Theoretical Computer Science 67 (1989) 75-85
North-Holland

A GRAMMATICAL CHARACTERIZATION OF ALTERNATING PUSHDOWN AUTOMATA

Etsuro MORIYA
Department of Mathematics, Tokyo Woman's Christian University, Zenpukuji, Suginami-ku, Tokyo 167, Japan

Communicated by A. Salomaa
Received January 1988

Abstract. The notion of an alternating context-free grammar is introduced and it is shown that the class of alternating context-free languages is equal to the class of languages accepted by alternating pushdown automata. Some properties on alternating context-free languages are also studied.

0. Introduction

The notion of alternation was introduced by Chandra, Kozen and Stockmeyer [2, 3, 8] to generalize the notion of nondeterminism. While the notion of nondeterminism has played a number of important roles in formal language and automata theories, the notion of alternation has contributed a considerable development to computational complexity theory. Accordingly, the alternating version of various types of automata have been studied [2, 5, 7, 9, 10, 12]. Up to now, however, there is no known characterization of an alternating automaton from a grammatical point of view.

In this paper we introduce the notion of an alternating context-free grammar in order to characterize the alternating pushdown automata grammatically. In Section 1, some basic definitions and properties on alternating context-free grammars are given. Section 2 is devoted to the proof of the equivalence of the alternating context-free grammars and the alternating pushdown automata.

1. Alternating context-free grammars

The reader is assumed to be familiar with the basic concepts in formal language and computational complexity theories. Unless stated otherwise, basic notations in this paper follow [6]. We begin with the following fundamental definition.

Definition. An *alternating context-free grammar* (*ACFG* for short) is a quintuple $G = (N, U, \Sigma, P, S)$, where (N, Σ, P, S) is a context-free grammar (CFG), called the

0304-3975/89/$3.50

underlying CFG of G, and U is a subset of N. Elements of U and $N-U$ are called *universal* and *existential nonterminals*, respectively. A production whose left side is an existential (resp. universal) nonterminal is called an *existential* (resp. *universal*) *production*.

Let α be in $(N\cup\Sigma)^*$. A finite tree T is a *derivation tree* or simply a *derivation* for G from α if the following properties are satisfied:

(a) Each node π of T is labeled with a string in $(N\cup\Sigma)^*$, $\ell(\pi)$; in particular, the root of T is labeled with α.

(b) If π is an internal node of T such that $\ell(\pi)=xAy$ with A in $N-U$, and $A\to z$ is a production in P, then π has exactly one son π' with $\ell(\pi')=xzy$. In this case, π is called an *existential node*.

(c) If π is an internal node of T such that $\ell(\pi)=xAy$ with A in U, and $A\to z_1|z_2|\cdots|z_k$ are the A-productions (i.e., productions whose left side is A) in P, then π has exactly k sons $\pi'_1, \pi'_2, \ldots, \pi'_k$ with $\ell(\pi'_i)=xz_iy$, $1\le i\le k$. In this case, π is called a *universal node*.

Note that for any path (i.e., a sequence of nodes) $\pi_1, \pi_2, \ldots, \pi_n$ in a derivation of an ACFG, $\ell(\pi_1)\Rightarrow\ell(\pi_2)\Rightarrow\cdots\Rightarrow\ell(\pi_n)$ is a derivation in the underlying CFG.

