ITERATION THEOREMS FOR FAMILIES OF STRICT DETERMINISTIC LANGUAGES

K. N. King*

Computer Science Division
Department of Electrical Engineering
and Computer Sciences

University of California
Berkeley, CA 94720

4/11/78

*The author is a National Science Foundation Graduate Fellow. This research was supported in part by the National Science Foundation under Grant MCS74-07636-A01.

Abstract. Two iteration theorems, one for strict deterministic languages of degree n, the other for simple deterministic languages, are presented. Examples demonstrating the use of these theorems are also given.

1. Introduction

Recent papers have extended the theorem of Bar-Hillel, Perles, and Shamir [1], and its refinement by Ogden [10,11], to apply to various classes of deterministic context-free languages. We refer to theorems modelled on the Bar-Hillel result as "iteration theorems." In [8], Harrison and Havel presented an iteration theorem for general deterministic context-free languages, Boasson [4] has given one for deterministic one-counter languages, and recently Beatty [2,3] has established two iteration theorems for LL(k) languages.

We introduce two more iteration theorems, each for a family of strict deterministic languages. The first theorem is for the family of strict deterministic languages of degree n, for any $n \geq 1$. Harrison and Havel [7] introduced these families and showed that they formed a hierarchy of strict deterministic languages. The second theorem is for the family of simple deterministic languages, which was defined by Korenjak and Hopcroft [9]. Our first iteration theorem is also applicable to the family of simple deterministic languages, since every such language is strict deterministic of degree 1. However, the second iteration theorem is stronger than the first for this special family of languages.

In Section 2, we define the families of languages to be studied and introduce notation for dealing with trees. Section 3 lists several lemmas needed in subsequent arguments. Section 4 reviews some previous iteration theorems. In Section 5, we prove an iteration theorem for strict deterministic languages of degree

n. Finally, in Section 6, we establish an iteration theorem for simple deterministic languages and use it to show that the family of simple deterministic languages is properly included in the class of prefix-free LL(1) languages.

2. Definitions

We first define some specialized terminology for discussing strings. Let Σ be an alphabet (a finite set of symbols). For $x,y \in \Sigma^*$, we say that y is a <u>prefix</u> of x if there exists $z \in \Sigma^*$ such that $x = yz$. If y is a prefix of x and $y \neq x$, then y is a <u>proper prefix</u> of x. A set of strings L is said to be <u>prefix-free</u> if $x, xy \in L$ implies[1] $y = \Lambda$ (i.e., no string in L is a proper prefix of any string in L). Let $w \in \Sigma^*$. We denote by ${}^{(n)}w$ the prefix of w of length[2] $\min\{n, \lg(w)\}$. A sequence of strings $(w_1, w_2, \ldots, w_m)$ is said to be a <u>factorization</u> of w if $w = w_1 w_2 \cdots w_m$. An integer i such that $1 \leq i \leq \lg(w)$ is called a <u>position</u> in w. By choosing some subset K of $\{1, \ldots, \lg(w)\}$, we specify a set of <u>distinguished positions</u> within w. For any set K of distinguished positions, a factorization $\phi = (w_1, \ldots, w_m)$ of w induces a partition $K/\phi = \{K_1, \ldots, K_m\}$ of K, where

$$K_i = \{k \in K \mid \lg(w_1 \cdots w_{i-1}) < k \leq \lg(w_1 \cdots w_i)\}.$$

We now turn to the definition of various types of context-free grammars. The reader is assumed to be familiar with the standard definition of context-free grammar (see, for example, [5]).

Let $G = (V, \Sigma, P, S)$ be a context-free grammar. G is said to be <u>reduced</u> if either $P = \emptyset$ or for each $A \in V$, there exists $\alpha, \beta \in V^*$, $w \in \Sigma^*$ such that $S \Rightarrow^* \alpha A \beta \Rightarrow^* w$. G is in <u>Greibach normal form</u> if every rule in P is of the form $A \rightarrow a\alpha$ for $a \in \Sigma$, $\alpha \in V^*$. (Note that this definition, unlike the standard definition of Greibach normal form, prevents Λ from being in L(G).)

Let $G = (V, \Sigma, P, S)$ be a context-free grammar and let π be a partition of V. We say that π is <u>strict</u> if

1. $\Sigma \in \pi$ and
2. for all $A, A' \in V - \Sigma$ and $\alpha, \beta, \beta' \in V^*$, if $A \rightarrow \alpha\beta$, $A' \rightarrow \alpha\beta'$ are in P and $A \equiv A' \pmod{\pi}$ then either
   (i) both $\beta, \beta' \neq \Lambda$ and ${}^{(1)}\beta \equiv {}^{(1)}\beta' \pmod{\pi}$, or
   (ii) $\beta = \beta' = \Lambda$ and $A = A'$.

If there exists a strict partition π of V, then G is said to be <u>strict</u> <u>deterministic</u>. A language L is strict deterministic if there exists a strict deterministic grammar G such that $L = L(G)$. For π a strict partition, we define[3]

$$\|\pi\| = \max_{V_i \in \pi - \{\Sigma\}} |V_i|.$$

If G is a strict deterministic grammar, the <u>degree</u> of G is

$$\underline{\text{deg}}(G) = \min\{\|\pi\| \mid \pi \text{ is a strict partition of } G\}.$$

For L a strict deterministic language, the degree of L is

$$\underline{\text{deg}}(L) = \min\{\underline{\text{deg}}(G) \mid G \text{ is strict deterministic and } L(G) = L\}.$$

A context-free grammar $G = (V, \Sigma, P, S)$ in Greibach normal form is <u>simple</u> <u>deterministic</u> if $A \rightarrow a\alpha$, $A \rightarrow a\beta$ in P implies $\alpha = \beta$, for all $A \in V - \Sigma$, $a \in \Sigma$, $\alpha, \beta \in V^*$. (Note that every simple deterministic grammar is strict deterministic of degree 1.) A language L is simple deterministic if $L = L(G)$ for some simple deterministic grammar G.

We now define a number of terms concerning trees. Our definitions come from [3], in which they are presented in more detail.

A <u>tree</u> $T = (V, E)$ is a connected dag (directed acyclic graph) in which every <u>node</u> (element of V) has exactly one entering <u>edge</u> (element of E), except for one node, denoted by rtn(T), which has no entering edges. We call rtn(T) the <u>root node</u> of T.

The set E defines the <u>immediate descendancy relation</u> on $V \times V$. The relation is written $x \Gamma y$, and we say that "x has immediate descendant y," and "y has parent x." The transitive closure of Γ is Γ^+, and the reflexive transitive closure of Γ is Γ^*.

The trees that we will be considering are ordered trees; that is, the immediate descendants of each node are ordered by some relation $\sqcap$. Thus, if $y_1, y_2, \ldots, y_r$ are the immediate descendants of a node in left-to-right order, then $y_1 \sqcap y_2 \sqcap \ldots \sqcap y_r$.

If $p \Gamma y$ and there is no node x such that $x \sqcap y$, then we write $p \Gamma_L y$. Similarly, if $p \Gamma x$ and there is no y such that $x \sqcap y$, we write $p \Gamma_R x$. We define the relation $\llcorner$ by

$$\llcorner = (\Gamma_R^{-1})^* \sqcap (\Gamma_L)^*.$$

Again, $\llcorner^+$ is the transitive closure of $\llcorner$, and $\llcorner^*$ is the reflexive transitive closure of $\llcorner$.

A sequence $(x_1\ x_2 \cdots x_m)$ of nodes in a tree T is a <u>maximal left-to-right sequence</u> if $x_1 \llcorner x_2 \llcorner \cdots \llcorner x_m$ and the sequence cannot be extended. Those nodes of T with no descendants are called <u>leaves</u>. Let leaves(T) denote the maximal left-to-right sequence of all leaves in T.

An <u>L-labelled tree</u> is a tree T and a function λ which assigns a label from L to each node of T. If $G = (V,\Sigma,P,S)$ is a context-free grammar, then T is a <u>tree over G</u> if T is a $(V \cup \{\Lambda\})$-labelled tree. The label of rtn(T) in a labelled tree T is denoted by rtl(T). The <u>frontier</u> of a labelled tree T, denoted by fr(T), is defined as follows:[4]

$$\underset{\sim}{fr}(T) = \lambda(\underset{\sim}{leaves}(T)).$$

Any node of a tree T which is not a leaf is said to be internal. If x is an internal node of T, then[5] $\{y \in T \mid x=y$ or $x \ulcorner y\}$ is the elementary subtree of T rooted at x.

The set of cross-sections of a tree T is defined inductively as follows:

(1) $(x_\emptyset)$, where $x_\emptyset = \underset{\sim}{rtn}(T)$, is a cross-section (CS) of level $\emptyset$.

(2) If $(x_1 \cdots x_k \cdots x_m)$ is a CS of level i and x_k is an internal node of T, then

$$(x_1 \cdots x_{k-1}\ y_1 \cdots y_r\ x_{k+1} \cdots x_m)$$

is a CS of level i + 1, where $y_1,\ldots,y_r$ are the immediate descendants of x_k in order (i.e., with respect to $\sqcap$).

