RESTRICTED PARALLELISM AND REGULAR GRAMMARS

RESTRICTED PARALLELISM AND REGULAR GRAMMARS

By

ROBERT DOUGLAS ROSEBRUGH, B.SC.

A Thesis

Submitted to the School of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Master of Science

McMaster University

November 1972

MASTER OF SCIENCE (1972) (Mathematics)

McMASTER UNIVERSITY Hamilton, Ontario.

TITLE: Restricted Parallelism and Regular Grammars

AUTHOR: Robert Douglas Rosebrugh, B.Sc. (McMaster University)

SUPERVISOR: Dr. Derick Wood

NUMBER OF PAGES: vi, 89

SCOPE AND CONTENTS:

This thesis studies the properties of k-parallel right-linear languages. An infinite hierachy of language families is found and closure properties of these families are studied. The language families are characterised in terms of simple languages and non-deterministic generalised sequential machine mappings. In addition a characterisation of k-right-linear simple matrix languages by k-parallel right-linear languages with a control device is given.

ACKNOWLEDGEMENTS

The author wishes to express his deep gratitude to his supervisor, Dr. Derick Wood, whose encouragement and criticism were of great value in the preparation of this thesis. Thanks are also due to Dr. Arto Salomaa of the University ot Turku, Finland, whose enlightening presentation of a graduate course on Formal Languages was most helpful.

The author also wishes to acknowledge the financial support of the National Research Council and express his appreciation to Ms. Carolyn Sheeler for her prompt and efficient typing of the manuscript.

TABLE OF CONTENTS

Chapter 1:	Introduction	1
§1.	Language and Grammar	1
§2.	Accepters and Machines	13
Chapter 2:	k-Parallel Right Linear Languages	18
§1.	Introduction	18
§2.	The Infinite Hierarchy and Related Results	24
§3.	ε-Rules and Factor Languages	39
§4.	Closure Properties	47
§5.	k-Parallel Left Linear Languages	59
§6.	Decidability Questions	64
Chapter 3:	Regulated Rewriting	68
§1.	k-Parallel Right-Linear with Regular Control Languages	68
§2.	Right-Linear Tuple Languages	74
§3.	Right-Linear Simple Matrix Languages	79
§4.	Another Restriction on Derivations	86
Bibliography		88

PREFACE

In recent years several studies have been made of phrase-structure grammars with rewriting methods which are "parallel" in that more than one rewriting rule is applied at each derivation step. This parallelism greatly increases the generative capacity of context-free productions in the case of scattered context languages as defined by Greibach and Hopcroft [2], and simple matrix languages, tuple languages and equal matrix languages as defined by Ibarra [4], Kuich and Mauer [5], and Siromoney [10] respectively. The absolutely parallel grammars of Rajlich [6] generate a smaller class of languages than the context-free languages. Rozenberg and Doucet [7] have studied 0-L systems which employ parallel rewriting without terminals.

This thesis arose from the notion of placing a "k - at a time" restriction on 0-L systems. In the present form it is more closely related to [4], [5] and [10].

Chapter 1 gives preliminary definitions and states some well-known results from Language Theory.

Proofs of these may be found in Hopcroft and Ullman [3], or in Salomaa [9].

In Chapter 2 we define k-parallel right linear grammars and study the properties of the families $\mathcal{L}_k$ which are generated by them. In §2 we show that the families $\mathcal{L}_k$ form a proper infinite hierarchy of language families. In §4 we consider closure properties of these families and give a characterisation of each by a simple language and non-deterministic generalised sequential machine mappings. In §5 we consider k-parallel left-linear languages and in §6 the decision properties of the families $\mathcal{L}_k$.

Chapter 3 is devoted to giving a new characterisation of k-right-linear simple matrix languages by k-parallel right-linear languages with a control device.

As far as the author knows, the families $\mathcal{L}_k$ are new, so all of Chapter 2 is original, although some of the proofs are standard. Except for Theorem 3.5 which was pointed out by Seymour Ginsburg [1], Chapter 3 is also new material.

CHAPTER 1

INTRODUCTION

§1. LANGUAGE AND GRAMMAR.

A non-empty finite set is called an alphabet or vocabulary. Elements of an alphabet are called letters or symbols. If V is an alphabet we denote by V^* the free monoid generated by V. Elements of V^* are called words or strings of symbols. The operation in V^* is called catenation and is denoted by juxtaposition i.e. if $x, y \in V^*$, their product is written xy. The neutral element of V^* (which is the string with no symbols) is called the empty word and is denoted by ε. We denote by V^+ the set $V^* - \{\varepsilon\}$. If $x, y \in V^*$, then y is a subword of x if there exist $z, w \in V^*$ such that $x = zyw$; if $z = \varepsilon$ then y is an initial subword, and if $w = \varepsilon$ then y is a final subword. If $x \in V^*$ then the mirror image of x, denoted $mi(x)$, is the element of V^* obtained by writing x backwards e.g. if $V = \{a, b\}$ and $x = abab$, then $mi(x) = baba$. By convention $mi(\varepsilon) = \varepsilon$.

We define a <u>length function</u> $|-|: V^* \to \mathbb{N} \cup \{0\}$ by (i) $|\varepsilon| = 0$, $|a| = 1$ for all $a \in V$

(ii) $|xy| = |x| + |y|$ for all $x, y \in V^*$.

Intuitively, the length of a word is just the number of symbols occurring in it.

Let V be an alphabet. A <u>language over V</u> is a subset of V^*. A <u>family of languages</u> is a pair $(\Sigma, \mathcal{L})$ where $\#(\Sigma) = \infty$ and $\mathcal{L}$ is a family of subsets of Σ^* satisfying

(i) there exists $L \in \mathcal{L}$ such that $L \neq \phi$.

(ii) for all $L \in \mathcal{L}$ there exists $\Sigma_L \subseteq \Sigma$ with $\#(\Sigma_L) < \infty$ and $L \subseteq \Sigma_L^*$.

In the sequel we will speak of a family of languages without mentioning the first component of the pair.

Given a family of languages $\mathcal{L}$ it is natural to ask if $\mathcal{L}$ is <u>closed</u> under <u>operations</u> which can be defined on $\mathcal{L}$. For example, since the members of are sets, we can ask if, given $L_1, L_2 \in \mathcal{L}$, whether $L_1 \cup L_2$, $L_1 \cap L_2$ and $L_1 - L_2$ are in $\mathcal{L}$. We now define several language-theoretic operations:

(1) the <u>catenation</u> (or <u>product</u>) of two languages L_1 and L_2 is defined by $L_1L_2 = \{xy \mid x \in L_1 \text{ and } y \in L_2\}$

(2) for a language L we define L^i, $i \geq 1$ to be the language obtained by catenating i copies of L (catenation is associative!), and $L^0 = \{\varepsilon\}$. The <u>catenation closure</u> of L is $L^* = \bigcup_{i=0}^{\infty} L^i$.

(3) the <u>left quotient</u> of a language L_1 by a language L_2 is defined by $L_2 \backslash L_1 = \{x \mid yx \in L_1$ for some $y \in L_2\}$. The <u>right quotient</u> is similarly defined: $L_1/L_2 = \{x \mid xy \in L_1$ for some $y \in L_2\}$.

(4) the <u>mirror image</u> of a language L is the collection of mirror images of its words i.e. $mi(L) = \{mi(x) \mid x \in L\}$

(5) let V be an alphabet and for each $a \in V$, let V_a be an alphabet. Let $\sigma(a)$ be a language over V_a for each $a \in V$. Define $\sigma(\varepsilon) = \{\varepsilon\}$ and $\sigma(xy) = \sigma(x)\sigma(y)$ for $x, y \in V^*$. Letting $\overline{V} = \bigcup_{a \in V} V_a$, σ defines a mapping of V^* into $2^{\overline{V}^*}$ which is called a <u>substitution</u>. For a language L over V we define $\sigma(L) = \{x \mid x \in \sigma(y)$ for some $y \in L\}$. A family of languages $\mathcal{L}$ is closed under substitution if whenever $L \in \mathcal{L}$ is a language over V and σ is a substitution such that $\sigma(a) \in \mathcal{L}$ for all $a \in V$ then $\sigma(L) \in \mathcal{L}$.

(6) a substitution such that $\#(\sigma(a)) = 1$ for all $a \in V$ is called a homomorphism. (Thus a homomorphism maps V^* into $\bar{V}^*$ and is a homomorphism of free monoids.)

We will define other closure operations below. We now define the four basic types of phrase-structure grammars and the associated language families.

DEFINITION 1.1: A generative grammar (of Type 0) is an ordered quadruple $G = (N, T, S, P)$ where N and T are disjoint alphabets, $S \in N$ and P is a finite set of pairs (u, v) such that $u \in (N \cup T)^*N(N \cup T)^*$ and $v \in (N \cup T)^*$.

Elements of N are called non-terminals, elements of T are called terminals and S is called the sentence symbol. Elements (u, v) of P are called rewriting rules or productions and are written $u \to v$.

DEFINITION 1.2: Let $G = (N, T, S, P)$ be a generative grammar. We define a binary relation $\underset{G}{\Rightarrow}$ ("yields") on $(N \cup T)^*$ by $x \underset{G}{\Rightarrow} y$ iff there exist $x_1, x_2, u, v \in (N \cup T)^*$ such that $x = x_1ux_2$, $y = x_1vx_2$ and $u \to v \in P$. We denote by $\overset{*}{\underset{G}{\Rightarrow}}$ $(\overset{+}{\underset{G}{\Rightarrow}})$ the reflexive, transitive closure (transitive closure) of $\underset{G}{\Rightarrow}$ i.e.

$x \overset{*}{\underset{G}{\Longrightarrow}} y$ iff either (1) $x = y$ or (2) there exist $x_0, x_1, \ldots, x_n$ such that $x = x_0$, $y = x_n$ and $x_{i-1} \underset{G}{\Longrightarrow} x_i$ for $1 \leq i \leq n$ ($x \overset{+}{\underset{G}{\Longrightarrow}} y$ iff 2 holds).

When no confusion can arise we will write simply $\Longrightarrow$ ($\overset{*}{\Longrightarrow}$, $\overset{+}{\Longrightarrow}$) instead of $\underset{G}{\Longrightarrow}$ ($\overset{*}{\underset{G}{\Longrightarrow}}$, $\overset{+}{\underset{G}{\Longrightarrow}}$). We note that later in this chapter, and especially in Chapters 2 and 3, the symbol $\Longrightarrow$ will have different meanings as different types of grammars are defined. The distinctions should be clear from the context.

A derivation by G, where $G = (N, T, S, P)$ is a generative grammar is a finite sequence $D: Q_0, Q_1, \ldots, Q_n$ $(n \geq 0)$ satisfying $Q_{i-1} \underset{G}{\Longrightarrow} Q_i$ $1 \leq i \leq n$.

DEFINITION 1.3: Let $G = (N, T, S, P)$ be a generative grammar. The language generated by G is $L(G) = \{x \in T^* \mid S \overset{*}{\underset{G}{\Longrightarrow}} x\}$.

Again, as several types of grammars are introduced below, the notation $L(G)$ will take on several meanings, but its meaning will always be clear from the context. We say two generative grammars G_1 and G_2 are equivalent if $L(G_1) = L(G_2)$. We denote by $\mathcal{L}_{RE}$ the family of languages generated by generative grammars of Type 0 and state

<u>THEOREM 1.1</u>: $\mathcal{L}_{RE}$ equals the family of recursively enumerable sets.

<u>DEFINITION 1.4</u>: A generative grammar $G = (N, T, S, P)$ is <u>context-sensitive</u> (or <u>Type 1</u>) iff each production in P is of the form $x_1Xx_2 \to x_1yx_2$ where $X \in N$, $x_1, x_2, y \in (N \cup T)^*$ and $y \neq \varepsilon$ with the possible exception of the production $S \to \varepsilon$ whose occurrence in P implies that S does not occur on the right side of any production in P.

A language L is <u>context-sensitive</u> if there exists a context sensitive grammar (csg) G such that $L = L(G)$. We denote the family of context-sensitive languages by $\mathcal{L}_{cs} = \{L \mid L = L(G)$ for some csg $G\}$.

<u>DEFINITION 1.5</u>: A <u>context-free grammar</u> (or <u>Type 2</u> grammar) is a generative grammar $G = (N, T, S, P)$ such that for each production $u \to v \in P$ we have $u \in N$.

A language L is a <u>context-free language</u> <u>(cfl)</u> if there exists a cfg G such that $L = L(G)$. We denote the family of context-free languages by $\mathcal{L}_{CF}$.

Since the application of a rewriting rule in a a derivation by a context-free grammar depends only on one non-terminal (independent of context - hence the name) we can assign a <u>derivation tree</u> to a derivation by a cfg. A <u>tree</u> is a directed graph satisfying

(1) there is exactly one node (the <u>vertex</u>) which no edge enters.

(2) there is exactly one path from the vertex to each other node.

A derivation by a cfg is <u>leftmost</u> if at each step the leftmost non-terminal is the one replaced. It is easy to show that every word in the language generated by a cfg has a leftmost derivation. To a leftmost derivation by a cfg it is possible to assign a unique "derivation tree". We give an example to illustrate this process.

<u>EXAMPLE 1.1</u>: Let $G = (\{S, X, Y\}, \{a, b\}, S, P)$ where P contains:

$S \to XY$

$X \to XX | aY | a$ (we use this notation as an abbreviation for $X \to XX$, $X \to aY$, $X \to a$)

$Y \to b$

G is clearly a cfg. Some sample leftmost derivations by G are:

(1) S==>XY==>aY==>ab

(2) S==>XY==>XXY==>aXY==>aaYY==>aabY==>aabb.

The tree associated with (1) is

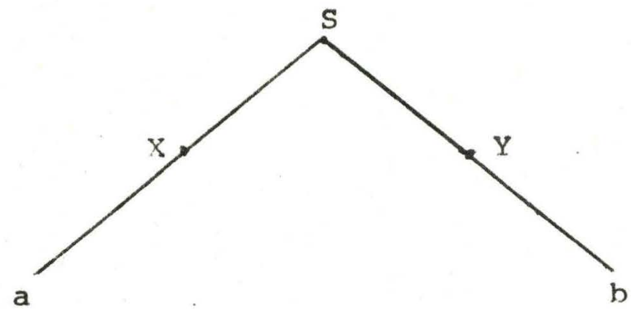


FIGURE 1.1

The tree associated with (2) is:

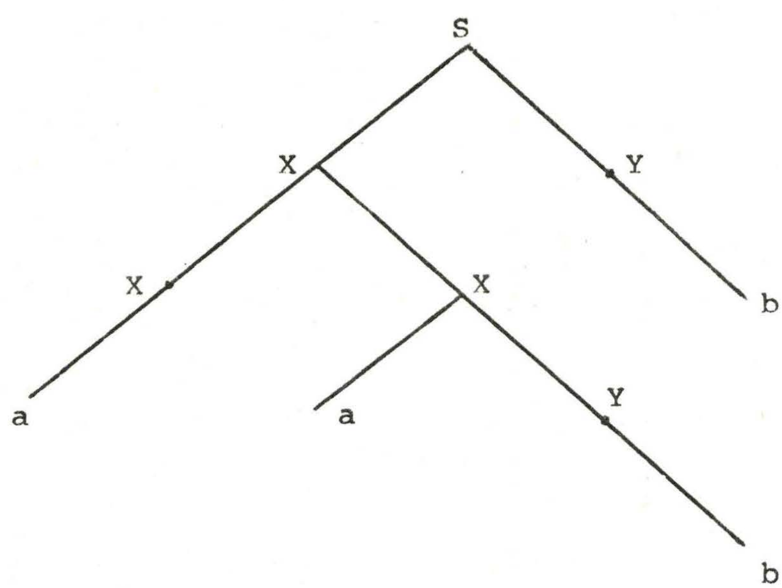


FIGURE 1.2

Note that the "leaves" (the nodes from which no edges emanate) are labelled by terminals, all other nodes are labelled by non-terminals, and the vertex is always labelled by S. The word generated can be read from the leaves from left to right.

DEFINITION 1.6: A *right-linear grammar (rlg) (regular grammar, Type 3 grammar)* is a context-free grammar $G = (N, T, S, P)$ such that if $X \to x \in P$ then $x \in T^*N \cup T^*$. A language L is a *regular language* (*regular set*, *finite-state language*) if there exists an rlg G such that $L = L(G)$. We denote the family of regular languages by $\mathcal{L}_{REG}$. Regular languages have been characterised in many ways. We give one which will introduce useful notation for the sequel.

DEFINITION 1.7: Let T and $V = \{\cup, *, \phi, (,)\}$ be disjoint alphabets. A word over $T \cup V$ is a *regular expression* over T if

(1) $x \in V$ or $x = \phi$, or

(2) x is one of the forms $(y \cup z)$, (yz) or y^* where y and z are regular expressions over T.

Each regular expression x over T denotes a language $\ell(x)$ according to the following conventions:

(1) the language denoted by ϕ is the empty language.

(2) the language denoted by $a \in T$ is $\{a\}$.

(3) for regular expressions x and y over T,

$\ell(x \cup y) = \ell(x) \cup \ell(y)$, $\ell((xy)) = \ell(x)\ell(y)$, $\ell(x^*) = \ell(x)^*$.

It is well known that a language is denoted by a regular expression iff it is regular.