To each node π of a derivation T, the *value* of π, $\mathrm{val}(\pi)$, is assigned as follows. The values are computed bottom up in T:

$$\mathrm{val}(\pi)=\begin{cases}\ell(\pi) & \text{if } \pi \text{ is a leaf of } T,\\ \mathrm{val}(\tau) & \text{if } \pi \text{ is an existential node and } \tau \text{ is its son,}\\ w & \text{if } \pi \text{ is a universal node and}\\ & \quad w=\mathrm{val}(\tau) \text{ for each son } \tau \text{ of } \pi,\\ \text{undefined} & \text{otherwise.}\end{cases}$$

A derivation is said to be *valid* if the value is defined at the root of it. The value of a derivation T is denoted by $\mathrm{value}(T)$. If T is valid, then T is said to *generate* $\mathrm{value}(T)$ *from* ℓ(the root of T). Note that all the leaves of a valid derivation T have the same label, $\mathrm{value}(T)$. The label of a node of a derivation whose root is labeled with the sentence symbol is called a *sentential form*. A *terminating derivation* is a derivation whose leaves each have a label in Σ^*. An *acceptable derivation* is a terminating valid derivation whose root is labeled with the sentence symbol. The *language generated* by G is defined to be the set

$$L(G)=\{\mathrm{value}(T)\mid T \text{ is an acceptable derivation in } G\}.$$

A language L is an *alternating context-free language* (*ACFL*) if $L=L(G)$ for some ACFG G.

As for alternating automata, we classify ACFGs by the number of alternations made in a derivation. A path in a derivation of an ACFG is said to *make an alternation* if a node on the path is existential and the immediate successor of it is universal, or vice versa. Let G be an ACFG and k a positive integer. A derivation

T in G is a *Σ_k-derivation* (resp. *Π_k-derivation*) if the root of T is existential (resp. universal) and each path in T from the root to a leaf makes at most $k-1$ alternations. G is *Σ_k-ACFG* (resp. *Π_k-ACFG*) if, for each w in $L(G)$, there exists an acceptable Σ_k-derivation (resp. Π_k-derivation) generating w. A language L is a *Σ_n-ACFL* (resp. *Π_n-ACFL*) if $L=L(G)$ for some Σ_n-ACFG (resp. *Π_n-ACFG*) G. Let Σ_n^{CF} and Π_n^{CF} denote the classes of Σ_n-ACFLs and Π_n-ACFLs, respectively. By definition, Σ_1^{CF} coincides with the class of context-free languages.

Example 1.1. Let $G_1=(\{S, A, A', C, C'\}, \{S\}, \{a, b, c\}, P_1, S)$ be an ACFG, where P_1 consists of the following productions:

$$S \to A \mid C,$$

$$A \to aA \mid A', \qquad A' \to bA'c \mid \lambda,$$

$$C \to Cc \mid C', \qquad C' \to aC'b \mid \lambda,$$

where λ denotes the empty string. Then any acceptable derivation in G_1 must be of the form shown in Fig. 1, where each of m, m', n and n' is a nonnegative integer. Let π be the root of the tree and let π_1 and π_2 be its left and right sons, respectively (i.e., $\ell(\pi)=S$, $\ell(\pi_1)=A$ and $\ell(\pi_2)=C$). Since $\mathrm{val}(\pi_1)=a^m b^n c^n$ and $\mathrm{val}(\pi_2)=a^{n'} b^{n'} c^{m'}$, π has a defined value if and only if $m=n'=n=m'$. Thus $L(G_1)=\{a^n b^n c^n \mid n \geq 0\}$ and it is in $\Pi_2^{\mathrm{CF}}-\Sigma_1^{\mathrm{CF}}$.

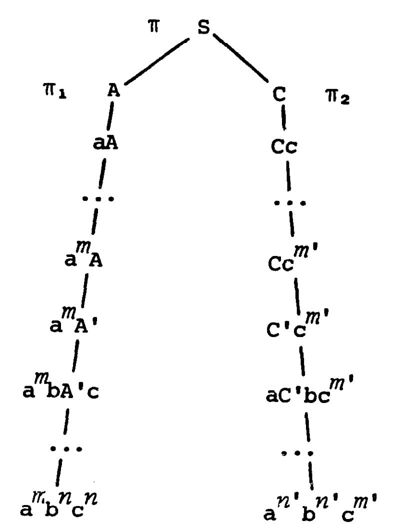

Fig. 1. A derivation in G_1.

