators. Ph. D. Thesis, Harvard Univ. (1963).
6. S. GREIBACH AND J. HOPCROFT. Independence of AFL Operations, SDC TM-738/031/00, July 1967.
7. L. HAINES. Generation and recognition of formal languages. Ph. D. Thesis, Massachusetts Inst. Tech. (1965).