A cfg $G = (N, T, S, P)$ is left-linear if $X \to x \in P$ implies $x \in NT^* \cup T^*$. The family of languages generated by left-linear grammars is $\mathcal{L}_{REG}$. Given a rlg $G = (N, T, S, P)$ we say that a non-terminal Y is reachable from a non-terminal X if there exists a derivation by G $D: X = Q_0 \Longrightarrow Q_1 \Longrightarrow \ldots \Longrightarrow Q_n = yY$ where $n \geq 1$ and $y \in T^*$.

We use the notion of regular set to define a new closure operation. A family of languages $\mathcal{L}$ is closed under intersection with a regular set if whenever $L \in \mathcal{L}$ and $R \in \mathcal{L}_{REG}$ then $L \cap R \in \mathcal{L}$.

The four families of languages we have defined are called the Chomsky Hierarchy and play a fundamental role in language theory. They are linked by:

THEOREM 1.2: $\mathcal{L}_{REG} \subsetneq \mathcal{L}_{CF} \neq \mathcal{L}_{CS} \neq \mathcal{L}_{RE}$.

The language families of the Chomsky Hierarchy are obtained by restricting the form of productions. It is also possible to restrict the manner of generation allowed. Several types of 'regulated rewriting' have been defined. We now introduce a type of restricted derivation which will be used in the sequel.

Consider a grammar $G = (N, T, S, P)$ with production set P. A labelling of productions is a one-one correspondence $Lab: P \longrightarrow Lab(P)$ where $Lab(P)$ is an alphabet. To each derivation by G there corresponds a control word over $Lab(P)$ consisting of the labels of productions applied in D in the order of their application. The language generated by G with control language C is the subset of $L(G)$ which consists of words having a derivation with a control word in C. We denote $L(G, C) = \{x \in T^* \mid \exists$ a derivation $D: S \overset{*}{\Longrightarrow} x$ and $u \in C$ such that u is a control word of $D\}$.

The study of grammars with control languages has been mainly restricted to the case where C is a regular language. It can be shown that if G is of Type 0, context-sensitive or regular then $L(G, C)$ (where C is a regular set) is also Type 0, context-sensitive or regular respectively. We shall have occasion to use the last case in Theorem 2.12. In fact the most interesting case of grammars with control languages are context-free grammars, for in this case the generative capacity is greatly increased by the addition of a control language.

§2. ACCEPTERS AND MACHINES.

In this section we define the language accepting and translating devices which we will use in Chapters 2 and 3.

DEFINITION 1.8: A finite state accepter (fsa) is an 5-tuple $M = (Q, \Sigma, \delta, q_0, Q_F)$ where Q and Σ are finite non-empty sets, $\delta: Q \times \Sigma \longrightarrow Q$, $q_0 \in Q$ and $Q_F \subseteq Q$.

We call Q the set of states, Σ the input alphabet, δ the transition function, q_0 the initial state and Q_F the final states. We can extend δ to δ^* defined on $Q \times \Sigma^*$ by

(i) $\delta^*(q, \varepsilon) = q$ for all $q \in Q$

(ii) $\delta^*(q, x) = \delta^*(\delta^*(q, y), a)$ for all $q \in Q$ where $ya = x \in \Sigma^+$ and $a \in \Sigma$.

The language accepted by M is $T(M) = \{x \in \Sigma^* \mid \delta^*(q_0, x) \in Q_F\}$. We denote the family of languages accepted by fsa's by $\mathcal{L}_{fsa}$.

DEFINITION 1.9: A nondeterministic finite state accepter (nfsa) is a 5-tuple $M = (Q, \Sigma, \delta, q_0, Q_F)$ where Q, Σ, q_0 and Q_F are as in Definition 1.8 and $\delta: Q \times \Sigma \longrightarrow 2^Q$.

Again, δ can be extended to δ^* defined on $Q \times \Sigma^*$ and the language accepted by an nfsa M is defined by $T(M) = \{x \in T^* | \delta^*(q_0, x) \cap Q_F \neq \phi\}$. The family of languages accepted by nfsa's is denoted by $\mathcal{L}_{nfsa}$. The result linking these two language families to the Chomsky Hierarchy is

<u>THEOREM 1.3</u>: $\mathcal{L}_{fsa} = \mathcal{L}_{nfsa} = \mathcal{L}_{REG}$.

If L is a language over T, we denote $F(L) = \{a \in T | ax \in L$ for some $x \in T^*\}$. If L is a regular language specified by an fsa, regular grammar or regular expression, there is an algorithm to find $F(L)$.

In §2 of Chapter 2 we will give a generalisation of the following well-known theorem on regular sets.

<u>THEOREM 1.4</u>: <u>(Iterating Factor Theorem)</u> Let L be a regular set. There exist natural numbers p and q such that if $x \in L$ and $|x| > p$ then $x = uvw$ with $q > |v| > 0$ and for all $i \geq 0$, $uv^iw \in L$.

<u>DEFINITION 1.10</u>: A <u>nondeterministic generalised sequential machine (ngsm)</u> is an ordered 6-tuple $S = (Q, \Sigma, \Delta, \delta, q_0, Q_F)$ where Q, Σ and Δ are alphabets, $q_0 \in Q$, $Q_F \subseteq Q$ and $\delta: Q \times \Sigma \longrightarrow 2^{Q \times \Delta^*}$ (finite subsets only).

We call Q the set of states, Σ the input alphabet, Δ the <u>output alphabet</u>, δ the transition function, q_0 the initial state and Q_F the final states. As for fsa's and nfsa's, δ can be extended to $Q \times \Sigma^*$. For $x \in \Sigma^*$ we denote

$S(x) = \{y \in \Delta^* \mid (q, y) \in \delta^*(q_0, x)$ for some $q \in Q_F\}$.

If L is a language over Σ we denote

$S(L) = \{y \in \Delta^* \mid y \in S(x)$ for some $x \in L\}$. We call $S(L)$ an <u>ngsm mapping</u>. If $\mathcal{L}$ is a family of languages, then $\mathcal{L}$ is <u>closed under ngsm mappings</u> if whenever $L \in \mathcal{L}$ and S is an ngsm, then $S(L) \in \mathcal{L}$.

We also note that an ngsm can also be defined as a 7-tuple $S = (Q, \Sigma, \Delta, \delta, \lambda, q_0, Q_F)$ where Q, Σ, Δ, q_0 and Q_F are as above, and $\delta: Q \times \Sigma \longrightarrow Q$ and $\lambda: Q \times \Sigma \longrightarrow 2^{\Delta^*}$ (finite subsets only). In this notation $S(x) = \{y \in \Delta^* \mid y \in \lambda^*(q_0, x)\}$ and $S(L)$ is as above. We shall use whichever formalism is more convenient in the sequel.

<u>DEFINITION 1.11</u>: A <u>push-down accepter (pda)</u> is an ordered 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, Q_F)$ where Q, Σ and Γ are alphabets, $q_0 \in Q$, $Z_0 \in \Gamma$, $Q_F \subseteq Q$ and $\delta: Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \longrightarrow 2^{Q \times \Gamma^*}$ (finite subsets only).

We call Q the set of states, Σ the input alphabet, Γ the set of pushdown symbols, δ the transition function, q_0 the initial state, Z_0 the bottom of pushdown marker, and Q_F the final states.

A configuration of a pda M is a pair (q, γ) where $q \in Q$ and $\gamma \in \Gamma^*$. If $a \in (\Sigma \cup \{\varepsilon\})$, $\gamma, \gamma' \in \Gamma^*$, $Z \in \Gamma$, and $(q', \gamma') \in \delta(q, a, Z)$ then we write $a\colon (q, Z_\gamma) \vdash (q', \gamma'\gamma)$. We can extend this notation in the obvious way to cover strings of symbols over $\Sigma \cup \{\varepsilon\}$ and we then write $x\colon (q,\gamma) \overset{*}{\vdash} (q', \gamma')$ for x a word over $\Sigma \cup \{\varepsilon\}$, $q, q' \in Q$ and $\gamma, \gamma' \in \Gamma^*$.

For a pda M we defined the language accepted by final state to be $T(M) = \{x \in \Sigma^* | x\colon (q_0, Z_0) \overset{*}{\vdash} (q, \gamma)$ for some $q \in Q_F$, $\gamma \in \Gamma^*\}$. We denote by $\mathcal{L}_{pda}$ the family of languages accepted by final state by a pda.

For a pda M, we define the language accepted by empty store to be $N(M) = \{x \in \Sigma^* | x\colon (q_0, Z_0) \overset{*}{\vdash} (q, \varepsilon)$ for some $q \in Q\}$. We denote by $\mathcal{L}_{es}$ the family of languages accepted by empty store by a pda. We have the following theorem linking these two types of acceptance by pda's and the Chomsky Hierarchy.

THEOREM 1.5: $\mathcal{L}_{pda} = \mathcal{L}_{es} = \mathcal{L}_{CF}$.

Before we go on to the study of parallelism and regular grammars we make one more remark. The proofs in Chapters 2 and 3 involve many constructions of new grammars and machines from given ones. In these constructions new symbols are added to given alphabets, and new symbols are constructed from old ones. We make the convention that any new symbols introduced are really new symbols i.e. they do not occur in any alphabet already given. In addition, abstract symbols will often be introduced as pairs of members from given alphabets. We use square brackets instead of round brackets for convenience of notation and to aid the reader e.g. given alphabets X and Y we form the new alphabet $X \times Y = \{[x, y] \mid x \in X, y \in Y\}$.

CHAPTER 2

k-PARALLEL RIGHT LINEAR LANGUAGES

§1. INTRODUCTION

In this chapter we introduce the notion of k-parallel right-linear grammar and study the families of languages generated by them. These grammars differ from conventional phrase-structure grammars in that k productions are applied at each derivation step with a resulting increase in generative capacity.

DEFINITION 2.1: For $k \in \mathbb{N}$, a k-parallel right-linear grammar (k-rlg) is a 5-tuple $G = (N, T, S, P, k)$ where

(1) (N, T, S, P) is a context-free grammar

(2) $S \to x \in P$ implies $x \in N^k \cup T^*$

(3) $X \to x \in P$ and $X \neq S$ implies $x \in T^*N \cup T^+$

(4) $X \to x \in P$ implies $x \neq ySz$ for all $y, z \in (N \cup T)^*$.

Points (2) and (4) of the definition mean that productions from S generate k non-terminals or a terminal word and that S can never appear on the right side of a production. Point (3) means that all other rules are right-linear rules.

<u>DEFINITION 2.2</u>: Let $G = (N, T, S, P, k)$ be a k-parallel right linear grammar. The yield relation $\underset{G}{\Longrightarrow}$ is defined on $(N \cup T)^* \times (N \cup T)^*$ by $x \underset{G}{\Longrightarrow} y$ if (1) $x = S$ and $S \to y \in P$ or (2) $x = x_1 X_1 x_2 \ldots x_k X_k$ and $y = x_1 z_1 x_2 \ldots x_k z_k$, and $X_i \to z_i \in P$ for $i = 1, 2, \ldots, k$. $\overset{*}{\underset{G}{\Longrightarrow}}$ ($\overset{+}{\underset{G}{\Longrightarrow}}$) is the reflexive, transitive closure (transitive closure) of $\underset{G}{\Longrightarrow}$

When no confusion can arise we will write $\underset{G}{\Longrightarrow}$ simply as $\Longrightarrow$.

<u>DEFINITION 2.3</u>. A language $L \subseteq T^*$ is called a <u>k-parallel right-linear language (k-rll)</u> iff there exists a k-rlg $G = (N, T, S, P, k)$ such that $L = L(G) = \{x \in T^* | S \overset{*}{\underset{G}{\Longrightarrow}} x\}$. We denote $\mathcal{L}_k = \{L | L$ is a k-rll$\}$ and $\mathcal{L} = \bigcup_{k=1}^{\infty} \mathcal{L}_k$.

<u>EXAMPLE 2.1</u>: Consider the 3-rlg $G_3 = (\{S, X, Y, Z\}, \{a, b, c\}, S, P, 3)$ where P contains:

$S \to XYZ$

$X \to aX | a$

$Y \to bY | b$

$Z \to cZ | c$.

Some examples of derivations by G_3 are:

$$S \Longrightarrow XYZ \Longrightarrow abc$$

$$S \Longrightarrow XYZ \Longrightarrow aXbYcZ \Longrightarrow a^2b^2c^2$$

$$S \Longrightarrow XYZ \Longrightarrow aXbYcZ \Longrightarrow a^2Xb^2Yc^2Z \Longrightarrow a^3b^3c^3.$$

From these it should be evident (and it is easy to show by induction) that $L(G_3) = \{a^nb^nc^n | n \geq 1\}$. This language is context-sensitive, but it is not context-free.

EXAMPLE 2.2: Consider the 2-rlg $G = (\{S, X, Y, W, Z\}, \{a, b, c, d\}, S, P, 2)$ where P contains:

$$S \rightarrow XY$$

$$X \rightarrow aX | X | Z | a$$

$$Z \rightarrow bZ | b$$

$$Y \rightarrow cY | W$$

$$W \rightarrow dW | d.$$

Some sample derivations by G are:

$$S \Longrightarrow XY \Longrightarrow aXcY \Longrightarrow aZcW \Longrightarrow abcd$$

$$S \Longrightarrow XY \Longrightarrow aXcY \Longrightarrow a^2XcW \Longrightarrow a^3cd$$

$$S \Longrightarrow XY \Longrightarrow aXW \Longrightarrow aZdW \Longrightarrow abd^2.$$

Again an induction shows that

$$L(G) = \{xy | x \in a^*bb^* \cup aa^*, y \in c^*dd^* \text{ and } |x| \leq |y| \text{ if } x \in a^*bb^*, |x| \leq |y| + 1 \text{ if } x \in aa^*\}.$$

L(G) is context-free (as are all 2-rll's, see Lemma 2.7) but is clearly not regular.

LEMMA 2.1: The family of regular languages equals $\mathcal{L}_1$.

PROOF: First, it is clear that any language in $\mathcal{L}_1$ is regular since it is generated by a context-free grammar with only right-linear productions. Next let $L \subseteq T^*$ be a regular language and $G = (N, T, S, P)$ be a right-linear grammar for L. If P contains no productions of the form $X \to \varepsilon$ then $G_1 = (N \cup \{S'\}, T, S', P \cup \{S' \to S\}, 1)$ is a 1-rlg for L. Otherwise[1] we construct $G_2 = (N_2, T, S', P_2, 1)$ where $N_2 = N \cup \{S'\} \cup \{X_a | X \in N, a \in T\}$. For each $X \in N$ let L(X) be the regular language generated by $G_X = (N, T, X, P)$ and recall that we can decide if $\varepsilon \in L(X)$ or not. P_2 contains:

(1) $S' \to S$, and $S' \to \varepsilon$ if $\varepsilon \in L$

(2) $X \to yY$ if $X \to yY \in P$, $y \in T^*$, $X, Y \in N$ and $\varepsilon \notin L(Y)$

---

1. It is well known that every regular set can be generated by a right-linear grammar without ε-rules. We give a construction to show this fact in order to introduce the notion L(X), for $X \in N$ (see Theorem 2.3 and Theorem 2.12) and to give an example of a type of construction used in Theorem 2.11 and Theorem 3.3.

(3) $X \to yY_a$ if $X \to yaY \in P$, $y \in T^*$, $a \in T$, $X, Y \in N$ and $\varepsilon \in L(Y)$

(4) $X_a \to Y_a$ if $X \to Y \in P$, $X, Y \in N$, for all $a \in T$

(5) $X_a \to ayY_b$ if $X \to ybY \in P$, $y \in T^*$, $b \in T$, $X, Y, \in N$ and $\varepsilon \in L(Y)$, for all $a \in T$

(6) $X_a \to ayY$ if $X \to yY \in P$, $y \in T^+$, $X, Y \in N$ and $\varepsilon \notin L(Y)$, for all $a \in T$

(7) $X_a \to ay$ if $X \to y \in P$, $y \in T^*$, $X \in N$ for all $a \in T$

(8) $X \to y$ if $X \to y \in P$, $y \in T^+$, $X \in N$.

G_2 imitates derivations by G. When G_2 detects that the corresponding G-derivation may end without further deposit of terminals, it attaches the last symbol which is deposited by G to its non-terminal (point 3) as a subscript. This terminal is carried along (4) and is deposited when the ε-rule would be applied in the corresponding G-derivation (7), or, if a non-trivial word could also be generated, before the next deposit of terminals takes place in the corresponding G-derivation (5 and 6). Note that the new sentence symbol guarantees that initial productions will be of the correct form for G_2 to be a 1-rlg. Now $L = L(G_2)$ so each regular set is in $\mathcal{L}_1$ and we are done.

LEMMA 2.2: Given a word $x \in T^*$, there is an algorithm to decide if $x \in L(G)$ where $G = (N, T, S, P, k)$ is a k-rlg.

PROOF: By the definition of k-rlg it is obvious that $\varepsilon \in L(G)$ iff the production $S \to \varepsilon$ is in P. Thus we may assume $x \neq \varepsilon$ and consider sequences of the form

(1) $$S = y_0, y_1, \ldots, y_{n-1}, y_n = x$$

where $n \geq 1$, y_i are pairwise distinct words over $N \cup T$ and for $0 \leq i \leq n-1$ we have $|y_i| \leq |y_{i+1}|$. Clearly the number of such sequences is finite. Moreover $x \in L(G)$ iff for some sequence (1) we have

(2) $$S = y_0 \underset{G}{\Longrightarrow} y_1 \underset{G}{\Longrightarrow} \ldots \underset{G}{\Longrightarrow} y_{n-1} \underset{G}{\Longrightarrow} y_n = x.$$

Thus it suffices to check, for each of the finitely many sequences (1), whether or not (2) is satisfied. This can be done since for any two words z_1 and z_2 and a k-rlg G we can decide by checking through productions of G whether or not $z_1 \underset{G}{\Longrightarrow} z_2$ holds.