The left canonical cross-sections (LCCS) of T are defined similarly, but with the restriction that x_k (the node that is replaced by its descendants) is the leftmost internal node in the original CS.

Trees T and T' are structurally isomorphic, written $T \cong T'$, if there exists a bijection h from the nodes of T to the nodes of T' such that, for all $x,y \in T$, (i) $x \ulcorner y$ if and only if $h(x) \ulcorner h(y)$, and (ii) $x \sqcap y$ if and only if $h(x) \sqcap h(y)$. If in addition, $\lambda(x) = \lambda(h(x))$ for all $x \in T$, then we write $T = T'$.

Let $G = (V,\Sigma,P,S)$ be a context-free grammar, and let T be a tree over G with labelling λ. T is a grammatical tree over G if $\underset{\sim}{fr}(T) \in \Sigma^*$ and either (i) T consists of a single node, or (ii) to every elementary subtree T' of T there corresponds a production

$A \to \alpha$ in P such that rtl(T') = A and fr(T') = α; furthermore, if any leaf of T' is labelled by Λ, then it is the only leaf in T' (hence $\alpha = \Lambda$). Leaves of a grammatical tree which are labelled with symbols in Σ are called terminal nodes. A grammatical tree T is called a derivation tree if rtl(T) = S.

Let T be a grammatical tree, and let m = lg(fr(T)). Let $(y_1 \cdots y_m)$ be a left-to-right sequence of all terminal nodes in T. For any n, $1 \leq n \leq m$, define the trees

$$^{[n]}T = \{x \in T \mid x \llcorner^* \ulcorner^* y_n\},$$

$$^{\{n\}}T = \{x \in T \mid x \llcorner^* \ulcorner^* y_n\} \cup$$

$$\{x \in T \mid \exists b \in T \text{ s.t. } b \ulcorner^* y_n \text{ and } b \sqcap^+ x\}.$$

Also, let $^{[\emptyset]}T$ and $^{\{\emptyset\}}T$ be the empty tree, and let $^{[n]}T = {}^{\{n\}}T = T$ if $n > m$. We call $^{[n]}T$ and $^{\{n\}}T$ left n-parts. $^{[n]}T$ contains all of the nodes of T which lie on the path from rtn(T) to y_n, plus all nodes of T to the "left" of that path. $^{\{n\}}T$ contains all nodes in $^{[n]}T$, and in addition contains all immediate descendants of nodes on tne path from rtn(T) to y_n. Figure 2.1 shows a grammatical tree T and its left n-parts, for n = 4.

3. Elementary properties of grammars and trees

The following lemmas will be used to prove the main theorems of this paper. The first two lemmas deal with the prefix-free properties of strings derivable in a strict deterministic grammar. The other lemmas (from [3]) concern cross-sections in grammatical trees; they are reproduced here for the convenience of the reader.

Lemma 3.1. Let $G = (V,\Sigma,P,S)$ be a strict deterministic grammar and let π be a strict partition of V. For any $A,A' \in V - \Sigma$, $w,u \in \Sigma^*$, if $A \equiv A' \pmod{\pi}$, $A \Rightarrow^* w$, and $A' \Rightarrow^* wu$, then $u = \Lambda$.

Proof. Identical to the proof of Theorem 2.2 in [7]. □

Definition 3.2. Let $G = (V,\Sigma,P,S)$ be a context-free grammar. For each $\alpha \in V^*$, define $L(\alpha) = \{w \in \Sigma^* \mid \alpha \Rightarrow^* w\}$.

Lemma 3.3. Let $G = (V,\Sigma,P,S)$ be a strict deterministic grammar. For each $\alpha \in V^*$, $L(\alpha)$ is a prefix-free set.

Proof. Use induction on $lg(\alpha)$ and apply Lemma 3.1 to each nonterminal in α. □

Lemma 3.4. [3, Theorem 3.35] Let T be a tree and let x be a node in T. Then x appears in at least one LCCS of T. Moreover, we may assume that there are no internal nodes to the left of x in this cross-section.

Lemma 3.5. [3, Theorem 3.38] Let $G = (V,\Sigma,P,S)$ be a context-free grammar and let T be a grammatical tree over G. If η and η' are LCCS's of level k and k + i, for any $k,i \geq 0$, then $\lambda(\eta) \Rightarrow^{i} \lambda(\eta')$ is a leftmost derivation in G.

<u>Lemma 3.6.</u> [3, Theorem 3.49] Let η be an LCCS of the grammatical tree T at level k and let n be a positive integer. If the restriction η' of η to ${}^{\{n\}}T$ contains an internal node of T then $\eta' = \eta$ and η' is an LCCS of level k in ${}^{\{n\}}T$ as well.

<u>Lemma 3.7.</u> [3, Theorem 3.57] Let T be a grammatical tree and let n be a positive integer. If η is an LCCS of ${}^{\{n\}}T$ then η is an LCCS of T as well.

4. Previous iteration theorems

Before presenting our new iteration theorems, we briefly review the development of iteration theorems for families of deterministic context-free languages.

The first iteration theorem ("pumping lemma," "intercalation theorem," "uvwxy theorem") was introduced by Bar-Hillel, Perles, and Shamir [1] and was applicable to the entire family of context-free languages. It has proved to be a very useful tool for showing that a language is not context-free. A still stronger result was proved later by Ogden [10,11].

<u>Theorem 4.1.</u> ("Ogden's Lemma") Let $G = (V,\Sigma,P,S)$ be a context-free grammar and let $L = L(G)$. There exists an integer p such that, for each $w \in L$ and each set K of p or more distinguished positions in w, there is a factorization $\phi = (w_1,w_2,w_3,w_4,w_5)$ of w such that

1. if $K/\phi = \{K_1,\dots,K_5\}$ then
   (i) either $K_1,K_2,K_3 \neq \emptyset$ or $K_3,K_4,K_5 \neq \emptyset$,
   (ii) $|K_2 \cup K_3 \cup K_4| \leq p$,
2. for each $n \geq 0$, $w_1w_2^nw_3w_4^nw_5 \in L$,
3. for some $A \in V - \Sigma$, $S \Rightarrow^* w_1Aw_5$, $A \Rightarrow^* w_2Aw_4$, and $A \Rightarrow^* w_3$.

A proof of Theorem 4.1 appears in [5].

By studying the special properties of grammatical trees over strict deterministic grammars, Harrison and Havel [8] were able to establish iteration theorems for both strict deterministic languages and (general) deterministic context-free languages.

<u>Theorem 4.2.</u> [8] Let L be a strict deterministic language. There exists an integer p such that, for each $w \in L$ and each set K of p or more distinguished positions in w, there is a factorization $\phi = (w_1, w_2, w_3, w_4, w_5)$ of w such that

1. $w_2 \neq \Lambda$,
2. if $K/\phi = \{K_1, \ldots, K_5\}$ then
   (i) either $K_1, K_2, K_3 \neq \emptyset$ or $K_3, K_4, K_5 \neq \emptyset$,
   (ii) $|K_2 \cup K_3 \cup K_4| \leq p$,
3. for each $n, m \geq 0$, $u \in \Sigma^*$, $w_1 w_2^{n+m} w_3 w_4^n u \in L$ if and only if $w_1 w_2^m w_3 u \in L$.

If, in Theorem 4.2, we replace 3 by

3'. for each $n \geq 0$, $w_1 w_2^n w_3 w_4^n w_5 \in L$, and if $w_5 \neq \Lambda$, then for each $n, m \geq 0$, $u \in \Sigma^*$, $w_1 w_2^{n+m} w_3 w_4^n u \in L$ if and only if $w_1 w_2^m w_3 u \in L$,

then the theorem holds for all deterministic languages.

In [6], the family of real-time strict deterministic languages is introduced. It follows from results in [6] that, for L a real-time strict deterministic language, Theorem 4.2 can be strengthened by adding

4. if $w_4 \neq \Lambda$, then for each $n \geq 0$, $u \in \Sigma^*$, $w_1 w_2^n u \in L$ implies $\lg(u) \geq n$.

Beatty has proved two iteration theorems for LL(k) languages [2,3]. One of these theorems is presented below for comparison with our results in Section 6.