Consider another ACFG $G_2=(\{S, A, B, A_i, A_i', B_i, B_i' \mid i=1,2\}, \{A, B\}, P_2, S)$, where P_2 consists of the following productions:

$$S \to A \mid B,$$

$$A \to A_1 \mid A_2, \qquad B \to B_1 \mid B_2,$$

$$A_1 \to A_1 c \mid A_1', \qquad B_1 \to aB_1 \mid B_1',$$

$$A_1' \to aA_1'b \mid \lambda, \qquad B_1' \to bB_1'c \mid \lambda,$$

$$A_2 \to aA_2 \mid A_2', \qquad B_2 \to B_2 c \mid B_2',$$

$$A_2' \to bA_2'c \mid bA_2' \mid \lambda, \qquad B_2' \to aB_2'b \mid B_2'b \mid \lambda.$$

It is easily seen that $L(G_2)$ is equal to $\{a^i b^j c^k \mid i, k \geq 0 \text{ and } j = \max\{i, k\}\}$ and is in $\Sigma_3^{\mathrm{CF}} - \Sigma_1^{\mathrm{CF}} = \Sigma_3^{\mathrm{CF}} - \Sigma_2^{\mathrm{CF}}$ (see Theorem 1.2 below).

Theorem 1.2. *For each integer $n \geq 1$,*

$$\Sigma_n^{\mathrm{CF}} \cup \Pi_n^{\mathrm{CF}} \subset \Sigma_{n+1}^{\mathrm{CF}} \cap \Pi_{n+1}^{\mathrm{CF}}.$$

Proof. Obviously $\Sigma_n^{\mathrm{CF}} \subset \Sigma_{n+1}^{\mathrm{CF}}$ and $\Pi_n^{\mathrm{CF}} \subset \Pi_{n+1}^{\mathrm{CF}}$ hold. Let $G = (N, U, \Sigma, P, S)$ be a Σ_n-ACFG. Let S' be a new symbol not in U and consider the ACFG $G' = (N \cup \{S'\}, U \cup \{S'\}, \Sigma, P \cup \{S' \to S\}, S')$. Clearly, G' is a Π_{n+1}-ACFG with $L(G') = L(G)$. This shows $\Sigma_n^{\mathrm{CF}} \subset \Pi_{n+1}^{\mathrm{CF}}$ and thus $\Sigma_n^{\mathrm{CF}} \cup \Pi_n^{\mathrm{CF}} \subset \Pi_{n+1}^{\mathrm{CF}}$. The inclusion $\Sigma_n^{\mathrm{CF}} \cup \Pi_n^{\mathrm{CF}} \subset \Sigma_{n+1}^{\mathrm{CF}}$ is proved similarly. □

Lemma 1.3. *Each nonempty language in Π_1^{CF} is a singleton.*

Proof. Let G be a Π_1-ACFG with $L(G) \neq \emptyset$, and let T_1 and T_2 be acceptable derivation trees in G. Consider an arbitrary path from the root to a leaf in T_1. The path can be regarded as a sequence of productions $p = (p_1, p_2, \ldots, p_n)$, where p_i $(1 \leq i \leq n)$ is the production used in the transition from the i-th node to its successor on the path. Using the fact that each node on the path is universal, one can show by induction on the length of p that there is a path q in T_2 which is a permutation of p. Since q generates the same string as p, value(T_1) = value(T_2). □

Corollary 1.4. *Π_1^{CF} consists of the empty set and all singleton languages.*

Notation. Let $G = (N, U, \Sigma, P, S)$ be an ACFG. For each A in U, let $G|A$ denote the Π_1-*ACFG* $(U, U, \Sigma, P_{\mathrm{U}}, A)$, where P_{U} is the set of universal productions in P.