The essential point in the proof is that we may assume $|y_i| \leq |y_{i+1}|$ for $0 \leq i \leq n-1$ since k-rlg's are 'length-increasing'. Lemma 2.2 means that the 'Membership Problem' is decidable for k-rlg's. We will make extensive use of this fact. Other decision problems are considered in §6.

§2. THE INFINITE HIERARCHY AND RELATED RESULTS

In this section we show that the families $\mathcal{L}_k$ form an infinite proper hierarchy of language families and present results relating these language families to the Chomsky Hierarchy.

THEOREM 2.3. For all $k \geq 1$, $\mathcal{L}_k \subsetneq \mathcal{L}_{k+1}$.

REMARKS: The proof of this theorem is quite involved, but the idea is simple: a derivation by a k-rlg is mimicked by a derivation of a constructed k+1-rlg which uses one of its "slots" to deposit only one letter of the word in question. We recall that for a regular language $L \subseteq T^*$ the set $F(L) = \{a \in T \mid \exists\, x \in T^* \text{ with } ax \in L\}$ can be found effectively.

PROOF: Let $L \in \mathcal{L}_k$ and $G = (N, T, S, P, k)$ a k-rlg such that $L = L(G)$. We construct a k+1-rlg $G' = (N', T, S, P', k+1)$ where $N' = N \cup \{[X, a] \mid X \in N, a \in T\} \cup \{Y_a \mid a \in T\}$. Let $m = \max\{|x| \mid X \to x \in P\}$. P' contains:

(1) $S \to X_1 \ldots X_{i-1} Y_a [X_i, a] X_{i+1} \ldots X_k$ if $S \to X_1 \ldots X_k \in P$, $X_i \in N, 1 \leq i \leq k$, and $a \in F(L(X_i))$.

(2) $S \to x$ if $S \to x \in P$ and $x \in T^*$.

(3) $S \to x$ if $x \in L(G)$ and $|x| \leq km$ (this step is okay by Lemma 2.2).

(4) $Y_a \to Y_a | a$ for all $a \in T$.

(5) $X \to yY$ if $X \to yY \in P$, $X, Y \in N$, $y \in T^*$.

(6) $X \to x$ if $X \to x \in P$, $X \in N$, $x \in T^*$.

(7) $[X, a] \to [Y, a]$ if $X \to Y \in P$, $a \in T$ and X is reachable by a sequence of chain rules from a non-terminal occurring on the right side of an initial production.

(8) $[X, a] \to yY$ if $X \to ayY \in P$, $X, Y \in N$, $y \in T^*$, $a \in T$.

G' is clearly a k+1-rlg. We now give a description of the operation of G':

(i) all words in L(G) of length $\leq km$ are generated by initial productions from S (point 3).

(ii) if a word of length $>km$ is to be generated non-trivially by G, at least one non-terminal on the right of an initial production must lead to at least two deposits of terminals. (This allows proper operation of 7 and 8). The productions of 1 allow G' to pick one such non-terminal. Productions from 3, 5 and 6 allow the derivation by G' to procede essentially as it did by G except for the presence of a Y_a. The non-terminal to the immediate right of the Y_a keeps track of Y_a until the first deposit of terminals (7). If the first terminal deposited by G_{X_i} in the G-derivation is an "a", then all terminals except "a" are deposited and generation now procedes as it did in G(8). When termination occurs the "a" is deposited in the correct place by Y_a (4).

We now give a detailed proof that $L(G) = L(G')$.

<u>CLAIM 1</u>: $L(G) \subseteq L(G')$.

<u>PROOF</u>: Let $x \in L(G)$. If $|x| \leq km$ then $x \in L(G')$ by construction. Otherwise $|x| > km$ and $S \to x \in P$ implies $x \in L(G')$ by (2), or there exists a derivation $D: S = Q_0 \underset{G}{\Longrightarrow} Q_1 \underset{G}{\Longrightarrow} \cdots \underset{G}{\Longrightarrow} Q_n = x$ with $n > 2$. (By the definition of m, the maximum possible length of Q_2 is km and since $|x| > km$ we know $n > 2$.)

In the second case x can be factored $x = x_1 \ldots x_k$ where $Q_1 = X_{11} \ldots X_{1k}$ and $x_i \in L(X_{1i}), 1 \leq i \leq k$. Also each x_i can be factored $x_i = y_{2i} y_{3i} \cdots y_{ni}$ where $X_{j-1,k} \to y_{ji} X_{ji}$ is the production applied to the i-th non-terminal in $Q_{j-1} \Longrightarrow Q_j$, $2 \leq j \leq n-1$ and $X_{n-1,i} \to y_{ni}$ is the production applied to the i-th non-terminal in $Q_{n-1} \Longrightarrow Q_n$. Hence $y_{ji} \in T^*$ $1 \leq i \leq k$, $2 \leq j \leq n-1$ and $y_{ni} \in T^+$, $1 \leq i \leq k$.

For some i there exists a $j < n$ such that

$y_{ji} \neq \varepsilon$ for otherwise x could have length at most km. We fix such an i and let j be the least j such that $y_{ji} \neq \varepsilon$. Suppose $y_{ji} = a\overline{y}_{ji}, a \in T, \overline{y}_{ji} \in T^*$. Clearly $a \in F(L(X_{1i}))$. By 7 and 8, we have the following list of productions in P':

$$[X_{1i}, a] \to [X_{2i}, a]$$
$$\vdots \qquad \vdots$$
$$[X_{j-2,i}, a] \to [X_{j-1,i}, a]$$
$$[X_{j-1,i}, a] \to \overline{y}_{ji} X_{ji}.$$

Thus we have the following derivation for x in G':

(I) $$S \Longrightarrow X_{11} \cdots X_{1,i-1} Y_a [X_i, a] \cdots X_{1k}$$

$$\stackrel{*}{\Longrightarrow} y_{21} \cdots y_{j-1,1} X_{j-1,1} \cdots Y_a [X_{j-1,i}, a] y_{2,i+1} \cdots y_{j-1,k} X_{j-1,k}$$

$$\Longrightarrow y_{21} \cdots y_{j1} X_{j1} \cdots Y_a \overline{y}_{ji} X_{ji} \cdots X_{jk}$$

$$\stackrel{*}{\Longrightarrow} y_{21} \cdots y_{n-1,1} X_{n-1,1} \cdots Y_a \overline{y}_{ji} \cdots y_{n-1,i} X_{n-1,i} \cdots X_{n-1,k}$$

$$\Longrightarrow y_{21} \cdots y_{n1} \cdots a\overline{y}_{ji} \cdots y_{ni} \cdots y_{nk} = x.$$

Thus $x \in L(G')$ and we have $L(G) \subseteq L(G')$.

<u>Claim 2</u>: $L(G') \subseteq L(G)$.

PROOF: If $x \in L(G')$ and $S \to x \in P'$ then $x \in L(G)$ by construction. Otherwise there exists a non-trivial derivation for x by G' which must take at least three steps. This is because all non-terminating initial productions are of the form 1 . A non-terminal of the $[X, a]$ type can lead to termination only after application of a production from 8 and a production from 6. This requires at least two steps after the initial production. Thus the derivation of x must have the form (I) above. Now by 7 and 8, the productions used in this derivation of x at the $i + 1$ - st non-terminal before the j-th step were constructed from productions of P to allow the following derivation of x by G (where x is factored as before):

$$S \Longrightarrow X_{11} \cdots X_{1k}$$

$$\overset{*}{\Longrightarrow} y_{21} \cdots y_{j-1,1} X_{j-1,1} \cdots X_{j-1,i-1} X_{j-1,i} y_{2,i+1} \cdots y_{j-1,k} X_{j-1,k}$$

$$\Longrightarrow y_{21} \cdots y_{j1} X_{j1} \cdots X_{j-1,i-1} a \bar{y}_{ji} X_{ji} \cdots X_{jk}$$

$$\overset{*}{\Longrightarrow} y_{21} \cdots y_{n1} \cdots y_{n,i-1} a \bar{y}_{ji} \cdots y_{ni} \cdots y_{nk} = x.$$

Thus $x \in L(G)$ and $L(G') \subseteq L(G)$.

Claim 1 and Claim 2 give $L(G) = L(G')$, so we have $L \in \mathcal{L}_{k+1}$.

EXAMPLE 2.3. Consider the 2-rlg $G = (\{S, A, B\}, \{a, b\}, S, P, 2)$ where P contains:

$$S \to AB$$
$$A \to aA | A | a$$
$$B \to bB | b.$$

Evidently $L(G) = \{a^n b^m | m \geq n \geq 1\}$. We apply the construction of Theorem 2.3 to give a 3-rlg for $L(G)$. First we note that $F(L(A)) = \{a\}$ and $F(L(B)) = \{b\}$, and that $m = 2$ so $km = 4$. The set of non-terminals for the new grammar is

$$N' = \{S, A, B, [S, a], [S, a], [S, b], [A, a], [A, b], [B, a], [B, b], Y_a, Y_b\}.$$

The new production set P' contains (where numbers below refer to the construction in Theorem 2.3):

(1) $S \to Y_a[A, a]B$, $S \to AY_b[B, b]$

(3) $S \to ab | a^2b^2 | ab^2 | ab^3$

(4) $Y_a \to Y_a | a$, $Y_b \to Y_b | b$

(5), (6) $A \to aA | A | a$, $B \to bB | b$

(7) $[A, a] \to [A, a]$

(8) $[A, a] \to A$, $[B, b] \to B$.

Now $G' = \{N', \{a, b\}, S, P', 3\}$ is a 3-rlg for $L(G)$.

We give some sample derivations by G':

$S \Longrightarrow Y_a[A,\ a]B \Longrightarrow Y_a AbB \Longrightarrow a^2b^2$

$S \Longrightarrow Y_a[A,\ a]B \Longrightarrow Y_a[A,\ a]bB \Longrightarrow Y_a AbbB$

$\Longrightarrow Y_a aAb^3B \Longrightarrow a^3b^4$

$S \Longrightarrow AY_b[B,\ b] \Longrightarrow AY_bB \Longrightarrow aAY_bbB \Longrightarrow a^2b^3.$

Our next result generalizes the iterating factor theorem for regular languages to 2-rlg's. First, however, some comments on derivation trees are in order. Since the grammar underlying a k-rlg is context-free we can attach a derivation tree to a generation of a word by a k-rlg. Since the form of productions is restricted and the manner of generation is "k-parallel" we can be quite specific about the nature of possible derivation trees. We first give examples to illustrate:

Example 2.4. Using the grammar G_3 from Ex. 2.1 we have the derivation $S \overset{*}{\Longrightarrow} a^3b^3c^3$. The tree associated with this derivation is

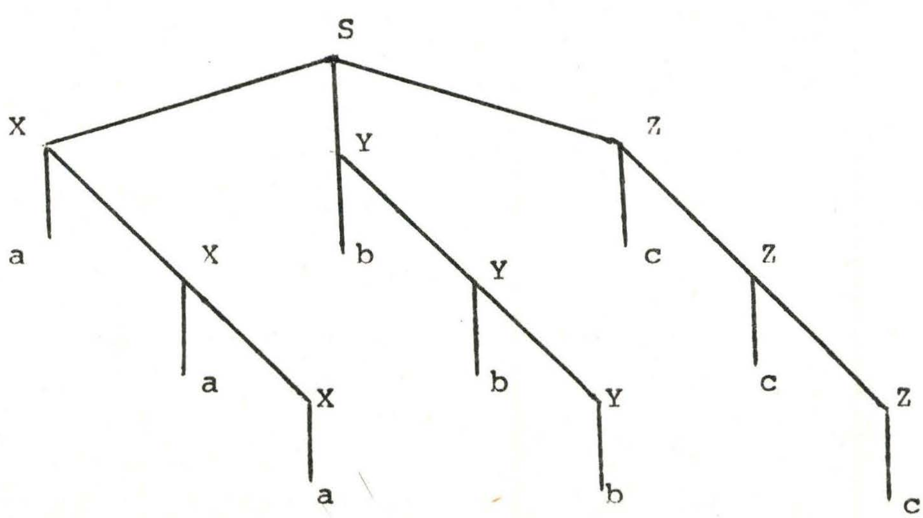


FIGURE 2.1

EXAMPLE 2.5. Using the grammar G from Ex. 2.2 we had a derivation $S \overset{*}{\Longrightarrow} a^3cd$. The tree associated with this derivation is

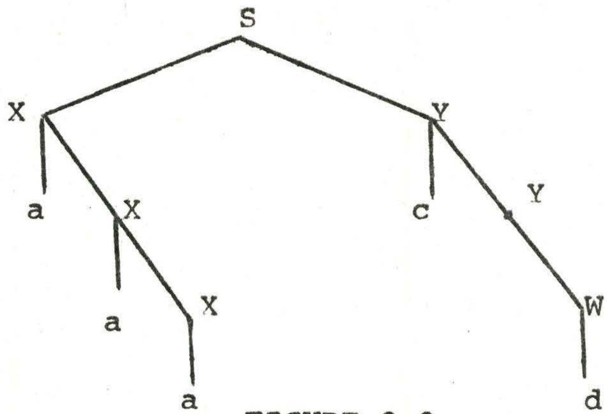
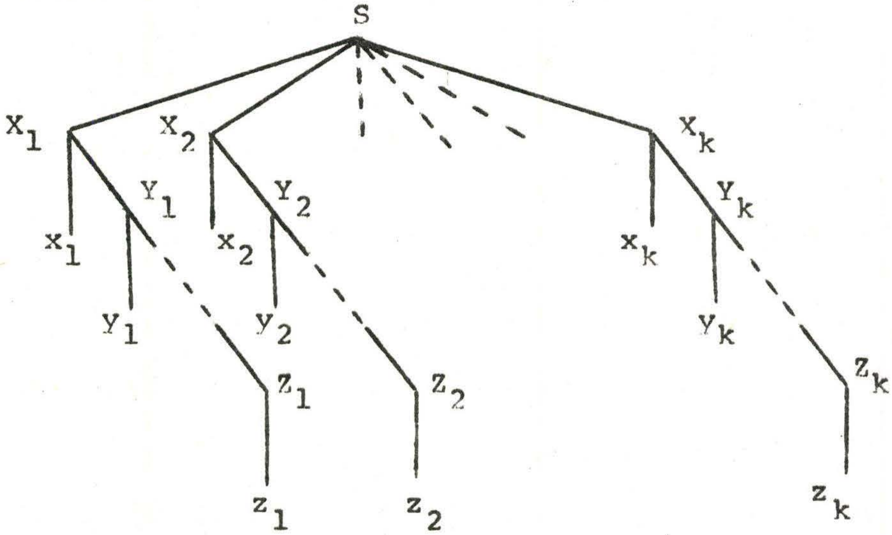


FIGURE 2.2

In general the trees associated with derivation by a k-rlg look like the tree of Fig. 2.3, i.e. an initial branching to k subtrees all of which have the same length and all of which have leaves only to the left.



note: Some of x_i, y_i, z_i may be labelled ε.

FIGURE 2.3

<u>THEOREM 2.4</u>: Let $L \in \mathcal{L}_2$. There exist positive integers p, n, r, s such that if $x \in L$ and $|x| > p$ then $x = uvwu'v'w'$ with $|v| + |v'| > 0$, $|v|, |v'| \leq n$ and for all $i > 0$ $uv^{\bar{r}i}wu'v'^{\bar{s}i}w' \in L$ where $q = \text{lcm}\{r, s\}$ and $\bar{r} = \frac{q}{r}$, $\bar{s} = \frac{q}{s}$.

<u>PROOF</u>: Let $G = (N, T, S, P, 2)$ be a 2-rlg such that $L = L(G)$ and suppose $\#(N) = j$ and $\max\{|x| \mid X \to x \in P\} = \ell$. Let $p = 2j\ell$ and suppose $x \in L$ and $|x| > p$. Then for some $A, B \in N$ we have $S \Longrightarrow AB \overset{*}{\Longrightarrow} x$, moreover there exist $y, z \in T^+$ such that $A \underset{G_A}{\overset{*}{\Longrightarrow}} y$, $B \underset{G_B}{\overset{*}{\Longrightarrow}} z$ and $x = yz$. One of $|y|$ and $|z|$ is greater than $j\ell$ and we conclude that in the corresponding subtree of the derivation tree for x there must be a repeated node-name. Moreover there must be a repetition of node-name which is "non-trivial" in that terminals are deposited between the first and second occurrences (for otherwise $j\ell$ terminals could never be deposited). Since the generation of x procedes in parallel the number of non-terminals appearing in the other subtree is equal to that of the first, and a node-name must be repeated there as well.

Now suppose there is a repeated node-name in the tree for y separated by $r - 1$ non-terminal nodes, and a repeated node name in the tree for z separated by $s - 1$ non-terminal nodes which satisfy the conditions:

(1) at least one of the repetitions is non-trivial and

(2) in each case the repeated node-name does not occur among the names for the separating nodes.

The subtrees thus picked out generate words v and v' respectively which are not both empty and since $r, s \leq j$ we have $|v|, |v'| \leq j\ell = n$.