<u>Theorem 4.3.</u> [2,3] Let L be an LL(k) language. There exists an integer p such that, for each $w \in L$ and each set K of p

or more distinguished positions in w, there is a factorization $\phi = (w_1, w_2, w_3, w_4, w_5)$ of w such that

1. $w_2 \neq \Lambda$,
2. if $K/\phi = \{K_1, \ldots, K_5\}$ then
   (i) either $K_1, K_2, K_3 \neq \emptyset$ or $K_3, K_4, K_5 \neq \emptyset$,
   (ii) $|K_2 \cup K_3 \cup K_4| \leq p$,
3. for each $u \in \Sigma^*$, if $w_1 w_2 u \in L$ and ${}^{(k)}u = {}^{(k)}w_3 w_4 w_5$, then there exists a factorization $\xi = (w_1, w_2, w_3', w_4', w_5')$ of $w_1 w_2 u$ such that
   (i) for each $n \geq 0$, for each $u_1, \ldots, u_n \in \{w_4, w_4'\}$, the following are all in L:[6]
   $w_1 w_2^n w_3 (\prod_{i=1}^{n} u_i) w_5$, $w_1 w_2^n w_3' (\prod_{i=1}^{n} u_i) w_5$,
   $w_1 w_2^n w_3 (\prod_{i=1}^{n} u_i) w_5'$, and $w_1 w_2^n w_3' (\prod_{i=1}^{n} u_i) w_5'$,
   (ii) for each $u_1, \ldots, u_n, \bar{u}_1, \ldots, \bar{u}_n \in \{w_4, w_4'\}$, if $\prod_{i=1}^{n} u_i = \prod_{i=1}^{n} \bar{u}_i$, then $u_i = \bar{u}_i$ for $1 \leq i \leq n$.

5. An iteration theorem for strict deterministic languages of degree n

The family of strict deterministic languages, first studied in [7], has been shown to coincide with the family of prefix-free deterministic languages. Thus, any deterministic language can be made a strict deterministic language by adding an endmarker. This fact indicates the usefulness of the class of strict deterministic languages, for by proving properties about it, we can often infer properties of the entire class of deterministic context-free languages.

One of the properties of strict deterministic languages that has been studied is the <u>degree</u> of such a language. One definition of degree has been given in section 2. It is also possible to view the degree of a strict deterministic language L as the number of states in a "minimal" deterministic pushdown automaton (dpda) accepting L by final state and empty store (see [7]).

Until now, there has been no good way to determine the degree of a strict deterministic language. Of course, it is possible to put an upper bound n on the degree of such a language by giving a strict deterministic grammar of degree n that generates the language, or a dpda with n states that recognizes it. Yet, there have been only *ad hoc* methods for showing that a language had degree at least n. In this section, we prove an iteration theorem that enables a lower bound to be placed on the degree of a strict deterministic language, and we give an example of how the theorem is used.

First, however, we quote a "left part theorem" from [8] that we will need to prove our iteration theorem.

Theorem 5.1. [8] Let $G = (V,\Sigma,P,S)$ be a reduced context-free grammar and let π be a partition on V such that $\Sigma \in \pi$. Then π is strict for G if and only if, for any $n \geq 0$ and any grammatical trees T,T' over G, if $\underset{\sim}{rtl}(T) \equiv \underset{\sim}{rtl}(T') \pmod{\pi}$ and ${}^{(n)}\underset{\sim}{fr}(T) = {}^{(n)}\underset{\sim}{fr}(T')$, then there exists a map h such that

(a) ${}^{[n+1]}T \cong {}^{[n+1]}T'$ under h,

(b) $\lambda(x) = \lambda(h(x))$ for all $x \in {}^{[n+1]}T$ such that $x \sqsubset^{+} y$ for some $y \in {}^{[n+1]}T$ (or if ${}^{[n+1]}T = {}^{[n]}T$, for all $x \in {}^{[n+1]}T$), and

(c) $\lambda(x) \equiv \lambda(h(x)) \pmod{\pi}$ for all $x \in {}^{[n+1]}T$.

We can now give the main result of this section.

Theorem 5.2. Let L be a strict deterministic language of degree n. There exists an integer p such that, for each $w \in L$ and each set K of p or more distinguished positions in w, there is a factorization $\phi = (w_1,w_2,w_3,w_4,w_5)$ of w such that

1. $w_2 \neq \Lambda$,
2. if $K/\phi = \{K_1,\ldots,K_5\}$ then
   (i) either $K_1,K_2,K_3 \neq \emptyset$ or $K_3,K_4,K_5 \neq \emptyset$,
   (ii) $|K_2 \cup K_3 \cup K_4| \leq p$,
3. for each $k,m \geq 0$, $u \in \Sigma^*$, $w_1 w_2^{k+m} w_3 w_4^{k} u \in L$ if and only if $w_1 w_2^{m} w_3 u \in L$,
4. for each $u_1,\ldots,u_{n+1} \in \Sigma^*$, if $w_1 w_2^{n_i} u_i \in L$ for $i = 1,\ldots,n+1$, where each $n_i \geq n$, then there exist $1 \leq i < j \leq n+1$, $1 \leq r \leq n_i$, $1 \leq r' \leq n_j$, and factorizations $\xi = (v,x,y,z)$ and $\xi' = (v',x',y',z')$ of u_i and

u_j, respectively, such that

(i) for all $m \geq 0$, the following are all in L:

$w_1 w_2^{(n_i - r)+mr} vxy^m z$, $\quad w_1 w_2^{(n_j - r')+mr'} v'x'y'^m z'$,

$w_1 w_2^{(n_i - r)+mr} v'xy^m z$, and $\quad w_1 w_2^{(n_j - r')+mr'} vx'y'^m z'$,

(ii) none of w_3, v, v' is a proper prefix of any of w_3, v, v'.

Proof. Let $G = (V, \Sigma, P, S)$ be a reduced strict deterministic grammar of degree n such that $L = L(G)$ and let π be a strict partition of V such that $\|\pi\| = n$. The proof of Theorem 4.2 (in [8]) shows that there exists an integer p such that, for each $w \in L$ and each set K of p or more distinguished positions in w, there is a factorization $\phi = (w_1, w_2, w_3, w_4, w_5)$ of w such that parts 1, 2, and 3 hold, and such that, for some $A \in V - \Sigma$,

$$S \Rightarrow^* w_1 A w_5 \Rightarrow^+ w_1 w_2 A w_4 w_5 \Rightarrow^+ w_1 w_2 w_3 w_4 w_5 = w. \qquad (1)$$

Thus, to complete the proof of Theorem 5.2, we need only show that ϕ satisfies part 4 of the theorem.

Assume that $w_1 w_2^{n_i} u_i \in L$ for $i = 1, \ldots, n+1$, where $u_1, \ldots, u_{n+1} \in \Sigma^*$, and each $n_i \geq n$. For each $i = 1, \ldots, n+1$, let T_i' be a derivation tree corresponding to $S \Rightarrow^* w_1 w_2^{n_i} u_i$. Hence, $\underline{rtl}(T_i') = S$ and $\underline{fr}(T_i') = w_1 w_2^{n_i} u_i$.

From (1) we obtain the derivation

$$S \Rightarrow^* w_1 A w_5 \Rightarrow^+ w_1 w_2 A w_4 w_5 \Rightarrow^+ w_1 w_2^2 A w_4^2 w_5 \Rightarrow^+ \cdots \Rightarrow^+$$
$$w_1 w_2^{n_i} A w_4^{n_i} w_5 \Rightarrow^+ w_1 w_2^{n_i} w_3 w_4^{n_i} w_5 \qquad (2)$$

for each i. Let T_i be a derivation tree corresponding to (2). For $j = 0, \ldots, n_i$, let x_j^i be the node of T_i labelled by A in the cross-section (CS) of T_i labelled by $w_1 w_2^j A w_4^j w_5$. Clearly $x_0^i \ulcorner^+ x_1^i \ulcorner^+ \cdots \ulcorner^+ x_{n_i}^i$. Let $k_i = \lg(w_1 w_2^{n_i})$ and let $y_{k_i+1}^i$ be the leaf of T_i which is labelled by the $(k_i+1)^{st}$ symbol in $w_1 w_2^{n_i} w_3 w_4^{n_i} w_5$ (such a node exists since $K_3 \neq \emptyset$). Then, for i = 1,...,n + 1,

$$\underset{\sim}{rtl}(T_i) = \underset{\sim}{rtl}(T_i') = S \text{ and}$$
$$^{(k_i)}\underset{\sim}{fr}(T_i) = {}^{(k_i)}\underset{\sim}{fr}(T_i') = w_1 w_2^{n_i}.$$

Therefore, by Theorem 5.1, there exist maps $h_1,\ldots,h_{n+1}$ such that, for i = 1,...,n + 1,

(a) ${}^{[k_i+1]}T_i \cong {}^{[k_i+1]}T_i'$ under h_i,

(b) $\lambda(x) = \lambda(h_i(x))$ for all $x \in {}^{[k_i+1]}T_i$ such that $x \sqsubset^+ y$ for some $y \in {}^{[k_i+1]}T_i$, and

(c) $\lambda(x) \equiv \lambda(h_i(x)) \pmod{\pi}$ for all $x \in {}^{[k_i+1]}T_i$.