Corollary 1.5. *Let $G = (N, U, \Sigma, P, S)$ be an ACFG and let $\sigma = \sigma_0 A_1 \sigma_1 \ldots A_n \sigma_n$ be a string in $(N \cup \Sigma)^*$ with each σ_i $(0 \leq i \leq n)$ in Σ^* and each A_i $(1 \leq i \leq n)$ in U. Let T be a valid terminating derivation in G whose root is labeled with σ and each of whose internal nodes is universal. Then* value$(T) = \sigma_0 L(G|A_1)\sigma_1 \ldots L(G|A_n)\sigma_n$.

Let G be an arbitrary ACFG and T a derivation in G. For convenience, the leaves of T are considered to be both existential and universal. An *existential segment* of T is a maximal subtree T' of T such that every node in T' is existential. A *universal segment* is defined similarly. One segment is said to be *younger* than the other if it is nearer to the leaves.

Theorem 1.6

$$\begin{array}{ccccccccccccccc} \Pi_1^{CF} & \subset & \Sigma_2^{CF} & \subset & \Pi_3^{CF} & \subset & \cdots & \subset & \Sigma_{2n}^{CF} & \subset & \Pi_{2n+1}^{CF} & \subset & \Sigma_{2n+2}^{CF} & \subset & \cdots \\ & & \| & & \| & & & & \| & & \| & & \| & & \\ & & \Sigma_1^{CF} & \subset & \Pi_2^{CF} & \subset & \cdots & \subset & \Sigma_{2n-1}^{CF} & \subset & \Pi_{2n}^{CF} & \subset & \Sigma_{2n+1}^{CF} & \subset & \cdots \end{array}$$

Proof. First, we show that $\Sigma_{2n}^{CF}=\Sigma_{2n-1}^{CF}$ for each $n \geq 1$. Obviously $\Sigma_{2n-1}^{CF} \subset \Sigma_{2n}^{CF}$. To prove the converse inclusion, let $G=(N, U, \Sigma, P, S)$ be a Σ_{2n}-ACFG. By Lemma 1.3, for each A in N, $L(G|A)$ is either the empty set or a singleton. Let

$P'=P \cup \{B \to \beta' \mid B \in N-U,\ \beta \in (U \cup \Sigma)^*,\ B \to \beta$ is in P, and β' is obtained from β by replacing zero or more occurrences of nonterminals in it with the corresponding $L(G|A)$'s provided $L(G|A) \neq \emptyset\}$.

Consider the ACFG $G'=(N, U, \Sigma, P', S)$. By the definition of P', any Σ_m-derivation $(m \leq 2n-1)$ for G is also a Σ_m-derivation for G'. Suppose there is an acceptable Σ_{2n}-derivation for G such that there is a path in it from the root to a leaf making $2n-1$ alternations. Then the youngest segment T' such that its root is on the path is universal (see Fig. 2). Let the root of T' and its predecessor be labeled with β and γ, respectively. Then β is in $(U \cup \Sigma)^*$ and there is an existential production $B \to \alpha$ such that $\gamma=\gamma' B \gamma''$ and $\beta=\gamma' \alpha \gamma''$.

Let $\alpha=\alpha_0 A_1 \alpha_1 \ldots A_n \alpha_n$ with each α_i in Σ^* and each A_i in U. Let

$$\alpha'=\alpha_0 L(G|A_1)\alpha_1 \ldots L(G|A_n)\alpha_n \quad \text{and} \quad \beta'=\gamma' \alpha' \gamma''.$$