Since q is a common multiple of r and s, a subtree of length q may be inserted in the y-tree and the z-tree which generates respectively $v^{\bar{r}}$ and $v'^{\bar{s}}$. The resulting tree is a tree for a terminating derivation by G of $uv^{\bar{r}}wu'v'^{\bar{s}}w'$ where $y = uvw$ and $z = u'v'w$. We may iterate the insertion of subtrees of length q to get $uv^{\bar{r}i}wu'v'^{\bar{s}i}w' \in L(G)$ for all $i > 0$.

We can generalize this result to

<u>THEOREM 2.5</u>: Let $L \in \mathcal{L}_k$. Then there exist positive integers $p, n, r_1, \ldots, r_k$ such that if $x \in L$ and $|x| > p$ then $x = u_1v_1w_1 \ldots u_kv_kw_k$, v_i not all ε, $|v_i| \leq n$ $1 \leq i \leq k$ and for all $j > 0$ $u_1v_1^{\bar{r}_1 j}w_1 \ldots u_kv_k^{\bar{r}_k j}w_k \in L$ where $q = \ell cm\ \{r_1, \ldots, r_k\}$ and $\bar{r}_i = \frac{q}{r_i}$, $i \in [1, k]$.

<u>PROOF</u>: Take $p = kj\ell$ and procede as above.

<u>THEOREM 2.6</u>. $\mathcal{L}_k \subsetneq \mathcal{L}_{k+1}$ for all $k \geq 1$. Thus the families $\mathcal{L}_k$ form a proper infinite hierarchy of

language families.

PROOF: By Theorem 2.3 we have only to show the existence of a language in $\mathcal{L}_{k+1} - \mathcal{L}_k$ for all $k \geq 1$. When $k = 1$ we can use $L_2 = \{a^n b^n | n \geq 1\}$ for this language is clearly in $\mathcal{L}_2$ (modify Ex. 2.1 to give a 2-rlg for it) but L_2 is not regular, so not in $\mathcal{L}_1$. When $k = 2$ we can use $L_3 = L(G_3) = \{a^n b^n c^n | n \geq 1\}$. By Ex. 2.1 $L_3 \in \mathcal{L}_3$ and we apply Theorem 2.4 to show $L_3 \notin \mathcal{L}_2$. Suppose $L_3 \in \mathcal{L}_2$ and let p, n, r, s be positive integers satisfying Theorem 2.4 for L_3. Let q be a positive integer so that $|a^q b^q c^q| > p$, then $a^q b^q c^q = uvwu'v'w'$ with v and v' not both ε. Neither v nor v' can consist of a single letter for if it did increasing powers of that letter (those letters) would occur while the third letter did not increase in power since $uv^{\overline{ri}}wu'v'^{\overline{si}}w' \in L_3$ all $i > 0$ by Th. 2.4. Now if either of v or v' has more than one letter we should have words in L_3 containing powers of one of $a^\ell b^m$, $a^\ell b^m c^k$ or $b^m c^k$ for integers $k, \ell, m \leq q$. This is impossible. We conclude $L_3 \notin \mathcal{L}_2$.

By similar arguments $L_{k+1} = \{a_1^n a_2^n \ldots a_{k+1}^n | n \geq 1\}$ is in $\mathcal{L}_{k+1}$ (modify G_3 to G_k) but not in $\mathcal{L}_k$ (by Theorem 2.5) for all $k > 0$. This completes the proof.

Before we summarize the known relationships between the families $\mathcal{L}_k$ and $\mathcal{L}$ and the Chomsky Hierarchy we give a relevant lemma.

<u>Lemma 2.7</u>: $\mathcal{L}_2 \subseteq \mathcal{L}_{CF}$.

<u>PROOF</u>: Letting $L \in \mathcal{L}_2$ implies there exists a 2-rlg $G = (N, T, S, P, 2)$ such that $L = L(G)$. We construct a pda M such that $L = N(M)$. We let $M = (Q, T, \delta, \{A, B\}, S, B, \phi)$ where $Q = \{S\} \cup (V \times (N \cup \{\varepsilon\}))$ and

$$V = N \cup \{xX \mid x \in T^+,\ X \in N,\ \exists\, Y \to zxX \in P,\ z \in T^*\}$$
$$\cup \{x \in T^+ \mid \exists\, Y \to zx \in P,\ z \in T^*,\ Y \in N\} \cup \{\varepsilon\}.$$

δ is constructed as follows:

(1) $\delta(S, \varepsilon, B) = \{([X_1, X_2], B) \mid S \to X_1X_2 \in P\} (\cup \{(S, \varepsilon)\}$ if $S \to \varepsilon \in P)$

$\delta(S, a, B) = \{([y, \varepsilon], B) \mid S \to ay \in P,\ y \in T^+\} (\cup \{(S, \varepsilon)\}$ if $S \to a \in P)$

(2) $\delta([X_1, X_2], a, B) = \{([yY, X_2], AB) \mid X_1 \to ayY \in P,\ y \in T^*\}$

$\cup \{([y, X_2], AB) \mid X_1 \to ay \in P,\ y \in T^+\}$

$(\cup \{([X_2, \varepsilon], AB)\}$ if $X_1 \to a \in P)$

$\delta([X_1, X_2], a, A) = \{([yY, X_2], AA) \mid X_1 \to ayY \in P,\ y \in T^*\}$

$\cup \{([y, X_2], AA) \mid X_1 \to ay \in P,\ y \in T^+\}$

$(\cup \{([X_2, \varepsilon], AA)\}$ if $X_1 \to a \in P)$

$\delta([X_1, X_2], \varepsilon, B) = \{([Y, X_2], AB) | X_1 \to Y \in P, Y \in N\}$

$\delta([X_1, X_2], \varepsilon, A) = \{([Y, X_2], AA) | X_1 \to Y \in P, Y \in N\}$

(3) $\delta([xX_1, X_2], a, A) = \{([yX_1, X_2], A) | x = ay, y \in T^*\}$

(4) $\delta([y, X_2], a, A) = \{([z, X_2], A) | y = az, z \in T^+\}$

$\delta([a, X_2], a, A) = \{([X_2, \varepsilon], A)\}$

(5) $\delta([X_2, \varepsilon], a, A) = \{([yY, \varepsilon], \varepsilon) | X_2 \to ayY \in P, y \in T^*, Y \in N\}$

$\{([y, \varepsilon], \varepsilon) | X_2 \to ay \in P, y \in T^*\}$

$\delta([X_2, \varepsilon], \varepsilon, A) = \{([Y, \varepsilon], \varepsilon) | X_2 \to Y \in P, Y \in N\}$

(6) $\delta([yX, \varepsilon], a, A) = \{([zX, \varepsilon], A) | y = az, z \in T^*\}$

(7) $\delta([y, \varepsilon], a, B) = \{([z, \varepsilon], B) | y = az, z \in T^+\}$

$\delta([a, \varepsilon], a, B) = \{([\varepsilon , \varepsilon], \varepsilon)\}$

(8) $\delta(q, b, c) = \phi$ $c \in \{A, B\}$, $q \in Q$, $b \in T \cup \{\varepsilon\}$ in all other cases.

While the construction of M is quite complex its operation is simply described. M adds one symbol to the pushdown store each time a production is found in the tree resulting from the first non-terminal of an initial production (point 2) . The second non-terminal of this initial production is "remembered" in the second component of the state. When the derivation in the first tree terminates this initial non-terminal is moved to the first component of the state (point 4) and the productions used are counted off as they are found (point 5). If an equal number of productions have been found when this derivation terminates (point 7), the input word is accepted by empty store. Note that words in $L(G)$ by an $S \to x$ production, where $x \in T^*$, are accepted by operation of 1 and 7.

Finally, $L = N(M)$ so $L \in \mathcal{L}_{CF}$.

COROLLARY 2.8: $\mathcal{L}_2$ is contained in the family of one-counter languages.

PROOF: We used only a bottom-marker and one other push-down symbol (the "counter") in our construction.

THEOREM 2.9.

(1) The family $\mathcal{L}_1$ equals the family of regular sets and for every $k \geq 2$, $\mathcal{L}_k$ contains non-regular languages.

(2) $\mathcal{L}_2 \subsetneqq \mathcal{L}_{CF}$ and for every $k \geq 3$, $\mathcal{L}_k$ contains non-context free languages.

(3) $\mathcal{L} \subsetneqq \mathcal{L}_{CS}$.

(4) There exist context-free languages not in $\mathcal{L}$ (and so not in $\mathcal{L}_k$ for any k).

PROOF:

(1) The first part is by Lemma 2.1. $L_2 = \{a^n b^n | n \geq 1\}$ is a non-regular language in $\mathcal{L}_2$ and hence in $\mathcal{L}_k$ for all $k \geq 2$.

(2) The first part is by Lemma 2.7 and 4 below. $L_3 = \{a^n b^n c^n | n \geq 1\}$ is a non-context-free language in $\mathcal{L}_3$ and hence in $\mathcal{L}_k$ for all $k \geq 3$.

(3) This follows from $\mathcal{L}_k \subseteq \mathcal{R}_k$ and Corollary 3.6 below, and Theorem 1.3 of Ibarra [4].

(4) This is from Corollary 3.7 below.

§3. ε-RULES AND FACTOR LANGUAGES

In this section we show that allowing ε-rules does not change the generative capacity of k-rlg's and that the 'language of i-th factors' of a k-rll is regular.

LEMMA 2.10: The family $\mathcal{L}_k$ is closed under union for all $k \geq 1$.

PROOF: Let $L_1, L_2 \in \mathcal{L}_k$ and let $G_1 = (N_1, T, S_1, P_1, k)$ and $G_2 = (N_2, T, S_2, P_2, k)$ be such that $L_1 = L(G_1)$ and $L_2 = L(G_2)$. We assume $N_1 \cap N_2 = \phi$ and $S \notin N_1 \cup N_2$. Let $G = (N_1 \cup N_2 \cup \{S\}, T, S, P, k)$ where P contains:

(1) $S \to X_1 \ldots X_k$ if $S_1 \to X_1 \ldots X_k \in P_1$, $X_i \in N_1$, $1 \leq i \leq k$

$S \to Y_1 \ldots Y_k$ if $S_2 \to Y_1 \ldots Y_k \in P_2$, $Y_i \in N_2$, $1 \leq i \leq k$

(2) $S \to x$ if $S_1 \to x \in P_1$, $x \in T^*$

$S \to y$ if $S_2 \to y \in P_2$, $y \in T^*$

(3) $X \to yY$ if $X \to yY \in P_1 \cup P_2$, $y \in T^*$ and $X, Y \in N_1$ or $X, Y \in N_2$.

(4) $X \to x$ if $X \to x \in P_1 \cup P_2$, $x \in T^*$, $X \in N_1 \cup N_2$.

Clearly $L(G) = L(G_1) \cup L(G_2) = L_1 \cup L_2$, therefore $L_1 \cup L_2 \in \mathcal{L}_k$.

NOTATION: In what follows we denote for $1 \leq i \leq k$

$$\mathcal{R}_i = \{\varphi: [1, i] \to [1, k] \mid \varphi \text{ is one-one and } n < m \Longrightarrow \varphi(n) < \varphi(m) \text{ for all } n, m \in [1, i]\}.$$

We define a k-parallel right-linear grammar with ε-rules (ε- krlg) exactly as in Dfn. 2.1 except that point (3) is modified to (3') $X \to x \in P$ implies $x \in T^*N \cup T^*$. This means we allow terminating rules of the form $X \to \varepsilon$. We define the yield relation for an ε-krlg exactly as in Dfn. 2.2 and denote the family of languages generated by ε-krlg's by $\mathcal{L}_k^{\varepsilon}$. It is immediate from Lemma 2.1 that $\mathcal{L}_1 = \mathcal{L}_1^{\varepsilon} = \mathcal{L}_{REG}$. We also note that a slight modification of Lemma 2.2 shows that the membership problem is decidable for languages specified by ε-krlg's.

Definition 2.4: Let $L \in \mathcal{L}_k^{\varepsilon}$ and $x \in L$. Fix an ε-krlg G for L. Then $\pi_i(D, x)$ $1 \leq i \leq k$ is defined to be the subword of x generated by the i'th non-terminal on the right side of the initial production of some derivation D of x by G.

Note that $\pi_i(D, x)$ is defined only if there is a non-trivial derivation of x by G and in this case $x = \pi_1(D, x)\pi_2(D, x)\ldots\pi_k(D, x)$ for all derivations D of x by G.

THEOREM 2.11. $\mathcal{L}_k = \mathcal{L}_k^{\varepsilon}$.

PROOF: Each language L in $\mathcal{L}_k$ is generated by a k-rlg which is trivially an ε-krlg, so $L \in \mathcal{L}_k^{\varepsilon}$. Thus $\mathcal{L}_k \subseteq \mathcal{L}_k^{\varepsilon}$.

The reverse inclusion is more interesting: let $L \in \mathcal{L}_k^{\varepsilon}$ and $G = (N, T, S, P, k)$ be an ε-krlg such that $L = L(G)$. For all $i \in [1, k]$, for all $\varphi \in \mathfrak{K}_i$ we define

$$L_i^{\varphi} = \{x \in L \mid \exists \text{ a derivation } D \text{ of } x \text{ by } G \text{ satisfying } \pi_j(D, x) \neq \varepsilon \text{ for all } j \in \operatorname{im}\varphi,\ \pi_j(D, x) = \varepsilon \text{ otherwise}\}.$$

Define $L_i = \bigcup_{\varphi \in \mathfrak{K}_i} L_i^{\varphi}$ $1 \leq i \leq k$ and $L_0 = \{x \in T^* \mid S \to x \in P\}$.

Then

$$L = \begin{cases} \bigcup_{i=0}^{k} L_i & \text{if } \varepsilon \notin L \\ \bigcup_{i=0}^{k} L_i \cup \{\varepsilon\} & \text{if } \varepsilon \in L. \end{cases}$$

We next claim $L_i^{\varphi} \in \mathcal{L}_i$ for all $\varphi \in \mathfrak{K}_i$. To see this we construct $G_i^{\varphi} = (N', T, S, P_i^{\varphi}, i)$ where $N' = N \cup \{[X, a] \mid X \in N, a \in T\}$ and P_i contains:

(1) $S \to X_{\varphi(1)} \ldots X_{\varphi(k)}$ whenever $S \to X_1 \ldots X_k \in P$.

(2) $X \to yY$ whenever $X \to yY \in P$, $y \in T^*$, $X, Y \in N$ and $\varepsilon \notin L(Y)$.

(3) $X \to y[Y, a]$ whenever $X \to yaY \in P$, $a \in T$, $y \in T^*$, $X, Y \in N$ and $\varepsilon \in L(Y)$.

(4) For all $a \in T$, $[X, a] \to [Y, a]$ whenever $X \to Y \in P$, $X, Y \in N$.

(5) For all $a \in T$, $[X, a] \to ay[Y,b]$ whenever $X \to ybY \in P$, $b \in T$, $y \in T^*$ $X, Y \in N$ and $\varepsilon \in L(Y)$.

(6) For all $a \in T$, $[X, a] \to ayY$ whenever $X \to yY \in P$, $y \in T^*$, $X, Y \in N$ and $\varepsilon \notin L(Y)$.

(7) For all $a \in T$, $[X, a] \to ay$ whenever $X \to y \in P$, $X \in N$, $y \in T^*$.

(8) $X \to y$ whenever $X \to y \in P$, $y \in T^+$.

The construction of G_i^{φ} is essentially similar to that in Lemma 2.1. Since for all $x \in L_i^{\varphi}$, $\pi_j(D, x) \neq \varepsilon$ for all $j \in \text{im}\,\varphi$ we know that at least one terminal letter is deposited in the j'th subword of x. A terminal letter which is potentially the last one deposited is carried through the derivation (points 3 and 4) until either more terminals are deposited (5 and 6) or the derivation terminates (7). The other productions are as before (2 and 8) except that the initial productions pick out only the productive non-terminals (1). Thus G_i^{φ} is an i-rlg which generates L_i^{φ}. By Lemma 2.10, $L_i \in \mathcal{L}_i$ $1 \leq i \leq k$. $L_0 \in \mathcal{L}_1$ since it is finite. Thus $L_i \in \mathcal{L}_k$, $0 \leq i \leq k$ (by Theorem 2.3) and since we can decide if $\varepsilon \in L$ or not, we have $L \in \mathcal{L}_k$ (another application of 2.10). Thus $\mathcal{L}_k^{\varepsilon} \subseteq \mathcal{L}_k$. This completes the proof.

REMARK: This theorem leads to the question 'Why not allow $X \to \varepsilon$ rules in the first place?' for then the analogue of Theorem 2.3 would be a triviality. The answer is that Theorem 2.11, which is a most desirable result in either case, does not follow without heavy use of Theorem 2.3 for k-rlg's as we have defined them.

DEFINITION 2.5: Let $L \in \mathcal{L}_k$ and G be a k-rlg for L. For $1 \leq i \leq k$, $X \in N$ we define

$$\hat{L}_i(X) = \{\pi_i(D, z) \mid \exists \text{ a derivation } D: S \Longrightarrow X_1 \ldots X_k \Longrightarrow \ldots \Longrightarrow z$$

$$\text{where } X_i = X\}$$

and $\hat{L}_i = \bigcup_{X \in N} \hat{L}_i(X)$.

This means $\hat{L}_i(X)$ is the language consisting of i'th factors of words generated when X is the i'th non-terminal on the right side of an initial production. L_i is the language consisting of all i'th factors of non-trivially generated words.