Since w_3 contains a distinguished position, it is nonempty; hence $y^i_{k_i+1}$ is labelled by the first symbol in w_3, so $x^i_{n_i} \ulcorner^+ y^i_{k_i+1}$. Thus,

$$x^i_\emptyset \ulcorner^+ x^i_1 \ulcorner^+ \cdots \ulcorner^+ x^i_{n_i} \ulcorner^+ y^i_{k_i+1},$$

so $x^i_\emptyset,\ldots,x^i_{n_i} \in {}^{[k_i+1]}T_i$. Let $z^i_j = h_i(x^i_j)$ for i = 1,...,n + 1, $j = \emptyset,\ldots,n_i$. By (a), $z^i_\emptyset \ulcorner^+ z^i_1 \ulcorner^+ \cdots \ulcorner^+ z^i_{n_i}$.

By (c), $\lambda(x^i_{n_i}) \equiv \lambda(z^i_{n_i}) \pmod{\pi}$ for i = 1,...,n + 1. Since $\|\pi\| = n$, and $\lambda(x^i_{n_i}) = A$ for all i, there exist i,j, where $1 \leq i < j \leq n + 1$, such that $\lambda(z^i_{n_i}) = \lambda(z^j_{n_j})$. For the remainder of this proof, i and j are fixed at these values. Let $B = \lambda(z^i_{n_i}) = \lambda(z^j_{n_j})$.

Also, for $q = \emptyset,\ldots,n_i$, $\lambda(z^i_q) \equiv \lambda(x^i_q) = A$, so each $\lambda(z^i_q)$ is in the same equivalence class as A. Since $\|\pi\| = n$, there are at most n elements in this equivalence class, so since $n_i \geq n$, there exist $\emptyset \leq s < t \leq n_i$ such that $\lambda(z^i_s) = \lambda(z^i_t)$. By a symmetrical argument, there exist $\emptyset \leq s' < t' \leq n_j$ such that $\lambda(z^j_{s'}) = \lambda(z^j_{t'})$.

We fix the values of s,t,s', and t' for the remainder of the proof. Let $C = \lambda(z^i_s) = \lambda(z^i_t)$ and $D = \lambda(z^j_{s'}) = \lambda(z^j_{t'})$. The trees T_i, T_j, T'_i, and T'_j now appear as in Figure 5.1.

Let η_1, η_2, η_3 be the CS's of T_i in which only x^i_s, x^i_t, $x^i_{n_i}$, respectively, are internal nodes. Then

$$\lambda(\eta_1) = w_1 w_2^s A w_4^s w_5,$$
$$\lambda(\eta_2) = w_1 w_2^t A w_4^t w_5, \text{ and}$$
$$\lambda(\eta_3) = w_1 w_2^{n_i} A w_4^{n_i} w_5$$

by the definition of x^i_s, x^i_t, and $x^i_{n_i}$. Similarly, let η'_1, η'_2, η'_3 be the CS's of T'_i in which only z^i_s, z^i_t, $z^i_{n_i}$, respectively, are internal nodes.

We have already seen that $x^i_s, x^i_t, x^i_{n_i} \in {}^{[k_i+1]}T_i$. Hence, by (b), each node to the left of z^i_s (resp. z^i_t, $z^i_{n_i}$) in η'_1 (resp. η'_2, η'_3) is labelled the same as the corresponding node in η_1 (resp. η_2, η_3). Therefore, for some $x,y,z \in \Sigma^*$, we have

$$\lambda(\eta'_1) = w_1 w_2^s Cz,$$
$$\lambda(\eta'_2) = w_1 w_2^t Cyz, \text{ and}$$
$$\lambda(\eta'_3) = w_1 w_2^{n_i} Bxyz.$$

Let v be the frontier of the tree rooted at $z^i_{n_i}$ and let $r = t - s$ (hence $1 \le r \le n_i$). From η'_1, η'_2, η'_3 we obtain the following derivation:

$$S \Rightarrow^* w_1 w_2^s Cz \Rightarrow^+ w_1 w_2^s w_2^r Cyz \Rightarrow^+$$
$$w_1 w_2^s w_2^r w_2^{n_i-(s+r)} Bxyz \Rightarrow^+ w_1 w_2^s w_2^r w_2^{n_i-(s+r)} vxyz. \qquad (3)$$

Thus, $\xi = (v,x,y,z)$ is a factorization of u_i. Also, from (3) we see that

$$w_1 w_2^{(n_i-r)+mr} vxy^m z \in L$$

for all $m \ge 0$, which satisfies part of 4(i).

The arguments of the last two paragraphs apply if we use T'_j instead of T'_i. Hence, there exist $1 \leq r' \leq n_j$, $v',x',y',z' \in \Sigma^*$ such that $\xi' = (v',x',y',z')$ is a factorization of u_j and

$$\begin{aligned} S &\Rightarrow^* w_1 w_2^{s'} Dz' \Rightarrow^+ w_1 w_2^{s'} w_2^{r'} Dy'z' \Rightarrow^+ \\ &w_1 w_2^{s'} w_2^{r'} w_2^{n_j-(s'+r')} Bx'y'z' \Rightarrow^+ \\ &w_1 w_2^{s'} w_2^{r'} w_2^{n_j-(s'+r')} v'x'y'z'. \end{aligned} \tag{4}$$

Again, from (4) we have that

$$w_1 w_2^{(n_j-r')+mr'} v'x'y'^m z' \in L$$

for all $m \geq 0$.

By substituting the last part of (4), i.e., $B \Rightarrow^+ v'$, into (3), we see that

$$w_1 w_2^{(n_i-r)+mr} v'xy^m z \in L$$

for all $m \geq 0$. Similarly, by substituting $B \Rightarrow^+ v$ into (4), it is clear that

$$w_1 w_2^{(n_j-r')+mr'} vx'y'^m z' \in L$$

for all $m \geq 0$. Thus, 4(i) holds.

Since $A \Rightarrow^* w_3$, $B \Rightarrow^* v$, $B \Rightarrow^* v'$, and $A \equiv B \pmod{\pi}$, none of w_3,v,v' is a proper prefix of any of w_3,v,v', by Lemma 3.1. This establishes 4(ii), completing the proof of Theorem 5.2. □

<u>Definition 5.3.</u> For $n \geq 1$, let L_n denote the context-free language $\{a^m b^k a^m b^k \mid 1 \leq m, 1 \leq k \leq n\}$.

In [7], a hierarchy of strict deterministic languages by degree is established by proving that, for $n > 1$, L_n is not strict deterministic of degree $n - 1$ (or less). The proof there is quite complicated. Using Theorem 5.2, we give a short proof of the same result.

<u>Theorem 5.4.</u> For all $n > 1$, L_n is not strict deterministic of degree $n - 1$.

<u>Proof.</u> Assume for the sake of contradiction that L_n is strict deterministic of degree $n - 1$. Let p be the constant of Theorem 5.2. Let $w = a^p b^n a^p b^n$ and let the leftmost block of p a's be distinguished. By invoking Theorem 5.2, we obtain a factorization $\phi = (w_1, w_2, w_3, w_4, w_5)$ of w such that parts 1 through 4 hold. In order to satisfy 1, 2, and 3, we must have $w_1 = a^s$, $w_2 = a^t$, $w_3 \in a^{p-(s+t)} b^n a^*$, $w_4 = a^t$, and $w_5 \in a^* b^n$, for some $s,t \geq 1$.

Now let

$$u_i = a^{p-(s+t)} b^i a^{p+(n-2)t} b^i$$

for $1 \leq i \leq n$. Clearly $w_1 w_2^{n-1} u_i \in L$ for $1 \leq i \leq n$, so by part 4 of the theorem, there exist $1 \leq i < j \leq n$, $1 \leq r, r' \leq n - 1$, and factorizations $\xi = (v,x,y,z)$ and $\xi' = (v',x',y',z')$ of u_i and u_j, respectively, such that 4(i) and 4(ii) hold. Since v is a prefix of u_i and v' is a prefix of u_j, and, by 4(ii), neither v nor v' is a proper prefix of w_3, it must be the case that $v \in a^{p-(s+t)} b^i a^+$ and $v' \in a^{p-(s+t)} b^j a^*$ (see Figure 5.2). (Observe that, by 4(i), $w_1 w_2^{(n-1-r)+mr} vxy^m z \in L$ for all $m \geq 0$. Since $w_2 \neq \Lambda$ and $r \geq 1$, this implies that $y \in a^+$. Similarly, we must have $y' \in a^+$. Thus, neither v nor v' can include the entire block of $p + (n-2)t$ a's in u_i or u_j, respectively.) By 4(i), with $m = 1$, $w_1 w_2^{n-1} v'xyz \in L$. However, since $w_1 w_2^{n-1} v'wyz \in a^* b^j a^* b^i$ and $i \neq j$, this is impossible. Therefore, L_n is not strict deterministic of degree $n - 1$. □

6. An iteration theorem for simple deterministic languages

In [9], Korenjak and Hopcroft defined the family of simple deterministic languages. This family was originally studied because it was the first nontrivial class of languages for which the equivalence problem was known to be decidable.

It has been shown that the family of simple deterministic languages coincides with the family of strict deterministic languages of degree 1 (except for $\{\Lambda\}$, which is not simple deterministic). Hence, Theorem 5.2 (with $n = 1$) can be used to show that a language is not simple deterministic. However, using the special properties of the simple deterministic languages, we prove in this section a stronger and more concise iteration theorem for this family.