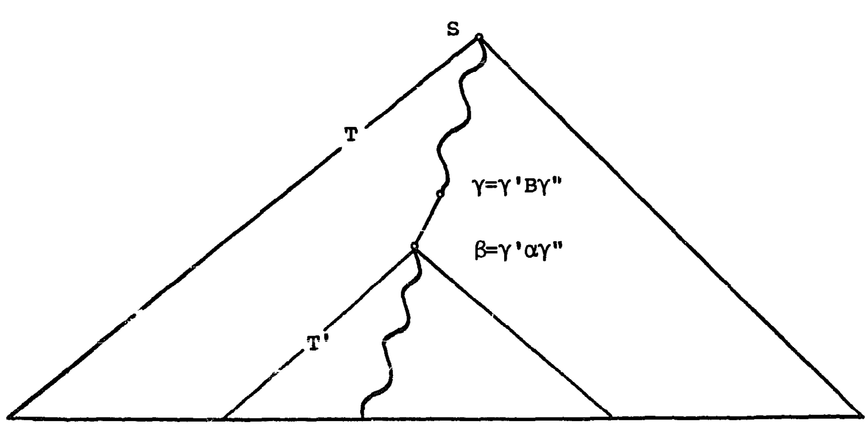

Fig. 2. A derivation in G.

Let T'' be the tree obtained from T' by deleting every subtree corresponding to the derivation for $G|A$, $A \Rightarrow^* L(G|A)$, A being a nonterminal appearing in α. Let the root of T'' be labeled with β' and each of the remaining nodes be labeled with a string which is obtained from the original label of the node in T' by replacing the substring corresponding to α with α'.

Replace the subtree T' of T by T''. From the definition of P', the resulting tree, say T''', is a derivative for G' generating the same terminal string as T.

In case $\gamma'\gamma''$ is in Σ^*, T''' is an acceptable Σ_{2n-1}-derivation for G' and we thus have shown that G' is Σ_{2n-1}-ACFG with $L(G') = L(G)$. (Obviously, $L(G) \subset L(G')$, and by Corollary 1.5, $L(G') \subset L(G)$.) If $\gamma'\gamma''$ still contains universal nonterminals directly derived in T by the application of an existential production, repeat the above procedure until either $\gamma'\gamma''$ belongs to Σ^* or each nonterminal in $\gamma'\gamma''$ is such that it is derived by a universal production. In the former case the youngest universal segment is removed from T, while in the latter case the youngest existential segment is removed. In each case the resulting tree is a valid Σ_{2n-1}-derivation for G', generating the same terminal string as T. This completes the proof of $\Sigma_{2n}^{\mathrm{CF}} \subset \Sigma_{2n-1}^{\mathrm{CF}}$.

The proof of $\Pi_{2n+1}^{\mathrm{CF}} \subset \Pi_{2n}^{\mathrm{CF}}$ is analogous. (Note that the youngest segment of a Π_{2n+1}-derivation is universal).

Finally, the inclusion $\Sigma_{2n-1}^{\mathrm{CF}} \subset \Pi_{2n}^{\mathrm{CF}} \subset \Sigma_{2n+1}^{\mathrm{CF}}$ for $n \geq 1$ follows from Theorem 1.2. □

If $\mathscr{L}_1$ and $\mathscr{L}_2$ are classes of languages, then let

$$\mathscr{L}_1 \vee \mathscr{L}_2 = \{L_1 \cup L_2 \mid L_i \in \mathscr{L}_i \ (i = 1, 2)\},$$

$$\mathscr{L}_1 \wedge \mathscr{L}_2 = \{L_1 \cap L_2 \mid L_i \in \mathscr{L}_i \ (i = 1, 2)\},$$

$$\mathscr{L}_1\mathscr{L}_2 = \{L_1 L_2 \mid L_i \in \mathscr{L}_i \ (i = 1, 2)\}, \quad \text{and}$$

$$\mathscr{L}_1^* = \{L_1^* \mid L_1 \in \mathscr{L}_1\}.$$

Theorem 1.7. *For each n,*

(1) $\Sigma_n^{\mathrm{CF}} \vee \Sigma_n^{\mathrm{CF}} = \Sigma_n^{\mathrm{CF}}$ *and* $(\Sigma_n^{\mathrm{CF}} \cup \Pi_n^{\mathrm{CF}}) \vee \Pi_n^{\mathrm{CF}} \subset \Sigma_{n+1}^{\mathrm{CF}}$,