EXAMPLE 2.5: Consider $G = (\{S,A,B,C,D\},\{a,b\},S,P,2)$ where P is given by:

$S \to AB$ $\quad A \to C$ $\quad B \to bB|b$

$C \to D$ $\quad D \to aA|a.$

Clearly $L(G) = \{a^n b^{3n} | n \geq 1\}$ but here $b^* = L(B) \neq \hat{L}_2(B) = \hat{L}_2 = \{b^{3n} | n \geq 1\}$. Thus while $L(X)$ is regular for all $X \in N$ $(X \neq S)$, we have to consider $\hat{L}(X)$ and so $\hat{L}_i$ separately.

THEOREM 2.12. Let $L \in \mathcal{L}_k$ and fix a k-rlg $G = (N, T, S, P, k)$ for L. Then $\hat{L}_i$ is regular $i = 1, \ldots, k$.

PROOF: We will show that each L_i is generated by a right-linear grammar with a regular control language and so [by Salomaa [9]) is regular. First let Lab(P) be a set of labels for productions in P, say $Lab(P) = \{a_j | 1 \leq j \leq n\}$ and we denote by $X \xrightarrow{a_j} x$ that a_j is a label for $X \to x \in P$.

We say a k-tuple of non-terminals $(X_1, \ldots, X_k) \in N^k$ "terminates" if there is an $x_j \in T^+$ such that $X_j \to x_j \in P$

$1 \leq j \leq k$.

We say a k-tuple of non-terminals $(X_1,\ldots,X_k)$ "yields" another k-tuple $(Y_1,\ldots,Y_k)$ (written as $(X_1,\ldots,X_k)\rightarrow(Y_1,\ldots,Y_k)$) if there exist productions in P: $X_j \rightarrow y_j Y_j$, $y_j \in T^*$ $1 \leq j \leq k$.

We now construct k nfsa's M_i $1 \leq i \leq k$ by $M_i = (N^k \cup \{S\} \cup \{F\}, Lab(P), \delta_i, S, \{F\})$ where δ_i is defined by:

(1) $\delta_i(S, a_j) = \{(X_1,\ldots,X_k) \in N^k \mid S \xrightarrow{a_j} X_1 \ldots X_k \quad P\}, 1 \leq j \leq n.$

(2) $\delta_i((X_1,\ldots,X_k), a_j) = \{(Y_1,\ldots, Y_k) \in N^k \mid (X_1,\ldots, X_k) \rightarrow (Y_1,\ldots,Y_k)$ and $X_i \xrightarrow{a_j} yY_i$ some $y \in T^*\}(\cup\{F\}$ if $(X_1,\ldots, X_k)$ terminates and $X_i \xrightarrow{a_j} x$ for some $x \in T^+)$ $1 \leq j \leq n$.

(3) $\delta_i(q, a_j) = \phi$ otherwise for all $q \in N^k \cup \{S\} \cup \{F\}$, $1 \leq j \leq n$.

We now define k right-linear grammars G_i by $G_i = (N, T, S, P_i)$ where $P_i = (P - \{S \rightarrow x \mid S \rightarrow x \in P\}) \cup \{S \rightarrow X_i \mid S \rightarrow X_1 \ldots X_k \in P, X_j \in N, 1 \leq j \leq k\}$. We now label the productions of P_i by using the same labels as above for productions of P and giving the new productions the label of the production of P from which they were constructed (i.e. $S \xrightarrow{a_j} X_i$ if $S \xrightarrow{a_j} X_1 \ldots X_k$).

We claim that $\hat{L}_i = L(G_i, T(M_i))$. Now $x_i \in \hat{L}_i$ iff there exists $x \in L$ such that $x = x_1 \ldots x_k$ and $x_j \in \hat{L}_j$ $1 \leq j \leq n$ iff there exists a derivation $S \Longrightarrow X_1 \ldots X_k \Longrightarrow \ldots \Longrightarrow x$ with the productions at the i-th place labelled so that the control word is in $T(M_i)$ iff $x_i \in L(G_i, T(M_i))$. Thus $\hat{L}_i$ is generated by a right-linear grammar G_i, with regular control language $T(M_i)$ and therefore $\hat{L}_i$ is a regular set.

§4. CLOSURE PROPERTIES

In this section we consider closure properties of the families $\mathcal{L}_k$ and we then give a simple characterization of $\mathcal{L}_k$.

THEOREM 2.13: For all $k \geq 1$, $\mathcal{L}_k$ is closed under union and finite substitution.

PROOF: Closure under union is by Lemma 2.10. Next let $L \in \mathcal{L}_k$ and $G = (N, T, S, P, k)$ be a k-rlg for L. Let $f: T \to 2^{\Sigma^*}$ be a finite substitution. We define an ε-krlg $G_f = (N, \Sigma, S, P_f, k)$ for $f(L)$ where P_f contains:

(1) $S \to X_1 \ldots X_k$ whenever $S \to X_1 \ldots X_k \in P$, $X_i \in N$, $1 \leq i \leq k$.

(2) $X \to zY$ if $z \in f(y)$, $X \to yY \in P$, $y \in T^*$, $X, Y \in N$.

(3) $X \to z$ if $z \in f(x)$, $X \to x \in P$, $x \in T^+$, $X \in N$.

Clearly $L(G_f) = f(L)$, hence $f(L) \in \mathcal{L}_k^\varepsilon$ and, by Theorem 2.11, $f(L) \in \mathcal{L}_k$.

COROLLARY 2.14: $\mathcal{L}$ is closed under union and finite substitution.

PROOF: Let $L_1, L_2 \in \mathcal{L}$, then $L_1 \in \mathcal{L}_{k_1}$ and $L_2 \in \mathcal{L}_{k_2}$ for some k_1, k_2. Let $k = \max\{k_1, k_2\}$ and we have $L_1, L_2 \in \mathcal{L}_k$, so $L_1 \cup L_2 \in \mathcal{L}_k$ and thus $L_1 \cup L_2 \in \mathcal{L}$. Similarly we have closure under finite substitution.

COROLLARY 2.15. $\mathcal{L}_k$ and $\mathcal{L}$ are closed under homomorphism.

THEOREM 2.16: For all $k \geq 1$, $\mathcal{L}_k$ is closed under intersection with a regular set.

PROOF: Let L be a k-rll and $G = (N,T,S,P,k)$ be a k-rlg for L. Let R be a regular set and $M = (Q, T, \delta, s_0, F)$ an fsa such that $R = T(M)$. We will construct a new k-rlg for $L \cap R$. Let $G' = (N', T, S, P', k)$ where $N' = \{S\} \cup (Q \times N \times Q) \cup (Q \times N)$. P' contains:

(1) $S \to x$ if $S \to x \in P$, $x \in T^*$ and $x \in R$.

(2) $S \to [s_0, X_1, s_1][s_1, X_2, s_2] \ldots [s_{k-1}, X_k]$ for all sequences $s_1, \ldots, s_{k-1}$ of members of Q if $S \to X_1 \ldots X_k \in P$, $X_i \in N$ $1 \leq i \leq k$.

(3) $[s_i, X, s_j] \to y[\delta^*(s_i, y), Y, s_j]$ if $X \to yY \in P$, $y \in T^*$, $X, Y \in N$ and $s_i, s_j \in Q$.

(4) $[s_i, X, s_j] \to x$ if $X \to x \in P$, $X \in N - \{s\}$, $x \in T^+$ and $\delta^*(s_i, x) = s_j$.

(5) $[s_i, X] \to y[s_j, Y]$ if $X \to yY \in P$, $y \in T^*$, $X, Y \in N$ and $\delta^*(s_i, y) = s_j$, $s_i, s_j \in Q$.

(6) $[s_i, X] \to x$ if $X \to x \in P$, $X \in N - \{S\}$, $x \in T^+$, $s_i \in Q$ and $\delta^*(s_i, x) \in F$.

In point 1 all words generated trivially by G that are in R are generated by G'. A word is generated non-trivially by G' if it is generated by G (the cores of productions from points 3-6) and is accepted by M (the state components of non-terminals in productions from points 3-6 contain information as to the state of M as it processes a word generated by G. If M is in a final state at the end of a word generated by G, then G' is allowed to generate it.) Since this type of construction will be used again below we give a detailed proof that $L(G') = L \cap R$.

<u>CLAIM 1</u>: $L(G') \subseteq L \cap R$.

<u>PROOF</u>: Let $x \in L(G')$, then either $S \to x \in P'$ and so $x \in L \cap R$ or there exists a derivation $D: S = P_0 \Longrightarrow P_1 \Longrightarrow \ldots \Longrightarrow P_n = x$ in G' and $n \geq 2$. We then have $P_1 = [s_0, X_1, s_1][s_1, X_2, s_2] \ldots [s_{k-1}, X_k]$ for some $X_1, \ldots, X_k \in N$ and $s_1, \ldots, s_{k-1} \in Q$. Moreover $x = x_1 x_2 \ldots x_k$ where $x_i \in L([s_{i-1}, X_i, s_i])$ $1 \leq i \leq k - 1$,

and $x_k \in L([s_{k-1}, X_k])$. Thus, by points 3-6 of the construction, $x_i \in L(X_i)$ $1 \leq i \leq k$ and there is a derivation of x_i of length $n - 1$ from X_i. Hence $S \underset{G}{\Longrightarrow} X_1 \ldots X_k \overset{*}{\underset{G}{\Longrightarrow}} x_1 \ldots x_k = x$ (utilizing also point 2 for the initial production) and so $x \in L(G) = L$. Also, by points 3 and 4,

$$\delta^*(s_i, x_{i+1}) = s_{i+1} \quad 0 \leq i \leq k - 2 \quad \text{and} \quad \delta^*(s_{k-1}, x_k) \in F$$

(by 5 and 6), hence $\delta^*(s_0, x) = \delta^*(s_0, x_1 \ldots x_k) \in F$ and $x \in R$. Thus $x \in L \cap R$ which proves Claim 1.

<u>CLAIM 2</u>: $L \cap R \subseteq L(G')$.

<u>PROOF</u>: Let $x \in L \cap R$, then either $S \to x \in P$ and $\delta^*(s_0, x) \in F$ giving $x \in L(G')$ by point 1 or $\delta^*(s_0, x) \in F$ and there exists a derivation D: $S = P_0 \Longrightarrow P_1 \Longrightarrow \ldots \Longrightarrow P_n = x$ in G where $n \geq 2$. We can factor x for this derivation D as we did in the proof of Theorem 2.3 i.e. $x = x_1 \ldots x_k$ and for $1 \leq i \leq k$ $x_i = y_{2i} \ldots y_{ni}$ with $y_{ji} \in T^*$ $2 \leq j \leq n - 1$ and $y_{ni} \in T^+$. We denote the corresponding productions of P by $X_{ji} \to y_{j+1,i} X_{j+1,i}$ $1 \leq j \leq n - 2$ and $X_{n-1,i} \to y_{ni}$ $1 \leq i \leq k$.

There exist $s_i \in Q$ $1 \leq i \leq k - 1$ such that $\delta^*(s_0, x_1) = s_1$, $\delta^*(s_{i-1}, x_i) = s_i$ and $\delta^*(s_{k-1}, x_k) \in F$. We also have $s_{ji} \in Q$ $1 \leq i \leq k$, $2 \leq j \leq n$ such that $\delta^*(s_{j-1,i}, y_{ji}) = s_{ji}$ $2 \leq j \leq n$, $s_{1i} = s_{i-1}$ and

$s_{ni} = s_i$ $1 \le i \le k$. Now by construction we have
$S \rightarrow [s_0, X_{11}, s_1][s_1, X_{12}, s_2]\ldots[s_{k-1}, X_{1k}]$ in P'.
$[s_{ji}, X_{ji}, s_i] \rightarrow y_{j+1,i}[s_{j+1,i}, X_{j+1,i}, s_i]$ $1 \le j \le n-2$
and $[s_{n-1,i}, X_{n-1,i}, s_i] \rightarrow y_{ni}$ in P' $1 \le i \le k-1$,
by points 3 and 4. We also have
$[s_{jk}, X_{jk}] \rightarrow y_{j+1,k}[s_{j+1,k}, X_{j+1,k}]$ for $1 \le j \le n-2$
and $[s_{n-1,k}, X_{n-1,k}] \rightarrow y_{nk}$ in P' by points 5 and 6.

Thus we have the following derivation of x by G':

$$S \Longrightarrow [s_0, X_{11}, s_1]\ldots[s_{k-1}, X_{1k}]$$
$$\Longrightarrow y_{21}[s_{21}, X_{21}, s_1]\ldots y_{2k}[s_{2k}, X_{2k}]$$
$$\overset{*}{\Longrightarrow} y_{21}\ldots y_{n-1,1}[s_{n-1,1}, X_{n-1,1}, s_1]\ldots y_{n-1,k}[s_{n-1,k}, X_{n-1,k}]$$
$$\Longrightarrow y_{21}\ldots y_{n-1,1}y_{n,1}\ldots y_{n-1,k}y_{nk} = x.$$

Thus $x \in L(G')$ which completes the proof of Claim 2.
Claim 1 and Claim 2 give $L(G') = L \cap R$, so $L \cap R \in \mathcal{L}_k$.

<u>COROLLARY 2.17</u>. $\mathcal{L}$ is closed under intersection with a regular set.

<u>COROLLARY 2.18</u>: $\mathcal{L}_k$ for all $k \ge 1$ and $\mathcal{L}$ are closed under right quotient with a regular set.

<u>PROOF</u>: Lemma 9.5 page 131 of Hopcroft and Ullman [3].

Next we show that, while $\mathcal{L}_1$ is closed under intersection (this is well-known by Lemma 2.1), none of the other families under consideration are closed under intersection.

THEOREM 2.19: For all $k \geq 2$, $\mathcal{L}_k$ is not closed under intersection.

PROOF: We first consider $k = 2$ to make the argument clear: let $L_c = \{(a \cup b)^{2n} c^n | n \geq 1\}$, that is the language consisting of all words of length $3n$ whose first $2n$ letters consist of a's and b's and whose last n letters are c. Let $L_a = \{a^n (b \cup c)^{2n} | n \geq 1\}$. Both L_c and L_a are in $\mathcal{L}_2$. L_c is generated by $G = (\{S, X, C, D\}, \{a, b, c\}, S, P, 2)$ where P contains:

$$S \rightarrow XC$$
$$C \rightarrow D$$
$$D \rightarrow cC | c$$
$$X \rightarrow aX | bX | a | b.$$

A similar 2-rlg generates L_a. Now we consider $L_c \cap L_a$. Let $x \in L_c \cap L_a$ then for some $n \geq 1$ $x = a^n y c^n$ where $|y| = n$ and $y \in (a \cup b \cup c)^*$. Now y has no occurrence of c since $x \in L_c$ and the first $2n$ letters of x must be a or b. Similarly y has no occurrence of a. Hence $y = b^n$. Thus $L_c \cap L_a \subseteq \{a^n b^n c^n | n \geq 1\}$.

Clearly $\{a^n b^n c^n | n \geq 1\} \subseteq L_c \cap L_a$, so $L_c \cap L_a = \{a^n b^n c^n | n \geq 1\} = L_3$. But in Theorem 2.6 we showed that $L_3 \notin \mathcal{L}_2$. Hence $\mathcal{L}_2$ is not closed under intersection.

We can generalize this counterexample by considering $L_{k1} = \{(a_1 \cup a_2)^{2n} a_3^n \ldots a_{k+1}^n | n \geq 1\} \in \mathcal{L}_k$ and $L_{k2} = \{a_1^n a_2^n \ldots a_{k-1}^n (a_k \cup a_{k+1})^{2n} | n \geq 1\} \in \mathcal{L}_k$ and noting that $L_{k1} \cap L_{k2} = \{a_1^n a_2^n \ldots a_{k+1}^n | n \geq 1\} \notin \mathcal{L}_k$.

<u>COROLLARY 2.20</u>: For all $k \geq 2$, $\mathcal{L}_k$ is not closed under complement.

<u>PROOF</u>: If some $\mathcal{L}_k$ were closed under complement, closure under union would imply closure under intersection, contradicting Theorem 2.10.

<u>THEOREM 2.21</u>: For all $k \geq 1$, $\mathcal{L}_k$ is closed under ngsm maps.

<u>PROOF</u>: Let $L \in \mathcal{L}_k$ and $G = (N, T, S, P, k)$ be a k-rlg for L. Let $S = (Q, T, \Delta, \delta, \lambda, q_0, F)$ be an ngsm. We give an ε-krlg for $S(L)$ which shows $S(L) \in \mathcal{L}_k^\varepsilon = \mathcal{L}_k$. Let $G' = (N', \Delta, S, P', k)$ where $N' = (Q \times N \times Q) \cup (Q \times N) \cup \{S\}$ and P' contains:

(1) $S \to z$ if $z \in \lambda^*(q_0, x)$ and $S \to x \in P$, $x \in T^*$.

(2) $S \to [q_0, X_1, q_1][q_1, X_2, q_2] \ldots [q_{k-1}, X_k]$ for all sequences $q_1, \ldots, q_{k-1}$ of members of Q if

$S \to X_1 \ldots X_k \in P$, $X_i \in N$ $1 \le i \le k$.

(3) $[q_i, X, q_j] \to z[\delta^*(q_i, y), Y, q_j]$ if $X \to yY \in P$, $y \in T^*$, $X, Y \in N$, $q_i, q_j \in Q$ and $z \in \lambda^*(q_i, y)$; $[q_i, X] \to z[\delta^*(q_i, y), Y]$ if $X \to yY \in P$, $y \in T^*$, $X, Y \in N$, $q_i \in Q$ and $z \in \lambda^*(q_i, y)$.