The following theorem is due to Beatty.

Theorem 6.1. [3] Let $G = (V,\Sigma,P,S)$ be a reduced context-free grammar. Then G is LL(k) if and only if, for any $n \geq 0$ and any grammatical trees T,T' over G, if $\underset{\sim}{rtl}(T) = \underset{\sim}{rtl}(T')$ and ${}^{(n+k)}\underset{\sim}{fr}(T) = {}^{(n+k)}\underset{\sim}{fr}(T')$, then ${}^{\{n+1\}}T = {}^{\{n+1\}}T'$.

From Theorem 6.1 we can derive a theorem characterizing the grammatical trees of a simple deterministic grammar. This theorem will then allow us to prove the main result of this section.

Theorem 6.2. Let $G = (V,\Sigma,P,S)$ be a reduced context-free grammar in Greibach normal form. Then G is simple deterministic if and only if

(*) for any $n \geq 0$ and any grammatical trees T,T' over G, if $\underline{rtl}(T) = \underline{rtl}(T')$ and ${}^{(n)}\underline{fr}(T) = {}^{(n)}\underline{fr}(T')$ then ${}^{\{n\}}T = {}^{\{n\}}T'$.

Proof. Suppose that G is simple deterministic. Every simple deterministic grammar is LL(1) [12], so by Theorem 6.1, for any $n \geq 0$ and any grammatical trees T,T', if $\underline{rtl}(T) = \underline{rtl}(T')$ and ${}^{(n+1)}\underline{fr}(T) = {}^{(n+1)}\underline{fr}(T')$, then ${}^{\{n+1\}}T = {}^{\{n+1\}}T'$. Since ${}^{\{0\}}T = {}^{\{0\}}T'$ for any T,T', we can replace n + 1 by n to get (*).

Conversely, suppose that (*) holds. Then, for any $n \geq 0$ and any grammatical trees T,T', if $\underline{rtl}(T) = \underline{rtl}(T')$ and ${}^{(n+1)}\underline{fr}(T) = {}^{(n+1)}\underline{fr}(T')$, then ${}^{\{n+1\}}T = {}^{\{n+1\}}T'$. Hence, by Theorem 6.1, G is LL(1). By [12], G is simple deterministic, since G is LL(1) and in Greibach normal form. □

We now prove an iteration theorem for simple deterministic languages.

Theorem 6.3. Let L be a simple deterministic language. There exists an integer p such that, for each $w \in L$ and each set K of p or more distinguished positions in w, there is a factorization $\phi = (w_1, w_2, w_3, w_4, w_5)$ of w such that

1. $w_2 \neq \Lambda$,
2. if $K/\phi = \{K_1, \ldots, K_5\}$ then
   (i) either $K_1, K_2, K_3 \neq \emptyset$ or $K_3, K_4, K_5 \neq \emptyset$,
   (ii) $|K_2 \cup K_3 \cup K_4| \leq p$,
3. for each $u \in \Sigma^*$, if $w_1 w_2 u \in L$, then there exists a factorization $\xi = (w_1, w_2, w_3', w_4', w_5')$ of $w_1 w_2 u$ such that
   (i) for each $n \geq 0$, for each $u_1, \ldots, u_n \in \{w_4, w_4'\}$, the following are all in L:

$$w_1 w_2^n w_3 (\prod_{i=1}^{n} u_i) w_5, \qquad w_1 w_2^n w_3' (\prod_{i=1}^{n} u_i) w_5,$$

$$w_1 w_2^n w_3 (\prod_{i=1}^{n} u_i) w_5', \text{ and } \qquad w_1 w_2^n w_3' (\prod_{i=1}^{n} u_i) w_5',$$

(ii) w_3 (resp. w_4, w_5) is not a proper prefix of w_3' (resp. w_4', w_5') and vice-versa.

Proof. (Our proof is similar to the proof of Theorem 4.3 as given in [3], but is somewhat less formal.) Let $G = (V, \Sigma, P, S)$ be a reduced simple deterministic grammar such that $L = L(G)$. Thus, as we noted in Section 2, G is strict deterministic. The proof of Theorem 4.2 in [8] shows that there exists an integer p such that, for each $w \in L$ and each set K of p or more distinguished positions in w, there is a factorization $\phi = (w_1, w_2, w_3, w_4, w_5)$ of w such that 1 and 2 of the theorem hold, and such that, for some $A \in V - \Sigma$,

$$S \Rightarrow^* w_1 A w_5 \Rightarrow^+ w_1 w_2 A w_4 w_5 \Rightarrow^+ w_1 w_2 w_3 w_4 w_5 = w. \qquad (1)$$

We must now show that ϕ satisfies part 3 of the theorem. Let T be a derivation tree corresponding to (1). Let x (resp. y) be the node of T labelled by A in the CS of T labelled by $w_1 A w_5$ (resp. $w_1 w_2 A w_4 w_5$). Clearly $x \Gamma^* y$.

Suppose that $w_1 w_2 u \in L$ for some $u \in \Sigma^*$. Let T' be a derivation tree corresponding to $S \Rightarrow^* w_1 w_2 u$. Thus, $\underset{\sim}{rtl}(T') = S$ and $\underset{\sim}{fr}(T') = w_1 w_2 u$. Let $k = lg(w_1 w_2)$. Since $\underset{\sim}{rtl}(T) = \underset{\sim}{rtl}(T') = S$ and ${}^{(k)}\underset{\sim}{fr}(T) = {}^{(k)}\underset{\sim}{fr}(T') = w_1 w_2$, we have by Theorem 6.2 that ${}^{\{k\}}T = {}^{\{k\}}T'$. Let h be the isomorphism that maps nodes of ${}^{\{k\}}T$ to nodes of ${}^{\{k\}}T'$.

Let y_k (resp. y_{k+1}) denote the leaf of T labelled by the k^{th} (resp. $(k+1)^{st}$) symbol in $w = w_1 w_2 w_3 w_4 w_5$. Since $w_2 \neq \Lambda$ by part

1 of the theorem, and $w_3 \neq \Lambda$ by part 2, y_k is labelled by the last symbol of w_2, and y_{k+1} is labelled by the first symbol of w_3. By the definition of x and y, we have that $x \Gamma^* y_k$ and $y \Gamma^* y_{k+1}$.

Since y_k and y_{k+1} are leaves of T, we have that $y_k L^+ y_{k+1}$. Suppose that there exists a leaf $y' \in T$ such that $y_k L^+ y' L^+ y_{k+1}$. Since G is in Greibach normal form, y' is labelled by some $a \in \Sigma$. But then y_{k+1} cannot be labelled by the $(k+1)^{st}$ symbol in w, which is a contradiction. Therefore, $y_k L y_{k+1}$.

By the definition of L, there exist $z_1, z_2 \in T$ such that

$$y_k (\Gamma_R^{-1})^* z_1 \sqcap z_2 (\Gamma_L)^* y_{k+1}.$$

Let z be the parent of z_1 and z_2 (see Figure 6.1). Since $y \Gamma^* y_{k+1}$, either $y \Gamma^* z$ or $z_2 \Gamma_L^* y$. Suppose that $y \Gamma^* z$. Then $y \Gamma^+ z_1 \Gamma^* y_k$, which is impossible since both y_k and y appear in the CS of T labelled by $w_1 w_2 A w_4 w_5$. Thus, it must be the case that $z_2 \Gamma_L^* y$.

Suppose that $z_2 \neq y$. Let z' be the leftmost immediate descendant of z_2. Since G is in Greibach normal form, z' is labelled by some $a \in \Sigma$. Since $z_2 \Gamma_L^* y$ and $z_2 \neq y$, we have that $z' \Gamma_L^* y$. However, $z' \neq y$ since $\lambda(y) = A$, and z' has no descendants, so it is not possible that $z' \Gamma_L^* y$, which is a contradiction. Hence, $z_2 = y$. Since $z_1 \Gamma^* y_k$ and $z_1 \sqcap y$, $y \in {}^{\{k\}}T$.

Since $x \Gamma^* y_k$, $x \in {}^{\{k\}}T$. Thus, both x and y are in ${}^{\{k\}}T$, so we have that

$A = \lambda(x) = \lambda(h(x))$ and

$A = \lambda(y) = \lambda(h(y))$.

Let η and θ be LCCS's of T in which the leftmost internal nodes are x and y, respectively. Such LCCS's must exist by Lemma 3.4. From the definition of x and y, we see that

$\lambda(\eta) = w_1 A\beta$ and

$\lambda(\theta) = w_1 w_2 A\alpha\beta$

for some $\alpha, \beta \in V^*$.