(2) $\Pi_n^{\mathrm{CF}} \wedge \Pi_n^{\mathrm{CF}} = \Pi_n^{\mathrm{CF}}$ *and* $(\Sigma_n^{\mathrm{CF}} \cup \Pi_n^{\mathrm{CF}}) \wedge \Sigma_n^{\mathrm{CF}} \subset \Pi_{n+1}^{\mathrm{CF}}$,

(3) $\Sigma_n^{\mathrm{CF}}\Sigma_n^{\mathrm{CF}} = \Sigma_n^{\mathrm{CF}}$, $\Pi_n^{\mathrm{CF}}\Pi_n^{\mathrm{CF}} = \Pi_n^{\mathrm{CF}}$ *and* $\Sigma_n^{\mathrm{CF}}\Pi_n^{\mathrm{CF}} \cup \Pi_n^{\mathrm{CF}}\Sigma_n^{\mathrm{CF}} \subset \Sigma_n^{\mathrm{CF}} \cap \Pi_n^{\mathrm{CF}}$, *and*

(4) $(\Sigma_n^{\mathrm{CF}})^* = \Sigma_n^{\mathrm{CF}}$ *and* $(\Pi_n^{\mathrm{CF}})^* \subset \Sigma_{n+1}^{\mathrm{CF}}$.

Proof. To prove (2), for example, let $G_i = (N_i, U_i, \Sigma, P_i, S_i)$, $i = 1, 2$, be ACFGs. We may assume that $N_1 \cap N_2 = \emptyset$. Let $G = (N_1 \cup N_2 \cup \{S\}, U_1 \cup U_2 \cup \{S\}, \Sigma, P\ S)$ be the ACFG defined by $P = \{S \to S_1 \mid S_2\} \cup P_1 \cup P_2$, where S is a new symbol not in $N_1 \cup N_2$. Then $L(G) = L(G_1) \cap L(G_2)$.

The remaining relations can be proved similarly. □

Corollary 1.8. *The ACFLs are closed under union, concatenation, Kleene *, and intersection, but not closed under homomorphism.*

Proof. The nonclosure under homomorphism follows from the fact that every recursively enumerable set can be expressed as a homomorphic image of the intersection of two context-free languages [4]. □

2. Alternating pushdown automata

The alternating version of pushdown automata were first introduced by Chandra, Kozen and Stockmeyer [2] and subsequently studied by Ladner, Lipton and Stockmeyer [9, 10] from a standpoint of computational complexity theory. In fact, they characterized the alternating pushdown automata (APDAs) in terms of time bounded Turing machines, i.e., the class of APDA languages is proved to be precisely the class $\bigcup_{n>0} \mathrm{DTIME}(c^n)$; moreover, this is true if the APDA has either a one-way input head or two-way one [2].

In this section, we shall characterize the APDAs in terms of ACFGs. For this purpose, we briefly review the definition of an APDA. In our model, the machine is one-way and acceptance is defined by "empty stack", so the set of "accepting states" is not designated in specifying the machine.

Definition. An *alternating pushdown automaton* (*APDA* for short) is a system $M = (Q, U, \Sigma, \Gamma, \delta, q_0, Z_0)$, where Q is the set of states, $U \subset Q$ is the set of *universal states* ($Q - U$ is the set of *existential states*), Σ is the input alphabet, Γ is the pushdown alphabet, $q_0 \in Q$ is the initial state, Z_0 is the start symbol on the pushdown stack, and δ is the transition function from $Q \times (\Sigma \cup \{\lambda\})$ to the finite subsets of $Q \times \Gamma^*$.