(4) $[q_i, X, q_j] \to z$ if $X \to x \in P$, $x \in T^*$, $X \in N$, $\delta^*(q_i, x) = q_j$ and $z \in \lambda^*(q_i, x)$; $[q_i, X] \to z$ if $X \to x \in P$, $x \in T^+$, $X \in N$, $\delta^*(q_i, x) \in F$ and $z \in \lambda^*(q_i, x)$.

G' generates all of $S(x)$ for each $x \in T^*$ generated trivially by G (point 1). If a word x is generated non-trivially by G, each word in $S(x)$ is generated by G' which deposits the "translation" of a word deposited by G, and keeps track of the state of S in its first component. The third component is used to match states at the boundaries corresponding to a factorisation of the word according to the non-terminal from which it is generated (points 2 - 4). The detailed proof that $S(L) = L(G')$ follows the method of Theorem 2.16 and is ommitted.

<u>COROLLARY 2.22</u>: $\mathcal{L}$ is closed under non-deterministic gsm maps.

We are now in a position to give a characterisation of the family $\mathcal{L}_k$ in terms of a closure property. The languages L_k defined above play a fundamental role in the theory of k-parallel right-linear languages so we recall that $L_k = \{a_1^n a_2^n \ldots a_k^n | n \geq 1\}$.

THEOREM 2.23. $\mathcal{L}_k$ is the smallest family of languages containing L_k and closed under non-deterministic gsm mappings for all $k \geq 1$.

PROOF: Let $\mathcal{F}_k$ be the smallest family of languages containing L_k and closed under non det. gsm maps. Since $L_k \in \mathcal{L}_k$ we have $\mathcal{F}_k \subseteq \mathcal{L}_k$ by Theorem 2.21. To show the reverse inclusion let $L \in \mathcal{L}_k$, and $G=(N, T, S, P, k)$ be a k-rlg for L. We will construct an ngsm $M = (Q, \Sigma_k, T, \delta, q_0, F)$ such that $L = M(L_k)$. We first construct $G' = (N', T, S, P', k)$ with $L = L(G')$ where $N' = (N \times \{1, 2, \ldots, k\}) \cup \{S\}$ and P' contains

(1) $S \to x$ if $S \to x \in P$ and $x \in T^*$

(2) $S \to [X_1, 1][X_2, 2] \ldots [X_k, k]$ if $S \to X_1 \ldots X_k$ P
$X_i \in N$, $1 \leq i \leq k$.

(3) $[X, i] \to y[Y, i]$ if $X \to yY \in P$, $X, Y \in N$, $y \in T^*$, $1 \leq i \leq k$.

(4) $[X, i] \to x$ if $X \to x \in P$, $X \in N$, $x \in T^+$, $1 \leq i \leq k$.

Note that each non-terminal in G' carries with it information specifying from which of the k original non-terminals it is generated.

Next we number the initial productions letting the first n be the non-trivial initial productions and productions numbered from $n + 1$ to m be the trivial ones.

Now we can construct M. $\Sigma_k = \{a_1, a_2, \ldots, a_k\}$, $Q = N' \times \{1, 2, \ldots, n\} \cup \{q_0, q_{n+1}, \ldots, q_m, q_f\}$ and $F = \{q_{n+1}, \ldots, q_m, q_f\}$. Next we specify δ:

(1) $\delta(q_0, a_1) = \{(q_i, x) \mid S \to x \in P,\ x \in T^*$ is the ith production$\}$

$\cup \{([Y, 1, j], y) \mid S \to [X_1, 1] \ldots [X_k, k]$ is the jth production and $X_1 \to yY \in P,\ y \in T^*\}$

$\cup \{([X_2, 2, j], x) \mid S \to [X_1, 1](X_2, 2] \ldots [X_k, k]$ is the j'th production and $X_1 \to x \in P$, $x \in T^*\}$.

(2) $\delta(q_i, a_j) = \{(q_i, \varepsilon)\}$ $n + 1 \leq i \leq m,\ 1 \leq j \leq k$.

(3) $\delta([X, i, j], a_i) = \{([Y, i, j], y) \mid [X, i] \to y[Y, i] \in P'$, $y \in T^*,\ X, Y \in N\}$

$\cup \{([Y, i+1, j], y) \mid [X, i] \to y \in P',\ y \in T^+$, Y is $i+1$st non-terminal in initial production $j\}$

for $1 \leq j \leq n,\ 1 \leq i \leq k - 1$.

(4) $\delta([X, k, j], a_k) = \{([Y, k, j], y) \mid [X, k] \rightarrow y[Y, k] \in P',$

$y \in T^*\}$

$\{(q_f, y) \mid [X, k] \rightarrow y \in P', y \in T^+\}$

(5) $\delta(q, a) = \phi$ otherwise for all $q \in Q$, $a \in \Sigma_k$.

M operates by either (1) outputting the result of a trivial derivation and reading the remainder of an input word in a final state with no output (1 and 2) or (2) using the states of M to keep track of a non-terminal in the first component, the position of the non-terminal in the second component, and which initial production was used in the third component. Reading an input symbol causes M to write any terminals deposited by a production from the non-terminal in the first component of the present state, and to change state so that the non-terminal on the right side of the production used appears as the first component of the new state (3). If a terminating production is possible its right side is written and the first component of the new state is the non-terminal in the next slot on the right side of the initial production identified in the third component (3). At the same time the second component is incremented by one. Note that M is allowed to procede only if the subscript of the input letter being read and the second component of the state agree. The input word is used to ensure that the derivation has the same length in each position. (1, 3, 4). An output word is in $M(L_k)$ if and only if it is the result of a terminating derivation by G'. Therefore $L = M(L_k)$, $\mathcal{L}_k \subseteq \mathcal{F}_k$ and the result follows.

REMARK: We can define an operator GSM on families of languages $\mathcal{F}$ (over a fixed countably infinite alphabet) by $GSM(\mathcal{F}) = \bigcap\{\mathcal{M} \mid \mathcal{M} \supseteq \mathcal{F}, \mathcal{M}$ closed under non-det. gsm maps$\}$. It is easy to verify that GSM is a closure operator. In this notation Theorem 2.23 reads $\mathcal{L}_k = GSM(\{L_k\})$.

In the next section we show one more closure property of the families $\mathcal{L}_k$, namely that they are closed under mirror image.

§5. k-PARALLEL LEFT LINEAR LANGUAGES.

In this section we define k-parallel left-linear grammars and show that they generate the same class of languages as k-rlg's.

DEFINITION 2.5: A k-parallel left-linear grammar (k-llg) is a 5-tuple $G = (N, T, S, P, k)$ satisfying (1), (2) and (4) of Definition 2.1 and

3ℓ) $X \to x$ P and $X \neq S$ implies $x \in NT^* \cup T^+$.

As for k-rlg's we can define the class of languages generated by k-llg's which we denote by $\mathcal{L}_k^{\ell}$ and call members of this class k-parallel left-linear languages (k-lll's).

EXAMPLE 2.6. Consider $G_3^{\ell} = (\{S, A, B, C\}, \{a, b, c\}, S, P, 3)$ where P contains:

$$S \to ABC$$
$$A \to Aa \mid a$$
$$B \to Bb \mid b$$
$$C \to Cc \mid c.$$

It should be clear that $L(G_3^{\ell}) = L_3 = \{a^n b^n c^n \mid n \geq 1\}$

which (recall Example 2.1) is also a 3-rll. We see from this example that $L_k \in \mathcal{L}_k^{\ell}$ for all $k \geq 1$, by modifying G_3^{ℓ} to G_k^{ℓ}.

Our aim is to show that $\mathcal{L}_k = \mathcal{L}_k^{\ell}$. To do this we will use Theorem 2.23.

<u>THEOREM 2.24</u>: $\mathcal{L}_k^{\ell}$ is the smallest family of languages containing L_k and closed under non-deterministic gsm mappings for all $k \geq 1$.

<u>PROOF</u>: We let $\mathcal{F}_k$ denote the smallest family. We know that $L_k \in \mathcal{L}_k^{\ell}$. Next we show that $\mathcal{L}_k^{\ell}$ is closed under non-deterministic gsm maps.

<u>CLAIM 1</u>: Allowing ε-rules in 3ℓ) of Definition 2.5 does not change the generative capacity of k-llg's.

<u>PROOF</u>: We observe that $\mathcal{L}_k^{\ell}$ is closed under union (proof similar to Lemma 2.10), then the claim follows by the right-left dual of the proof of Theorem 2.11.

<u>CLAIM 2</u>: $\mathcal{L}_k^{\ell}$ is closed under ngsm mappings.

<u>PROOF</u>: Let $L \in \mathcal{L}_k^{\ell}$ and $G = (N, T, S, P, k)$ be a k-llg for L. Let $M = (Q, T, \Delta, \delta, q_0, F)$ be a non-deterministic gsm. We construct a new k-llg G' for $M(L)$. Let $G' = (N', \Delta, S, P', k)$ where

$N' = (Q \times N \times Q) \cup (N \times Q) \cup \{S\}$ and P' contains:

(1) $S \to [X_1, q_1][q_1, X_2, q_2] \ldots [q_{k-1}, X_k, q_k]$ for all sequences $q_1, \ldots, q_{k-1}$ of members of Q if $S \to X_1 \ldots X_k \in P$, $X_i \in N$, $1 \leq i \leq k$, and $q_k \in F$.

(2) $[q_i, X, q_j] \to [q_i, Y, q_\ell]z$ if $X \to Yy \in P$, $y \in T^*$, $X, Y \in N$, and $(q_j, z) \in \delta^*(q_\ell, y)$.

(3) $[q_i, X, q_j] \to z$ if $X \to x \in P$, $X \in N$, $x \in T^+$ and $(q_j, z) \in \delta^*(q_i, x)$.

(4) $[X, q_j] \to [Y, q_\ell]z$ if $X \to Yy \in P$, $X, Y \in N$, $y \in T^*$ and $(q_j, z) \in \delta^*(q_\ell, y)$.

(5) $[X, q_j] \to z$ if $X \to x \in P$, $X \in N$, $x \in T^+$ and $(q_j, z) \in \delta^*(q_0, x)$.

The operation of G' is similar to that of the grammar constructed in Theorem 2.21. Here however, since generation procedes from right to left we insist that the matching of states in terminal productions take place from right to left (3), that the final state reached be terminal (1) and that the machine started operation from the initial state (5). We conclude that $M(L) = L(G')$ and this completes Claim 2.

We now conclude $\mathcal{F}_k \subseteq \mathcal{L}_k^\ell$ since $\mathcal{L}_k^\ell$ contains L_k and is closed under ngsm maps.

Next we show the reverse inclusion. Let $L \in \mathcal{L}_k^{\ell}$ and $G = (N, T, S, P, k)$ a k-llg for L. Since $\mathcal{F}_k (= \mathcal{L}_k)$ is closed under union, we can number the initial productions of G from 1 to n say and let K_i be the language generated by G when all initial productions but the ith are deleted from P. Clearly $L = K_1 \cup \ldots \cup K_n$. If the ith initial production is trivial then K_i has only one member and $K_i \in \mathcal{F}_k$ since $\mathcal{F}_k$ contains all regular sets. Otherwise let the i-th production be $S \to X_1 \ldots X_k$ say. We construct an ngsm $M_i = (Q, \Sigma_k, T, \delta_i, q_0, F)$ so that $M_i(L_k) = K_i$.

Let $Q = \{q_0\} \cup N'$ where $N' = N \times \{1, 2, \ldots, k\}$, $F = \{[X_k, k]\}$ and δ_i is given by:

(1) $\delta_i(q_0, a_1) = \{([X, 1], x) \mid X \to x \in P,\ X \in N,\ x \in T^+\}$.

(2) $\delta_i([Y, j], a_j) = \{([X, j], y) \mid X \to Yy \in P,\ X, Y \in N,\ y \in T^*\}$ for $1 \leq j \leq k$.

(3) $\delta_i([X_j, j], a_{j+1}) = \{([X, i+1], x) \mid X \to x \in P,\ X \in N,\ x \in T^+\}$ for $1 \leq j \leq k - 1$.

(4) $\delta_i(q, a_j) = \phi$ otherwise $q \in Q$, $1 \leq j \leq k$.

M_i uses the input word to count steps and gives what G would deposit as output in a manner similar to the construction of Theorem 2.23. An output word is in $M_i(L_k)$ if and only if it is the result of a generation from the i-th initial production of G. Hence $M_i(L_k) = K_i$. Hence in this case as well $K_i \in \mathfrak{F}_k$.

We conclude $K_i \in \mathfrak{F}_k$ $1 \leq i \leq k$ and so $L = \bigcup_{i=1}^{k} K_i \in \mathfrak{F}_k$. Thus $\mathcal{L}_k^{\ell} \subseteq \mathfrak{F}_k$ and therefore $\mathcal{L}_k^{\ell} = \mathfrak{F}_k$.

COROLLARY 2.25: $\mathcal{L}_k^{\ell} = \mathcal{L}_k$.

COROLLARY 2.26: $\mathcal{L}_k$ is closed under mirror image.

PROOF: Let $L \in \mathcal{L}_k$ and $G = (N, T, S, P, k)$ be a k-rlg for L. We construct a k-llg $G^R = (N, T, S, P^R, k)$ for mi(L). P^R contains:

(1) $S \to mi(x)$ if $S \to x \in P$, $x \in T^*$.

(2) $S \to X_k X_{k-1} \ldots X_1$ if $S \to X_1 \ldots X_k \in P$, $X_i \in N$, $1 \leq i \leq k$.

(3) $X \to Y mi(y)$ if $X \to yY \in P$, $X, Y \in N$, $y \in T^*$.

(4) $X \to mi(x)$ if $X \to x \in P$, $X \in N$, $x \in T^+$.

It is easy to verify that $L(G^R) = mi(L)$.

§6. DECIDABILITY QUESTIONS

In this section we consider two decidability questions relating k-rlg's and the generated languages which have a positive answer. We recall that in Lemma 2.2 we showed membership problem is decidable for k-rlg's.

Let $G = (N, T, S, P, k)$ be a k-rlg. We recall that in Theorem 2.12 we defined a relation "$\rightarrow$" on N^k by $(X_1, \ldots, X_k) \rightarrow (Y_1, \ldots, Y_k)$, $X_i, Y_i \in N$ $1 \leq i \leq k$ iff there exist $X_i \rightarrow y_i Y_i \in P$, $y_i \in T^*$, $1 \leq i \leq k$.

<u>DEFINITION 2.7</u>: An <u>N-sequence</u> for G is a finite sequence of members of N^k $\mathcal{S} = (s_i)_{i=1}^{n}$ such that $s_i \rightarrow s_{i+1}$ $1 \leq i \leq n - 1$.

Note that we can always associate an N-sequence with a non-trivially generated word $x \in L(G)$. If D is a derivation of x by G we denote the associated N-sequence by $\mathcal{S}(D, x)$ and the i-th member of this sequence by $s_i(D, x)$. We call a repetition $s_i(D, x) = s_j(D, x)$ $j > i$ in an N-sequence associated with a word x "trivial" if there are no terminals deposited in intervening steps. We can now show that the "emptiness problem" is decidable for k-rlg's.

<u>THEOREM 2.27</u>: Given a k-rlg $G = (N, T, S, P, k)$ there is an algorithm to decide whether $L(G) = \phi$ or not.

<u>PROOF</u>: Since $L(G)$ is recursive by Lemma 2.2 we have only to give an upper bound for the shortest non-trivially generated word in $L(G)$. Suppose $\#(N) = \ell$ and $\max \{|x| \mid X \to x \in P, X \in N\} = m$.

<u>CLAIM</u>: $L(G) \neq \phi$ iff there exists $x \in L(G)$ such that $|x| \leq mk\ell^k + mk$ or there exists a production $S \to x \in P$ with $x \in T^*$.

<u>PROOF</u>: <u>if</u>: obvious.

<u>only if</u>: Suppose G has no rules of the form $S \to x$, $x \in T^*$ and there does not exist $x \in L(G)$ with $|x| \leq mk\ell^k + mk$, but that $L(G) \neq \phi$. This implies there exists a shortest $y \in L(G)$ with $|y| > mk\ell^k + mk$. There exists a derivation $D: S = P_0 \Longrightarrow P_1 \Longrightarrow \ldots \Longrightarrow P_{n+1} = y$ for y and an N-sequence $\mathcal{S}(D, y) = (s_i(D, y))_{i=1}^{n}$. We may suppose $\mathcal{S}(D, y)$ has no trivial repetitions (for if it has we may find a shorter derivation for y with no trivial repetitions). Since each application of k non-terminating productions can deposit at most

$(m - 1)k$ terminals, it is clear that $|P_r| \leq rmk$, $r \leq n$. Thus $|y| \leq (n + 1)mk$ and so $n \geq \ell^k$. Hence there must be a repetition (non-trivial!) in $\mathcal{S}(D, y)$, say $s_p(D, y) = s_q(D, y)$. Then $s_1(D, y) \to s_2(D, y) \to \ldots \to s_p(D, y) \to s_{q+1}(D, y) \to \ldots \to s_n(D, y)$ is an N-sequence associated to a word y' L(G) and since $s_p(D, y) = s_q(D, y)$ is non-trivial, we have $|y'| < |y|$ contradicting the minimality of $|y|$. Hence no such y exists and we condlude $L(G) = \phi$. This completes the claim and so we are done.