Since x and y are internal nodes of T that belong to ${}^{\{k\}}T$, η and θ are LCCS's of ${}^{\{k\}}T$ (Lemma 3.6). But ${}^{\{k\}}T = {}^{\{k\}}T'$, so $h(\eta)$ and $h(\theta)$ are LCCS's of ${}^{\{k\}}T'$, hence by Lemma 3.7, $h(\eta)$ and $h(\theta)$ are LCCS's of T'. Since ${}^{\{k\}}T = {}^{\{k\}}T'$,

$\lambda(h(\eta)) = \lambda(\eta) = w_1 A\beta$ and

$\lambda(h(\theta)) = \lambda(\theta) = w_1 w_2 A\alpha\beta$.

Applying Lemma 3.5 to η and θ, we have

$$S \Rightarrow^* w_1 A\beta \Rightarrow^* w_1 w_2 A\alpha\beta \Rightarrow^* w_1 w_2 w_3 w_4 w_5 \qquad (2)$$

(where $A \Rightarrow^* w_3$, $\alpha \Rightarrow^* w_4$, and $\beta \Rightarrow^* w_5$, by the definition of α and β). Next, we apply Lemma 3.5 to $h(\eta)$ and $h(\theta)$ to get

$$S \Rightarrow^* w_1 A\beta \Rightarrow^* w_1 w_2 A\alpha\beta \Rightarrow^* w_1 w_2 u.$$

Let $w_3', w_4', w_5' \in \Sigma^*$ be such that $u = w_3' w_4' w_5'$, $A \Rightarrow^* w_3'$, $\alpha \Rightarrow^* w_4'$, and $\beta \Rightarrow^* w_5'$. (See Figure 6.2.)

Setting $\xi = (w_1, w_2, w_3', w_4', w_5')$, we have that ξ is a factorization of $w_1 w_2 u$. Let n be any nonnegative integer. From (2), we obtain the derivation

$$S \Rightarrow^* w_1 A\beta \Rightarrow^* w_1 w_2 A\alpha\beta \Rightarrow^* w_1 w_2^2 A\alpha^2\beta \Rightarrow^* \cdots \Rightarrow^* w_1 w_2^n A\alpha^n\beta.$$

A terminal string may now be derived by continuing with either

$$A \Rightarrow^* w_3 \quad \text{or} \quad A \Rightarrow^* w_3',$$

then n applications of any combination of

$$\alpha \Rightarrow^* w_4 \quad \text{and} \quad \alpha \Rightarrow^* w_4',$$

completing the derivation with either

$$\beta \Rightarrow^* w_5 \quad \text{or} \quad \beta \Rightarrow^* w_5'.$$

Clearly any of the strings in part 3(i) of the theorem may be obtained in this manner, so ξ satisfies 3(i).

Since G is strict deterministic, by Lemma 3.3 each of L(A), $L(\alpha)$ and $L(\beta)$ is a prefix-free set. Thus, since $w_3, w_3' \in L(A)$, w_3 is not a proper prefix of w_3' and vice-versa. Similarly, w_4 (resp. w_5) is not a proper prefix of w_4' (resp. w_5'), and vice-versa. Therefore, ξ satisfies part 3(ii), and the theorem is proved. □

Theorem 6.3 resembles Theorem 4.3 (Beatty's first iteration theorem for LL(k) languages) in the case that k = 1. This is understandable, since every simple deterministic language is LL(1). There are two differences between the theorems, however. First, condition 3 in Theorem 6.3 requires only a string $w_1w_2u \in L$, while Theorem 4.3 requires a string $w_1w_2u \in L$ such that $^{(1)}u = {}^{(1)}w_3$. Second, part 3(ii) in Theorem 6.3 is stronger than the corresponding condition in Beatty's theorem. In fact, part 3(ii) is very useful in practice, as we see in the following example.

<u>Theorem 6.4.</u> The language $L_1 = \{a^n(bd + b + c)^n\$ \mid n \geq 1\}$ (where (,), and + are metasymbols, + denoting alternation) is not simple deterministic.

<u>Proof.</u> Assume that L_1 is simple deterministic and let p be the constant that Theorem 6.3 asserts must exist. Let $w = a^p(bd)^p\$$ and let the symbols $(bd)^p$ be distinguished. By Theorem 6.3, there is a factorization $\phi = (w_1, w_2, w_3, w_4, w_5)$ of w which satisfies parts 1, 2, and 3 of the theorem. In particular, since

part 3 is satisfied, $w_1 w_2^n w_3 w_4^n w_5$ must be in L_1 for all $n \geq 0$. This implies that $w_1 \in a^*$, $w_2 \in a^+$, and either

I. $w_3 \in a^*(bd)^+$, $w_4 \in (bd)^+$, and $w_5 \in (bd)^+\$$, or

II. $w_3 \in a^*(bd)^*b$, $w_4 \in d(bd)^*b$, and $w_5 \in d(bd)^*\$$.

The first case occurs if w_4 begins witn a b and ends with a d, the second if w_4 begins with a d and ends with a b. Note that w_4 cannot both begin and end with a b, for then $w_1 w_3 w_5$ would contain a d not immediately preceded by a b. Neither can w_4 both begin and end with a d, since $w_1 w_2^2 w_3 w_4^2 w_5$ would then contain adjacent d's. We consider cases I and II separately. (See Figure 6.3.)

<u>Case I.</u> We can write $w_2 = a^k$, $w_4 = (bd)^k$, $w_5 = (bd)^m\$$ for some $k,m \geq 1$. Let $u = w_3(bd)^{k-1}bc^m\$$. Clearly $w_1 w_2 u \in L$. By Theorem 6.3, there exists a factorization $\xi = (w_1, w_2, w_3', w_4', w_5')$ of $w_1 w_2 u$ such that parts 3(i) and 3(ii) are satisfied. From 3(ii) we see that $w_3' = w_3$, $w_4' \in (bd)^{k-1}bc^+$, and $w_5' \in c^*\$$. But now $w_1 w_2^2 w_3' w_4'^2 w_5' \notin L$ (since $w_3' w_4'^2 w_5' \in a^*(bd)^{p-m-1}bc^+(bd)^{k-1}bc^m\$$, we have that $w_1 w_2^2 w_3' w_4'^2 w_5'$ contains p + k a's but more than p + k repetitions of bd, b, and c). Hence, 3(i) is contradicted.

<u>Case II.</u> We can write $w_4 = d(bd)^k b$, $w_5 = d(bd)^m\$$ for some $k,m \geq 0$. Let $u = w_3 c^{k+m+1}\$$. It is easy to verify that $w_1 w_2 u \in L$. By Theorem 6.3, there exists a factorization $\xi = (w_1, w_2, w_3', w_4', w_5')$ of $w_1 w_2 u$ such that parts 3(i) and 3(ii) are satisfied. From 3(ii) we have that $w_3' = w_3$, $w_4' \in c^+$, and $w_5' \in c^*\$$. However, $w_1 w_2 w_3 w_4' w_5 \notin L$ (since w_4' ends in a c and w_5 begins with a d), which contradicts 3(i).

In both cases we reach a contradiction, so L_1 cannot be simple deterministic. □

The language L_1 above is a variation on the LL(k) language $\{a^n(b^kd + b + cc)^n \mid n \geq 1\}$ (where k is any fixed value greater than or equal to 1) which Rosenkrantz and Stearns [12] showed could not be generated by an LL(k) grammar without Λ-rules. Since the class of simple deterministic languages is equal to the class of languages generated by LL(1) grammars without Λ-rules [12], their result shows that $\{a^n(bd + b + cc)^n \mid n \geq 1\}$ is not a simple deterministic language. Unfortunately, this also follows trivially from the observation that $\{a^n(bd + b + cc)^n \mid n \geq 1\}$ is not prefix-free. Thus, the added \$ is essential in Theorem 6.4. Note also that each of the alternates (bd, b, and c) in L_1 is necessary for L_1 to be nonsimple. An interesting exercise is to verify that the languages $\{a^n(bd + b)^n\$ \mid n \geq 1\}$, $\{a^n(bd + c)^n\$ \mid n \geq 1\}$, and $\{a^n(b + c)^n\$ \mid n \geq 1\}$ are all simple deterministic.

We have noted earlier that every simple deterministic language is both LL(1) and strict deterministic (hence prefix-free). The language L_1 is LL(1), since it is generated by the following LL(1) grammar:

$S \rightarrow aDA\$$,

$D \rightarrow aDA \mid \Lambda$,

$A \rightarrow bB \mid c$,

$B \rightarrow d \mid \Lambda$.

Hence, L_1 is a prefix-free LL(1) language which is not simple deterministic. Our final theorem follows immediately.

<u>Theorem 6.5.</u> The class of simple deterministic languages is properly included in the class of prefix-free LL(1) languages.

References

[1] Y. Bar-Hillel, M. Perles, and E. Shamir, On formal properties of simple phrase structure grammars, Zeitschrift für Phonetik, Sprachwissenschaft und Kommunikationsforschung 14 (1961) 143-172. Also available in Y. Bar-Hillel, Language and Information (Addison-Wesley, Reading, Mass., 1964) 116-150.

[2] J. C. Beatty, Iteration theorems for LL(k) languages, Proc. Ninth ACM Symp. on Theory of Computing (1977) 122-131.