An *instantaneous description* (*ID*) of M has the form (q, w, γ), where q is a state, w is a string of input symbols, and γ is a string of stack symbols. (q, w, γ) is *existential* if q is so; *universal* otherwise. The *initial ID* of M on input w is (q_0, w, Z_0); *accepting IDs* are those of the form (q, λ, λ), where $q \in Q$. A *computation* or *computation tree* of M on input w is a finite rooted tree whose nodes are labeled with IDs of M, with the following properties:

(a) the root is labeled with the initial ID of M on input w;

(b) if π is an existential internal node and its label, $\ell(\pi)$, is $(q, aw, Z\gamma)$, then π has exactly one son ρ such that $\ell(\rho) = (p, w, \beta\gamma)$, where (p, β) is an element of $\delta(q, a, Z)$; and

(c) if π is a universal internal node, its label is $(q, aw, Z\gamma)$ and $\delta(q, a, Z) = \{(p_i, \beta_i) \mid 1 \leq i \leq k\}$, then π has exactly k sons $\rho_1, \ldots, \rho_k$ such that $\ell(\rho_i) = (p_i, w, \beta_i\gamma)$.

When $U = \emptyset$, M is an ordinary pushdown automaton. If all universal states are forced to be existential, then the resulting machine is called the *underlying PDA* of M, and the "move" relation $\vdash$ on it is defined as usual.

An *accepting computation* of M on input w is a computation tree of M on input w, whose leaves each are labeled with an accepting ID. We say *M accepts w* if there is an accepting computation of M on input w. The set of input strings that M accepts, denoted by $L(M)$, is called an *APDA language*. Σ_k- and Π_k-*APDA*s and their languages are defined similarly as for Σ_k- and Π_k-ACFGs and their languages. By Σ_k^{PDA} and Π_k^{PDA} we denote the classes of Σ_k- and Π_k-APDA languages respectively. At present there is no known hierarchy for Σ_k- and Π_k-APDA languages [10].

A derivation tree for an ACFG is said to be *leftmost* if each path in the tree from the root to a leaf is a leftmost derivation for the underlying CFG.

Lemma 2.1. *Let G be an ACFG. For each w in $L(G)$, there exists a leftmost derivation generating w.*

Proof. Let T be a derivation tree generating w. If T is not leftmost, construct from T a derivation tree T' as follows. Let $p = (p_1, p_2, \ldots, p_k)$ be a path in T from the root to a leaf, where each p_i $(1 \leq i \leq k)$ is a production of G. The path forms a derivation for the underlying CFG $\underset{\sim}{G}$ of G. As is well-known, there exists a permutation p' of p such that it forms a leftmost derivation for $\underset{\sim}{G}$. Let T' be the smallest tree satisfying the property that p' is a path in T' from the root to a leaf if and only if p is a path in T from the root to a leaf.

T' can be constructed from T by traversing T in the depth-first order so that it is a leftmost derivation tree generating w. $\square$

The following two lemmas are an analogue to the well-known proof of the equivalence between the CFGs and the pushdown automata [6].

Lemma 2.2. *For each ACFL L, there exists an APDA M such that $L = L(M)$.*

Proof. Let $L = L(G)$, where $G = (N, U, \Sigma, P, S)$ is an ACFG. We shall construct an APDA M so that M simulates the leftmost derivations for G. Initially, M holds S on the pushdown stack as the start symbol. Subsequently, M will hold on its stack a prefix of a sentential form in a leftmost derivation for G. Let δ be the transition function of M.

(i) M has only two states; the one, say q_{E}, is existential, and the other, say q_{U}, is universal. The states may change from one to the other by a stationary move, i.e.,

$$\delta(q_{\mathrm{E}}, \lambda, X) = \{(q_{\mathrm{U}}, X)\} \quad \text{if } X \in U$$

and

$$\delta(q_{\mathrm{U}}, \lambda, X) = \{(q_{\mathrm{E}}, X)\} \quad \text{if } X \in N - U.$$

The initial state q_0 of M is q_{E} (resp. q_{U}) if S is existential (resp. universal).