By a similar method we can show that the "finiteness-infiniteness problem" is decidable for k-rlg's.

THEOREM 2.28: Given a k-rlg $G = (N, T, S, P, k)$, there is an algorithm to decide whether or not $\#(L(G)) = \infty$.

PROOF: We again use the fact that L(G) is recursive. Let m and ℓ be as above and $p = mk\ell$ (cf. Theorem 2.5). We claim that L(G) is infinite iff there exists a non-trivially generated $x \in L(G)$ with $p \leq |x| \leq p + mk\ell^k$. If L(G) is not infinite there cannot exist $x \in L(G)$ with $|x| \geq p$ (otherwise by

Theorem 2.5 there are infinitely many words in $L(G)$). If $L(G)$ is infinite, then there exists a shortest $x \in L(G)$ with $|x| \geq p$. If $|x| > p + mk\ell^k$ an argument similar to that of Theorem 2.27 shows that we can find an $x' \in L(G)$ with $p \leq |x'| < |x|$ contradicting the minimality of $|x|$. Thus if $L(G)$ is infinite there exists $x \in L(G)$ with $p \leq |x| \leq p + mk\ell^k$.

CHAPTER 3

REGULATED REWRITING

§1. k-PARALLEL RIGHT-LINEAR WITH REGULAR CONTROL LANGUAGES.

In this chapter we add a control device to k-parallel right-linear grammars, namely a regular control language. We show that the language families generated are the same as both the k-tuple languages of Kuich and Maurer [5] with a right-linear restriction and the k-right-linear simple matrix languages of Ibarra [4].

We wish to define "control word" for a derivation by a k-rlg. Since productions are applied k at a time except in the initial step, the labelling of derivation steps must take this fact into account.

DEFINITION 3.1: Let $G = (N, T, S, P, k)$ be a k-rlg. A labelling of productions from G is a 1-1 correspondence Lab: $\overline{P} \longrightarrow \mathrm{Lab}(\overline{P})$ where Lab $(\overline{P})$ is a finite set of "labels" and

$$\overline{P} = \{S\to x \mid S\to x \in P\} \cup \{(X_1,\ldots,X_k)\to(y_1,\ldots,y_k) \mid X_i\to y_i \in P, i=1,\ldots,k\}.$$

DEFINITION 3.2: Let $G = (N, T, S, P, k)$ be a k-rlg and Lab $(\overline{P})$ a set of labels for productions from G. Let D be a derivation by G. Then u is a control word for D if one of the following holds (i) D is $Q_0 \Longrightarrow Q_1$, $u = a \in$ Lab $(\overline{P})$ and a is the label of the production applied in $Q_0 \Longrightarrow Q_1$, or (ii) D is $Q_0 \overset{*}{\Longrightarrow} Q_n \overset{*}{\Longrightarrow} Q_m$, $u = u_1 u_2$ and u_1 is the control word of $Q_0 \overset{*}{\Longrightarrow} Q_n$ and u_2 is the control word of $Q_n \overset{*}{\Longrightarrow} Q_m$.

With these definitions we can assign to a pair (D, x), where D is a derivation by G of x, a control word denoted $u(D, x)$.

DEFINITION 3.3: $L \subseteq T^*$ is a k-parallel right-linear with regular control language (k-rrll) iff there exists a k-rlg $G = (N, T, S, P, k)$, a labelling of productions from G Lab, and a regular language C over Lab $(\overline{P})$ such that $L = L(G, C) = \{x \in L(G) \mid$ there exists a derivation D for x, and $u \in C$ with $u = u(D, x)\}$.

We denote the family of k-rrl's by $\mathcal{R}_k$.

EXAMPLE 3.1: We consider the 2-rlg $G = (N, T, S, P, z)$ where $N = \{S, A, X, B, C\}$, $T = \{a, b, c\}$ and P contains:

$S \to AX$

$A \to aA \mid B$

$B \to bB \mid b$

$X \to X \mid C$

$C \to cC \mid c.$

It is easy to show that $L(G) = \{a^i b^j c^k \mid i + j \geq k,\ i,j,k \geq 1\}$. We give labels to production pairs which will be allowed:

$S \to AX$: e

$(A, X) \to (aA, X)$: a

$(A, X) \to (B, C)$: b

$(B, C) \to (bB, cC)$: c

$(B, C) \to (b, c)$: d.

Let $D = ea^*bc^*d$, then $L(G, D) = \{a^n b^m c^m \mid n, m \geq 1\}$ $L(G, D)$ is a 2-rrll, but apparently not a 2-rll.

Example 3.1 may be generalized to give $L_{k,r} = \{a^n a_1^m a_2^m \ldots a_k^m \mid n, m \geq 1\}$ which is a k-rrll, but apparently not a k-rll for $k \geq 2$. When $k = 1$ we have $\mathcal{L}_1 = \mathcal{R}_1$ (by Salomaa [9]). For $k > 1$ we have $\mathcal{L}_k \subseteq \mathcal{R}_k$ since given a k-rlg $G = (N, T, S, P, k)$ we may take $C = \text{Lab}(\bar{P})^*$ and then $L(G) = L(G, C)$.

The first result we shall need is that the families $\mathcal{R}_k$ form a hierarchy.

<u>THEOREM 3.1</u>: For all $k \geq 1$, $\mathcal{R}_k \subseteq \mathcal{R}_{k+1}$.

<u>PROOF</u>: The method is to construct a k+1-rlg as in Theorem 2.3 and to construct a new control language.

Let $L \in \mathcal{R}_k$, $G = (N, T, S, P, k)$ a k-rlg and $M = (Q, \Sigma, \delta, q_0, F)$ an fsa such that $L = L(G, T(M))$ where Σ is a set of labels for productions of G. We apply the construction of Theorem 2.3 to give a k+1-rlg G' such that L(G) = L(G'). We will construct an nfsa M' such that L = L(G', T(M')). The idea is to associate to a control word of a derivation by G a control word of a derivation by G' in such a way that the new control word is accepted by M' iff the old word was accepted by M. In view of the fact that, except for a finite number of short words, derivations procede in G' in essentially the same way as they did in G, we can construct M'. (Note that $G' = (N', T, S, P', k + 1)$.)

Let $M' = (Q', \Sigma', \delta', q_0, F')$ where $Q' = Q \cup \{q_1\}$, $q_1 \notin Q$, Σ' is a set of labels for productions of G', $F' = F \cup \{q_1\}$ and δ' is constructed as follows:

(1) $\delta'(q_0, a) = q_1$ if $S \overset{a}{\rightarrow} x \in P'$, $x \in T^*$ (we again use the notation $X \overset{a}{\rightarrow} x$ to mean that a is the label for the production $X \rightarrow x$.)

(2) $\delta'(q_0, a) = q'$ if $S \overset{a}{\rightarrow} X_1 \ldots Y_c [X_i, c] \ldots X_k$ in G' and $\delta(q_0, b) = q'$ where $S \overset{b}{\rightarrow} X_1 \ldots X_i \ldots X_k$ in G.

(3) $\delta'(q, a) = q'$ if $(X_1, \ldots, Y_c, [X_i, c], \ldots, X_k) \overset{a}{\rightarrow}$ $(y, Y_1, \ldots, Y_c, \bar{y}, \ldots, y_k Y_k)$ where $y_j \in T^*$, $Y_j \in N$ $1 \le j \le i - 1$ and $i + 1 \le j \le k$, where either $\bar{y} = yY$, $y \in T^*$, $y \in N$ or $\bar{y} = [Y, c]$ and $(X_1, \ldots, X_k) \overset{b}{\rightarrow} (y_1 Y_1, \ldots, \bar{\bar{y}}, \ldots, y_k Y_k)$ in G where either $\bar{\bar{y}} = cyY$ or $\bar{\bar{y}} = Y$ and $\delta(q, b) = q'$.

(4) $\delta'(q, a) = q'$ if $(X_1, \ldots, Y_c, X_i, \ldots, X_k) \overset{a}{\rightarrow}$ $(y_1 Y_1, \ldots, Y_c, y_i Y_i, \ldots, y_k Y_k)$ where $(X_1, \ldots, X_k) \overset{b}{\rightarrow} (y_1 Y_1, \ldots, y_k Y_k)$ in G and $\delta(q, b) = q'$.

(5) $\delta'(q, a) = q'$ if $(x_1, \ldots, Y_c, X_i, \ldots, X_k) \overset{a}{\rightarrow}$ $(x_1, \ldots, c, x_i, \ldots, x_k)$ where $(X_1, \ldots, X_i, \ldots, X_k) \overset{b}{\rightarrow}$ $(x_1, \ldots, cx_i, \ldots, x_k)$ in G and $\delta(q, b) = q'$.

(6) $\delta'(q, a) = \phi$ otherwise.

By the construction, $x \in L(G', T(M'))$ iff $x \in L(G, T(M))$. Hence $L \in \mathcal{R}_{k+1}$ and we conclude $\mathcal{R}_k \subseteq \mathcal{R}_{k+1}$.

We also need

<u>LEMMA 3.2</u>: $\mathcal{R}_k$ is closed under union for all $k \geq 1$.

<u>PROOF</u>: Let $L_1, L_2 \in \mathcal{R}_k$, say $L_1 = L(G_1, C_1)$ and $L_2 = L(G_2, C_2)$. We construct G_3 so that $L(G_3) = L(G_1) \cup L(G_2)$ (as in Lemma 2.10). We label productions of G_3 by the labels of the corresponding productions of G_1 and G_2. Then $L_1 \cup L_2 = L(G_3, C_1 \cup C_2)$.

§2. RIGHT-LINEAR TUPLE LANGUAGES.

Kuich and Maurer [5] have defined "Tuple Languages" with context-free productions. We specialise this notion to allow only right-linear productions.

DEFINITION 3.4: Let T be a finite set of terminal symbols. Then we denote $T^* \times \ldots \times T^*$ (k times) by T^*_k, the set of k-tuples of words over T. Let $c_i: T^*_k \rightarrow T^*$ be the homomorphism defined by $c_i((x_1, \ldots, x_k)) = x_i$ for $1 \leq i \leq k$. If $x, y \in T^*_k$ then $xy = (c_1(x)c(y), \ldots, c_k(x)c_k(y))$. We define $\mu: T^*_k \rightarrow T^*$ by $\mu(z) = c_1(z)c_2(z)\ldots c_k(z)$, $z \in T^*_k$. Denote the k-tuple of ε's by ε.

DEFINITION 3.5: A right-linear k-tuple grammar (k-tlg) is a 5-tuple $G = (k, N, T, S, P)$ where

(1) $k \geq 1$ is an integer.

(2) N is a finite set (of non-terminal symbols).

(3) T is a finite set (of terminal symbols) with $T \cap N = \phi$.

(4) $S \in N$.

(5) P is a finite set of productions of the form $X \rightarrow x$ with $X \in N$ and $x \in T^*_k N \cup T^*_k$.

The "yields" relation $\Longrightarrow$ for words over $N \cup T_k^*$ is defined by $x \Longrightarrow y$ iff $x = uXv$, $y = uzv$ and $X \to z \in P$.

DEFINITION 3.6: $L \subseteq T^*$ is a right linear k-tuple language (k-tll) iff there exists a k-tlg $G = (k,N,T,S,P)$ such that $L = L(G) = \{\mu(x) \mid S \overset{*}{\Longrightarrow} x,\ x \in T_k^*\}$.

We denote the family of right-linear k-tuple languages by $\mathcal{T}_k$. We observe immediately that $\mathcal{T}_1 = \mathcal{L}_1$.

THEOREM 3.3: For all $k \geq 1$, $\mathcal{T}_k = \mathcal{R}_k$.

PROOF: CLAIM 1: $\mathcal{R}_k \subseteq \mathcal{T}_k$.

PROOF: Let $L \in \mathcal{R}_k$, then there exists a k-rlg $G = (N, T, S, P, k)$ and an fsa $M = (Q, \Sigma, \delta, q_0, F)$ such that Σ is a set of labels for productions of G and $L = L(G, T(M))$. We construct a k-tlg $G' = (k, N', T, [S, q_0], P')$ where $N' = \{[S,q_0]\} \cup (N^k \times Q)$ and P' contains:

(1) $[S, q_0] \to (x, \varepsilon, \ldots, \varepsilon)$ if $S \overset{a}{\to} x \in P$, $x \in T^*$ and $\delta(q_0, a) \in F$.

(2) $[S, q_0] \to [X_1, \ldots, X_k, q']$ if $S \overset{a}{\to} X_1 \ldots X_k \in P$, $X_i \in N$ $1 \leq i \leq k$ and $\delta(q_0, a) = q'$.

(3) $[X_1, \ldots, X_k, q] \to (y_1, \ldots, y_k)[Y_1, \ldots, Y_k, q']$ if $(X_1, \ldots, X_k) \overset{a}{\Rightarrow} (y_1Y_1, \ldots, y_kY_k)$ with $X_i, Y_i \in N$, $(y_1, \ldots, y_k) \in T_k^*$ and $\delta(q, a) = q'$.

(4) $[X_1, \ldots, X_k, q] \to (y_1, \ldots, y_k)$ if $(y_1, \ldots, y_k) \in T_k^*$, $(X_1, \ldots, X_k) \overset{a}{\Rightarrow} (y_1, \ldots, y_k)$ and $\delta(q, a) \in F$.

Now G' is a k-tlg which imitates a derivation by G while keeping track of the state of M in the last component of its non-terminals. A derivation by G' is allowed to terminate iff the control word of the corresponding derivation by G is in T(M). Thus $L = L(G, T(M)) = L(G') \in \mathcal{T}_k$ and we have $\mathcal{R}_k \subseteq \mathcal{T}_k$.

<u>CLAIM 2</u>: $\mathcal{T}_k \subseteq \mathcal{R}_k$.

<u>PROOF</u>: We use a technique similar to that used in Theorem 2.11. First, let $L \in \mathcal{T}_k$, say $L = L(G)$ for the k-tlg $G = (k, N, T, S, P)$. We again consider the sets of functions $\bar{\mathcal{R}}_i$, and note that the notion $\pi_i(D, x)$ for a derivation D of a word $x \in L(G)$ makes sense for $1 \leq i \leq k$. We define

$L_i^{\varphi} = \{x \in L \mid \pi_j(D,x) \neq \varepsilon \text{ all } j \in \operatorname{im}\varphi, \pi_j(D, x) = \varepsilon \text{ otherwise}\}$,

$L_i = \bigcup_{\varphi \in \bar{\mathcal{R}}_i} L_i^{\varphi}$. Then

$$L = \begin{cases} \bigcup_{i=1}^{k} L_i & \text{if } \varepsilon \notin L \\ \bigcup_{i=1}^{k} L_i \cup \{\varepsilon\} & \text{otherwise.} \end{cases}$$

Using the method used in Theorem 2.11 to construct the i-rlg G_i^φ, we construct an i-tlg $G_i^\varphi = (i, N_i^\varphi, T, S, P_i^\varphi)$ for L_i^φ with the property that if $X \to (x_1, \ldots, x_i) \in P_i^\varphi, x_j \in T^*$ and $X \neq S$, we have $x_j \neq \varepsilon$ $1 \leq j \leq i$. Using G_i^φ we will show $L_i^\varphi \in \mathcal{R}_i$ by constructing an i-rlg $\overline{G}_i = (\overline{N}_i, T, S', \overline{P}_i, i)$ and a control language. $\overline{N}_i^\varphi = N_i^\varphi \times \{1, \ldots, i\} \cup \{S'\}$. $\overline{P}_i^\varphi$ contains:

(1) $S' \to x_1 x_2 \ldots x_i$ if $S \to (x_1, \ldots, x_i) \in P_i^\varphi$.

(2) $S' \to [S, 1] \ldots [S, i]$.

(3) $[X, j] \to y_j [Y, j]$ for $1 \leq j \leq i$ if $X \to (y_1, \ldots, y_i) Y \in P_i^\varphi$, $X, Y \in N_i^\varphi$, $y_j \in T^*$, $1 \leq j \leq i$.

(4) $[X, j] \to x_j$ for $1 \leq j \leq i$ if $X \to (x_1, \ldots, x_i) \in P_i^\varphi$, $X \in N_i^\varphi$ and $x_j \in T^+$ $1 \leq j \leq i$.

We now suppose a set of labels for productions of $\overline{G}_i^\varphi$ has been introduced and define

$$A = \{a \mid (X, \ldots, X) \overset{a}{\Rightarrow} (y_1 Y, \ldots, y_i Y) \text{ and } X \to (y_1, \ldots, y_i) Y \in P_i^\varphi\}$$
$$\cup \{a \mid (X, \ldots, X) \overset{a}{\Rightarrow} (x_1, \ldots, x_i),\ X \to (x_1, \ldots, x_i) \in P_i^\varphi, x_j \in T^+\}$$

$$B = \{b \mid S' \overset{b}{\Rightarrow} x,\ x \in T^*\}$$

and we suppose c is the label for $S' \to [S, 1] \ldots [S, i]$. Define $C = B \cup cA^*$ which is a regular language over the set of labels for productions of $\overline{G}_i^{\varphi}$. Now we have $x \in L(\overline{G}_i^{\varphi}, C)$ iff x has a derivation D by $\overline{G}_i^{\varphi}$ with a control word in C iff D is either trivial, or it uses productions after the initial one with labels from A iff there is a derivation of x by G_i^{φ} iff $x \in L_i^{\varphi}$. Thus $L_i^{\varphi} = L(\overline{G}_i^{\varphi}, C)$ and so $L_i^{\varphi} \in \mathcal{R}_i$. By Lemma 3.2 $L_i = \bigcup_{\varphi \in \bar{\alpha}_i} L_i^{\varphi} \in \mathcal{R}_i$ $1 \leq i \leq k$ and by Theorem 3.1 $L_i \in \mathcal{R}_k$, so we have $L \in \mathcal{R}_k$. Thus $\mathcal{J}_k \subseteq \mathcal{R}_k$.