[3] J. C. Beatty, Iteration theorems for the LL(k) languages, Ph.D. Thesis, Univ. of California, Berkeley, Calif. (1977).

[4] L. Boasson, Two iteration tneorems for some families of languages, J. Comput. Sys. Sci. 7 (1973) 583-596.

[5] M. A. Harrison, Introduction to Formal Language Theory (Addison-Wesley, Reading, Mass., 1978).

[6] M. A. Harrison and I. M. Havel, Real-time strict deterministic languages, SIAM J. Comput. 1 (1972) 333-349.

[7] M. A. Harrison and I. M. Havel, Strict deterministic grammars, J. Comput. Sys. Sci. 7 (1973) 237-277.

[8] M. A. Harrison and I. M. Havel, On the parsing of deterministic languages, J. ACM 21 (1974) 525-548.

[9] A. J. Korenjak and J. E. Hopcroft, Simple deterministic languages, Conf. Rec. of the Seventh Ann. IEEE Symp. on Switching and Automata Theory, Berkeley, Calif. (1966) 36-46.

[10] W. F. Ogden, Intercalation theorems for pushdown store and stack languages, Ph.D. Thesis, Stanford Univ. (1968).

[11] W. F. Ogden, A helpful result for proving inherent ambiguity, Math. Systems Theory 2 (1968) 191-194.

[12] D. J. Rosenkrantz and R. E. Stearns, Properties of deterministic top-down grammars, Information and Control 17 (1970) 226-256.

Footnotes

[1]The empty string is denoted by Λ.

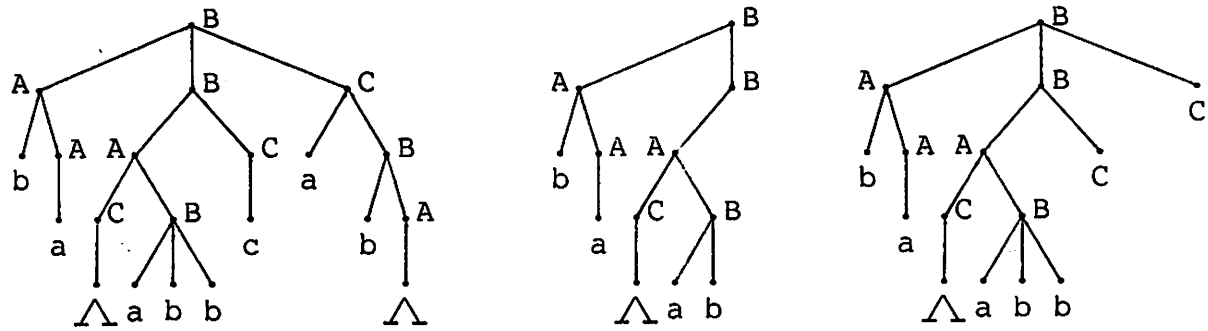
[2]The length of a string x is denoted by lg(x).

[3]We use |S| to denote the cardinality of a set S.

[4]We extend λ to sequences of nodes in the natural way: if η = $(x_1\ x_2\ \cdots\ x_m)$ is a sequence of nodes from T, then $\lambda(\eta) = \lambda(x_1)\lambda(x_2)\cdots\lambda(x_m)$.

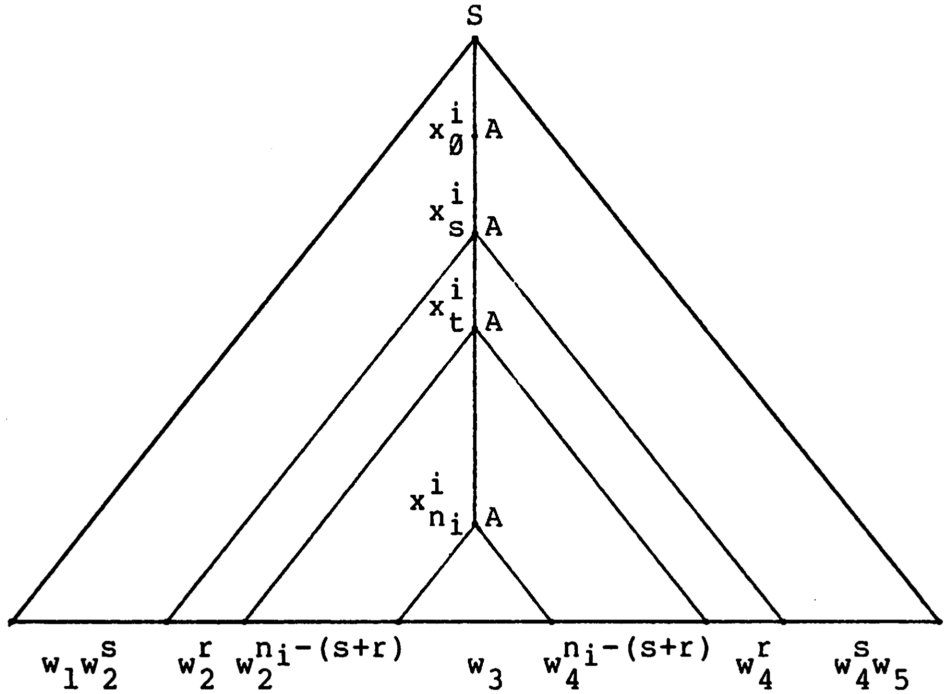
[5]When defining a subtree of T, we list only the nodes in that subtree. All edges of T which connect nodes of the subtree are implicitly included in the subtree. Also, for T = (V,E) we write $x \in T$ instead of $x \in V$.

[6]For $u_1,\ldots,u_n \in \Sigma^*$, we let $\prod_{i=1}^{n} u_i = u_1 u_2 \cdots u_n$.

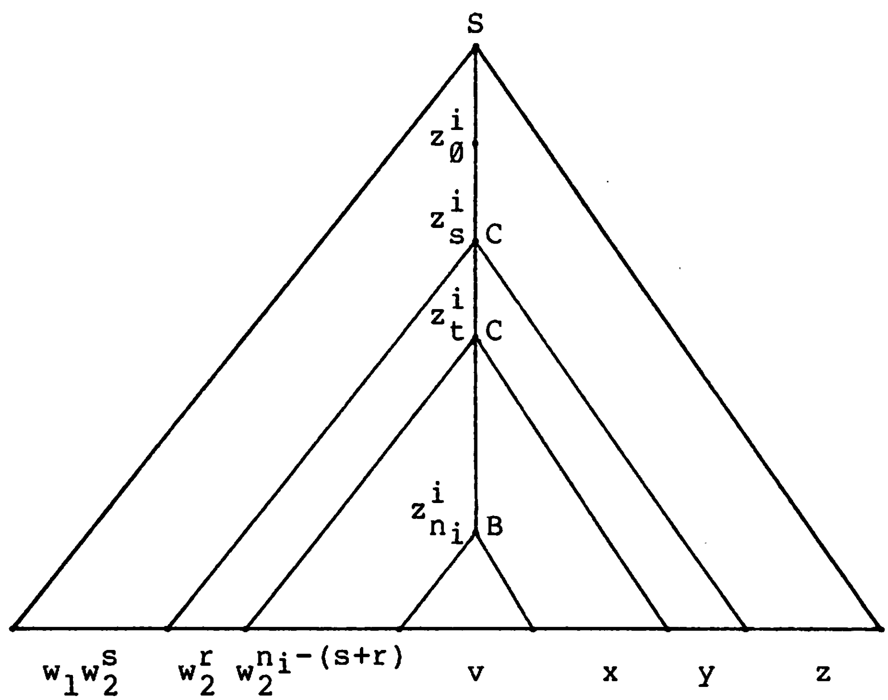


T [4]$_T$ {4}$_T$

Figure 2.1



T_i



T_i'

Figure 5.1
(continued on next page)