Combining the two results we have $\mathcal{J}_k = \mathcal{R}_k$.

§3. RIGHT-LINEAR SIMPLE MATRIX LANGUAGES

Ibarra [4] has introduced the notions of simple matrix language and right-linear simple matrix language and studied their properties extensively. In this section we relate the second of these concepts to the families $\mathcal{R}_k$.

DEFINITION 3.7: A k-right-linear simple matrix grammar (k-rlmg) is a (k+3)-tuple $G = (N_1, \ldots, N_k, T, S, P)$ where

(1) $N_1, N_2, \ldots, N_k$ are pairwise disjoint finite sets of non-terminals.

(2) T is a finite set of terminals and $T \cap N_i = \phi$ $1 \leq i \leq k$.

(3) S is the start symbol and $S \notin \bigcup_{i=1}^{k} N_i \cup T$.

(4) P is a finite set of matrix rewriting rules of the form

(i) $[S \to x]$, $x \in T^*$

(ii) $[S \to x_{11}X_{11}x_{12}X_{12} \cdots x_{1n}X_{1n} \cdots x_{k1}X_{k1} \cdots x_{kn}X_{kn}y]$ where $n \geq 1$, $y \in T^*$ and $1 \leq i \leq k$, $1 \leq j \leq n$ $X_{ij} \in N_i$ and $x_{ij} \in T^*$.

(iv) $[X_1 \to y_1 Y_1, \ldots, X_k \to y_k Y_k]$ where $X_i, Y_i \in N_i$ and $y_i \in T^*$ $1 \leq i \leq k$.

DEFINITION 3.8: Let $G = (N_1, \ldots, N_k, T, S, P)$ be a k-rlmg. We define the yield relation for $x, y \in (\bigcup_{i=1}^{k} N_i \cup T \cup \{S\})^*$ by $x \Longrightarrow y$ iff

(1) $x = S$ and $[S \to y] \in P$ or,

(2) There exist $y_1, \ldots, y_k \in T^*$, $w_1, \ldots, w_k, z_1, \ldots, z_k$ with $w_i, z_i \in (N_i \cup T)^*$ and $X_1, \ldots, X_k$ with $X_i \in N_i$ such that $x = y_1 X_1 z_1 \ldots y_k X_k z_k$, $y = y_1 w_1 z_1 \ldots y_k w_k z_k$ and $[X_1 \to w_1, \ldots, X_k \to w_k] \in P$. $\overset{*}{\Longrightarrow}$ is the reflexive transitive closure of $\Longrightarrow$. (Note that this is a "leftmost" derivation.)

DEFINITION 3.9: $L \subseteq T^*$ is a <u>k-right-linear simple matrix language (k-rlml)</u> iff there exists a k-rlmg $G = (N_1, \ldots, N_k, T, S, P)$ such that $L = L(G) = \{x \in T^* \mid S \overset{*}{\Longrightarrow} x\}$.

We denote the family of k-rlml's by $\mathcal{M}_k$. Before we give the main result of this section we need

LEMMA 3.4: If $L \in \mathcal{M}_k$ then L can be generated by a k-rlmg having rewriting rules only of the forms (i), (iii), (iv) and (ii'): $[S \to x_1 X_1 x_2 X_2 \ldots x_k X_k y]$ with $x_i, y \in T^*$ and $X_i \in N_i$ $1 \leq i \leq k$.

PROOF: Let $G = (N_1, \ldots, N_k, T, S, P)$ be a k-rlmg for L. If G has rewriting rules only of forms (i), (iii), (iv) and (ii') we are done. Otherwise let

$m = \max\{\ell \mid [S \to x_{11}X_{11} \ldots x_{1\ell}X_{1\ell} \ldots x_{k\ell}X_{k\ell}y] \in P\}$.

Let $\overline{N}_i = N_i \cup N_i^2 \cup \ldots \cup N_i^m \quad 1 \leq i \leq k$ and
$\overline{G} = (\overline{N}_1, \ldots, \overline{N}_k, T, S, \overline{P})$ where $\overline{P}$ contains:

(1) $[S \to x]$ if $[S \to x] \in P$ and $x \in T^*$.

(2) $[S \to x_{11}[X_{11}, X_{12}, \ldots, X_{1\ell}]x_{21} \ldots x_{k1}[X_{k1}, \ldots, X_{k\ell}]y]$ if $[S \to x_{11}X_{11} \ldots x_{1\ell}X_{1\ell} \ldots x_{k\ell}X_{k\ell}y] \in P$ where $y, x_{ij} \in T^*$ and $X_{ij} \in N_i \quad 1 \leq i \leq k,\ 1 \leq j \leq \ell$.

(3) $[[Z_1, X_{11}, \ldots, X_{1j}] \to y_1[Y_1, X_{11}, \ldots, X_{1j}], \ldots [Z_k, X_{k1}, \ldots, X_{kj}] \to y_k[Y_k, X_{k1}, \ldots, X_{kj}]]$ if $[Z_1 \to y_1Y_1, \ldots, Z_k \to y_kY_k] \in P$, $j \in \{1, \ldots, m-1\}$ and $X_{iq} \in N_i \quad 0 \leq q \leq j$.

(4) $[[Z_{1,j-1}, X_{1j}, \ldots X_{1\ell}] \to w_1x_{1j}[X_{1j}, \ldots, X_{1}\ ], \ldots, [Z_{k,j-1}, X_{kj}, \ldots, X_{k\ell}] \to w_kx_{kj}[X_{kj}, \ldots, X_{k\ell}]]$ if $[Z_{1,j-1} \to w_1, \ldots, Z_{k,j-1} \to w_k] \in P$ and $[S \to x_{11}X_{11} \ldots x_{1j}X_{1j} \ldots x_{1\ell}X_{1\ell} \ldots x_{k\ell}X_{k\ell}] \in P$ where $x_{ip}, x_i \in T^*, Z_{i,j-1}, X_{ip} \in N_i \quad 1 \leq i \leq k,\ 1 \leq p \leq \ell$.

(5) $[X_1 \to x_1, \ldots, X_k \to x_k]$ if $[X_1 \to x_1, \ldots, X_k \to x_k] \in P$ and $X_i \in N_i,\ x_i \in T^*,\ 1 \leq i \leq k$.

$\overline{G}$ simply imitates a derivation by G while keeping track of any unused non-terminals which resulted from its initial production in the components of its non-terminals. We conclude $L(\overline{G}) = L(G)$ and $\overline{G}$ has only productions of the desired types.

<u>THEOREM 3.5</u>: For all $k \geq 1$, $\mathfrak{T}_k = \mathfrak{M}_k$.

<u>PROOF</u>: Let $L \in \mathfrak{T}_k$ and $G = (k, N, T, S, P)$ a k-tlg for L. Let $N_i = \{[X, i] \mid X \in N - \{S\}\}$ and $\overline{G} = (N_1, \ldots, N_k, T, S, \overline{P})$ a k-rlmg where $\overline{P}$ contains:

(1) $[S \rightarrow w_1 w_2 \ldots w_k]$ if $S \rightarrow (w_1, \ldots, w_k) \in P, w_i \in T^*$, $1 \leq i \leq k$.

(2) $[S \rightarrow x_1 [X, 1] \ldots x_k [X, k]]$ if $S \rightarrow (x_1, \ldots, x_k) X \in P$, $X \in N$.

(3) $[[X, 1] \rightarrow y_1 [Y, 1], \ldots, [X, k] \rightarrow y_k [Y, k]]$ if $X \rightarrow (y_1, \ldots, y_k) Y \in P$ where $X, Y \in N$, $y_i \in T^*$.

(4) $[[X, 1] \rightarrow x_1, \ldots, [X, k] \rightarrow x_k]$ if $X \rightarrow (x_1, \ldots, x_k) \in P$, $X \in N$, $x_i \in T^*$.

Clearly $L(\overline{G}) = L(G) = L$. Hence $L \in \mathfrak{M}_k$ and we have $\mathfrak{T}_k \subseteq \mathfrak{M}_k$.

To show the reverse inclusion let $L \in \mathfrak{M}_k$ and $G = (N_1, \ldots, N_k, T, S, P)$ be a k-rlmg for L normalized as in Lemma 3.4. Let $W = \{y \in T^* \mid [S \rightarrow x_1 X_1 \ldots x_k X_k y] \in P\}$

and $\overline{N} = \{S\} \cup (N_1 \times N_2 \times \ldots \times N_k \times W)$. Define $\overline{G} = (k, \overline{N}, T, S, \overline{P})$ where $\overline{P}$ contains:

(1) $S \to (w, \varepsilon, \ldots, \varepsilon)$ if $[S \to w] \in P$.

(2) $S \to (x_1, \ldots, x_k)[X_1, \ldots, X_k, y]$ if $[S \to x_1 X_1 \ldots x_k X_k, y] \in P$ where $y, x_i \in T^*$, $X_i \in N_i$ $1 \leq i \leq k$.

(3) $[X_1, \ldots, X_k, y] \to (y_1, \ldots, y_k)[Y_1, \ldots, Y_k, y]$ if $[X_1 \to y_1 Y_1, \ldots, X_k \to y_k Y_k] \in P$, $y_i \in T^*$, $y \in W$, $X_i, Y_i \in N_i$.

(4) $[X_1, \ldots, X_k, y] \to (x_1, \ldots, x_k y)$ if $[X_1 \to x_1, \ldots, X_k \to x_k] \in P$, $y \in W$, $x_i \in T^*$, $X_i \in N_i$ $1 \leq i \leq k$.

Now $\overline{G}$ is clearly a k-tlg such that $L(\overline{G}) = L(G) = L$. Hence $L \in \mathcal{T}_k$ and $\mathcal{M}_k \subseteq \mathcal{T}_k$. This completes the proof.

<u>COROLLARY 3.6</u>: For all $k \geq 1$, $\mathcal{M}_k = \mathcal{R}_k$.

We now note that we could alter the definition of k-tlg to demand that if $X \to (x_1, \ldots, x_k)$ is a production and $X \neq S$ then $x_i \neq \varepsilon$ $1 \leq i \leq k$. Similarly, in the definition of k-rlmg we could demand that if $[X_1 \to x_1, \ldots, X_k \to x_k]$ is a rewriting rule, then $x_i \neq \varepsilon$ $1 \leq i \leq k$. We denote the family of languages generated by k-rlmg's with this restriction by $\mathcal{M}_k^{\varepsilon}$, and similarly define $\mathcal{T}_k^{\varepsilon}$. Now we can extend the definition of k-rrlg to allow the base grammar to be an ε-k-rlg and we denote the family of languages so obtained by $\mathcal{R}_k^{\varepsilon}$.

<u>COROLLARY 3.7</u>: For all $k \geq 1$

(i) $\mathcal{T}_k = \mathcal{T}_k^{\varepsilon}$ and $\mathcal{M}_k = \mathcal{M}_k^{\varepsilon}$.

(ii) $\mathcal{R}_k = \mathcal{R}_k^{\varepsilon}$.

<u>PROOF</u>: (i) It is clear that $\mathcal{T}_k^{\varepsilon} \subseteq \mathcal{T}_k$. Since $\mathcal{T}_k = \mathcal{R}_k$, we have the reverse inclusion when we note that in the construction of a right-linear k-tuple grammar from a k-rrll (Theorem 3.3) no terminating k-tuples contain ε's.

The second equality follows from Corollary 3.6 by a similar argument.

(ii) The family $\mathcal{M}_k$ is closed under homomorphism, hence so is $\mathcal{R}_k$. Now let $L \in \mathcal{R}_k^{\varepsilon}$ with a base grammar $G = (N, T, S, P, k)$. Let $a \notin T$ then the grammar obtained by substituting $X \rightarrow a$ for all rules of the form $X \rightarrow \varepsilon$ with $X \neq S$ is a k-rlg. Let L_a be the language obtained by making this substitution and using the same control language. Then $L_a \in \mathcal{R}_k$. Define $h: T \cup \{a\} \rightarrow T^*$ by $h|T = id_T$ and $h(a) = \varepsilon$. Clearly $L = h(L_a)$, so $L \in \mathcal{R}_k$. Therefore $\mathcal{R}_k^{\varepsilon} \subseteq \mathcal{R}_k$. The reverse inclusion is obvious and the result follows.

<u>COROLLARY 3.8</u>: There exist context-free languages which are not in $\mathcal{R}_k$ for any k, hence not in $\mathcal{L}_k$ for any k, or in $\mathcal{L}$.

PROOF: This is from Corollary 3.6 and Theorem 4.7 of Ibarra [4].

§4. ANOTHER RESTRICTION ON DERIVATIONS.

In this section we define another form of regulated rewriting for k-rlg's. As is the case for context-free grammars, periodically time varying k-rlg's and k-rlg's with regular control have the same generative capacity.

DEFINITION 3.10: A k-parallel right linear periodically time-varying grammar (k-rlg) is a pair (G, φ) where $G = (N, T, S, P, k)$ is a k-rlg and $\varphi : \mathbb{N} \to 2^{\overline{P}}$ ($\overline{P}$ as in Definition 3.1!) with the property that there exists $p \in \mathbb{N}$ such that $\varphi(j + p) = \varphi(j)$ for all $j \in \mathbb{N}$.

DEFINITION 3.11: Let (G, φ) be a k-rlpg where $G = (N, T, S, P, k)$. We define the yields relation on pairs from $(N \cup T^*) \times \mathbb{N}$ by $(P, j_1) \Longrightarrow (Q, j_2)$ iff either (1) $j_1 = 1$, $j_2 = 2$, $P = S$ and $S \to Q \in \varphi(1)$ or (2) $j_2 = j_1 + 1$, $P = z_1 X_1 \ldots z_k X_k$ and $Q = z_1 y_1 \ldots z_k y_k$ with $z_i \in T^*$, $X_i \in N$ $1 \leq i \leq k$ and $(X_1, \ldots, X_k) \to (y_1, \ldots, y_k) \in \varphi(j_1)$.

DEFINITION 3.12: $L \subseteq T^*$ is a k-parallel right-linear periodically time-varying language (k-rlpl) if there exists a k-rlpg (G, φ) where $G = (N, T, S, P, k)$

such that $L = L(G, \varphi) = \{x \in T^* \mid (S, 1) \stackrel{*}{\Longrightarrow} (x, j)$ for some $j \in \mathbb{N}\}$.

We denote the family of k-rlpl's by $\mathcal{V}_k$. Since the methods used to show the main result of this section have been developed above, and since they involve somewhat lengthy constructions, we simply state the result and sketch its proof.

<u>THEOREM 3.8</u>: For all $k \geq 1$, $\mathcal{R}_k = \mathcal{V}_k$.

<u>PROOF</u>: The first step is to show $\mathcal{V}_k \subseteq \mathcal{V}_{k+1}$ and $\mathcal{V}_k$ is closed under union for all $k \geq 1$. This is achieved by the methods of Theorem 3.1 and Lemma 3.2.

Next we show $\mathcal{J}_k \subseteq \mathcal{V}_k$. Given $L = L(G) \in \mathcal{J}_k$ it is easy to construct a k-rlg G_1 and φ with period 1 so that $L = L(G_1, \varphi)$. Finally we show $\mathcal{V}_k \subseteq \mathcal{R}_k$. Given $L = L(G, \varphi) \in \mathcal{V}_k$, we define an fsa which counts modulo p and accepts any control word of a derivation by G such that at the i-th step the productions used form a member of $\varphi(i)$.

BIBLIOGRAPHY

1. Ginsburg, S., Review of "The Structure Generating Function and Entropy of Tuple Languages" by Kuich, W. and Maurer, H., Computing Reviews, 13(2), 1972, [22, 680].

2. Greibach, S. A., and Hoproft, J. E., Scattered Context Grammars, Journal of Computer and System Sciences, 3 (1969), 233-247.

3. Hoproft, J. E., and Ullman, J. D., Formal Languages and their Relation to Automata, Addison Wesley (1969).

4. Ibarra, O. H., Simple Matrix Languages, Information and Control 17 (1970), 359-394.

5. Maurer, H., and Kuich, W., Tuple Languages, Proceedings of the A.C.M. International Computing Symposium, 1970, Bonn, 882-891.

6. Rajlich, V., Absolutely Parallel Grammars and Two-Way Deterministic Finite State Transducers, Proceedings of the Third SIGACT Conference, 1971, 132-137.

7. Rozenberg, G. and Doucet, P., On 0-L Languages, Information and Control, 19 (1971), 302-318.

8. Salomaa, A., Formal Languages, accepted for publication by Academic Press.

9. Salomaa, A., On Grammars with Restricted Use of Productions, Ann. Academiae Scientiarum Fennicæ, Series A, 454, 1969.

10. Siromoney, R., On Equal Matrix Languages, Information and Control, 14 (1969), 135-151.

11. Wood, D., Bibliography 23, Formal Language and Automata Theory, Computing Reviews, 11 (7), 1970, pp. 417-430.