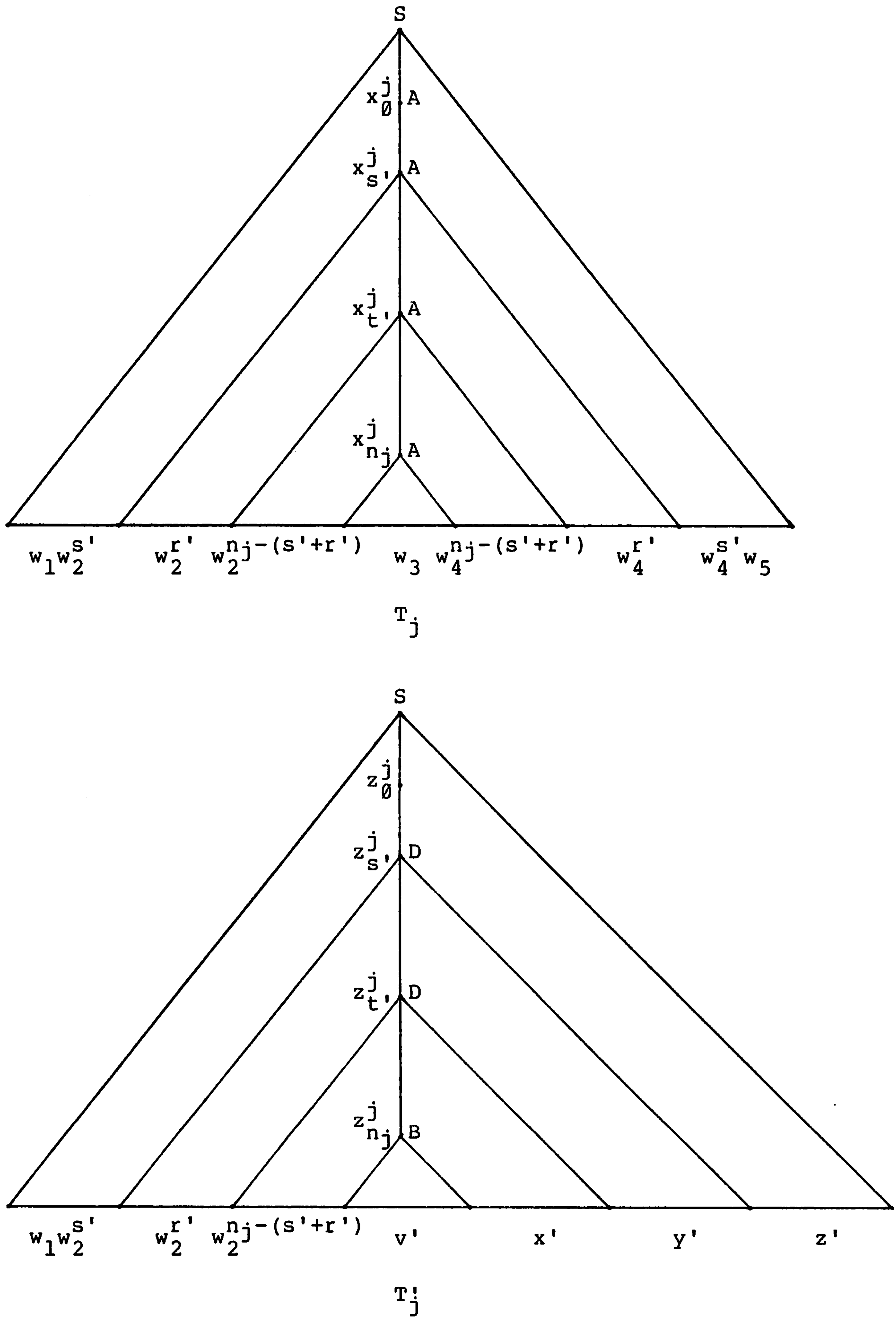

S
$x^j_\emptyset$ A
$x^j_{s'}$ A
$x^j_{t'}$ A
$x^j_{n_j}$ A
$w_1w_2^{s'}$
$w_2^{r'}$
$w_2^{n_j-(s'+r')}$
w_3
$w_4^{n_j-(s'+r')}$
$w_4^{r'}$
$w_4^{s'}w_5$
T_j
S
$z^j_\emptyset$
$z^j_{s'}$ D
$z^j_{t'}$ D
$z^j_{n_j}$ B
$w_1w_2^{s'}$
$w_2^{r'}$
$w_2^{n_j-(s'+r')}$
v'
x'
y'
z'
T'_j

Figure 5.1

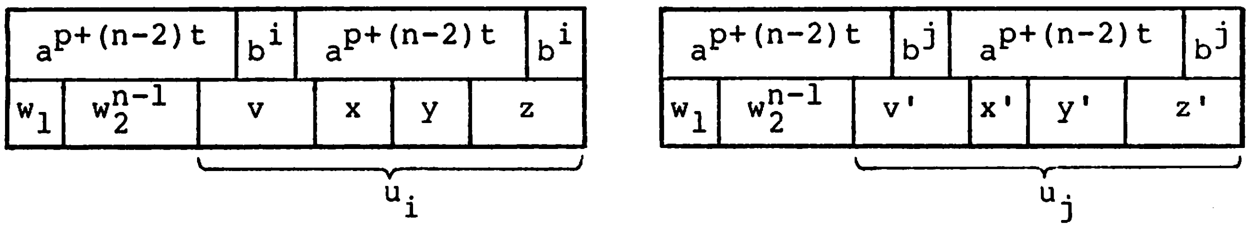


Figure 5.2

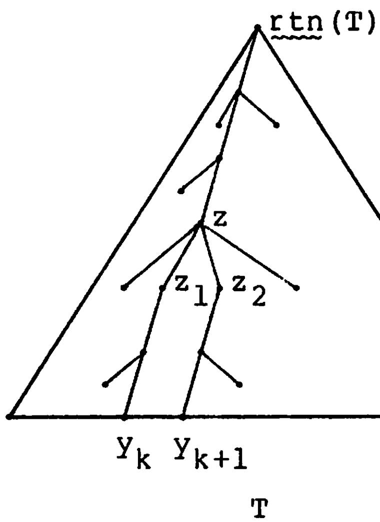


Figure 6.1

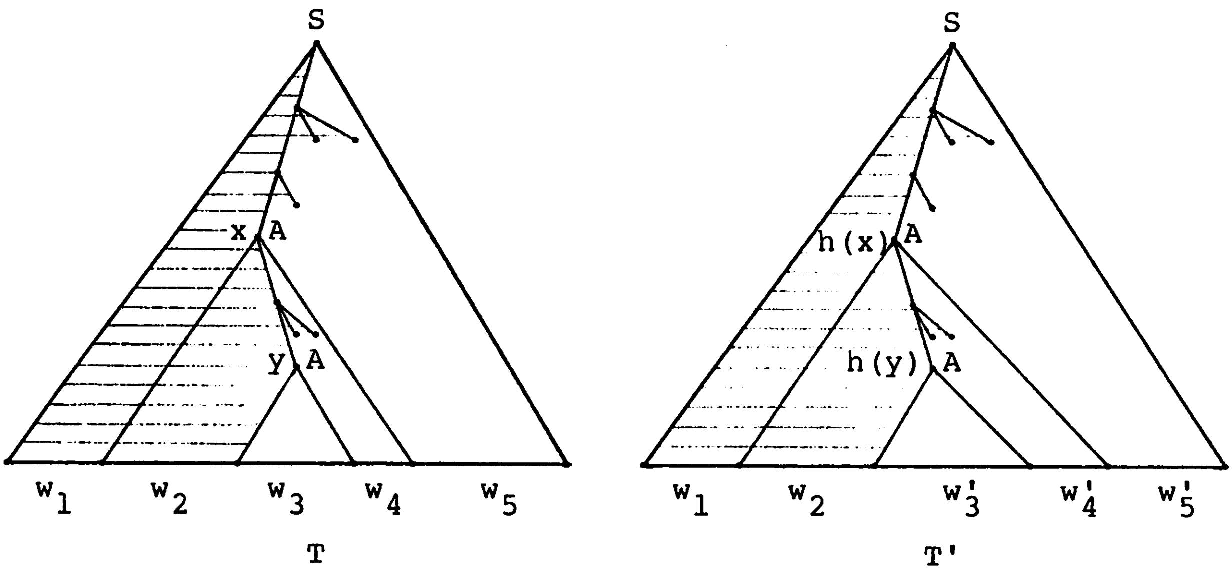


Figure 6.2

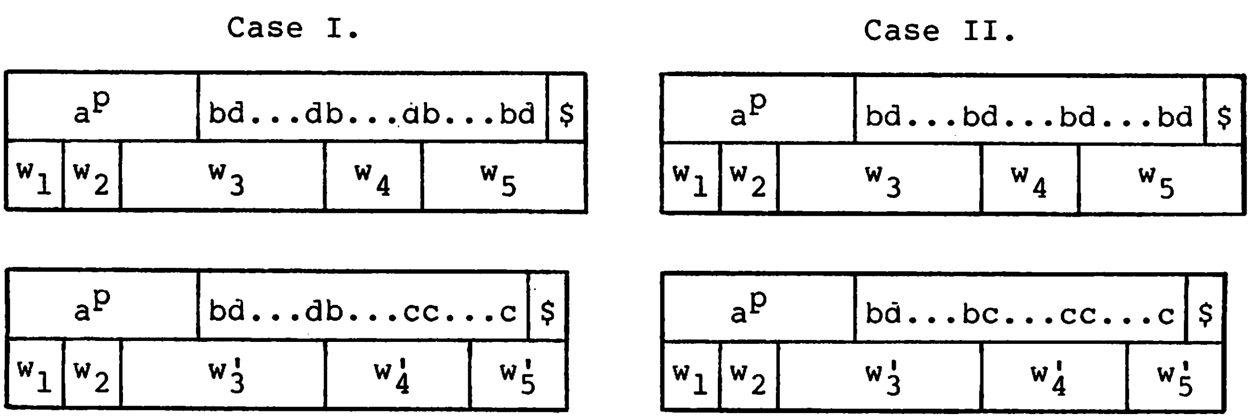


Figure 6.3