(ii) If the top symbol on the stack is a nonterminal, say X, then it is a leftmost nonterminal of a sentential form in a leftmost derivation for G which M is simulating.

If $X \to x$ is a production in P, then M pops up X from the stack and pushes x instead, consuming no input symbol. This is done existentially or universally according to X being existential or universal, i.e.,

$$\delta(q_E, \lambda, X) = \{(q_E, x) \mid X \to x \text{ is in } P\} \quad \text{if } X \in N - U$$

and

$$\delta(q_U, \lambda, X) = \{(q_U, x) \mid X \to x \text{ is in } P\} \quad \text{if } X \in U.$$

(iii) If the top symbol on the stack is a terminal symbol and if M is scanning just the same symbol on its input tape, then it is popped up from the stack, i.e.,

$$\delta(q_E, a, a) = \{(q_E, \lambda)\} \quad \text{and} \quad \delta(q_U, a, a) = \{(q_U, \lambda)\}$$

for each a in Σ.

By Lemma 2.1, there is a derivation tree for G generating a word w in Σ^* if and only if there is a leftmost derivation tree $T_{G,w}$ for G generating w. By the construction of M, there is a computation tree $T_{M,w}$ of M on input w such that there is a path in $T_{G,w}$ from the root to a node π (which is a leftmost derivation for the underlying CFG of G)

$$S \Rightarrow^* u\alpha, \quad u \text{ in } \Sigma^*, \ \alpha \text{ in } (N \cup \Sigma)^*, \text{ and } w = uv = \ell(\pi) \tag{2.1}$$

if and only if there is a path in $T_{M,w}$ from the root (which is a computation for the underlying PDA of M),

$$(q_0, uv, S) \vdash^* (q, v, \alpha) \quad \text{and} \quad w = uv, \tag{2.2}$$

where q is either q_E or q_U according to π being existential or universal. Moreover, $T_{G,w}$ and $T_{M,w}$ are homeomorphic with each other [1] such that (2.1) corresponds to (2.2). $T_{G,w}$ is acceptable if and only if $T_{M,w}$ is accepting. This shows that $L(G) = L(M)$. □

Corollary 2.3. *For each n, $\Sigma_n^{CF} \subset \Sigma_n^{PDA}$ and $\Pi_n^{CF} \subset \Pi_n^{PDA}$.*

Lemma 2.4. *For each APDA M, there exists an ACFG G such that $L(M) = L(G)$.*

Proof. Let $M = (Q, Q_U, \Sigma, \Gamma, \delta, q_0, Z_0)$. For each p, q in Q, each a in Σ, each Z in Γ, and each γ in Γ^*, let $[p, Z, q]$ and $\langle p, a, \gamma, q\rangle$ be new symbols. For a further new symbol S, let

$$\begin{aligned} N = U \cup \{S\} \cup \{[p, Z, q] \mid p, q \in Q,\ Z \in \Gamma\} \\ \cup \{[p, \gamma, q] \mid p, q \in Q \ \text{ and } \delta(p, a, Z) \text{ contains } (q, \gamma) \\ \text{for some } a \in \Sigma \cup \{\lambda\} \text{ and } Z \in \Gamma\}, \end{aligned}$$

where

$$U = \{\langle p, a, Z, q\rangle \mid p, q \in Q,\ a \in \Sigma \cup \{\lambda\},\ Z \in \Gamma\}.$$

Consider the ACFG $G = (N, U, \Sigma, P, S)$, where P is defined as follows:

(i) $S \to [q_0, Z_0, q]$ is in P for each q in Q.

(ii) Suppose $p \in Q$, $a \in \Sigma \cup \{\lambda\}$ and $Z \in \Gamma$.

(1) If $p \in Q - Q_U$ and $\delta(p, a, Z)$ contains (q, γ), then, for each $r \in Q$, P contains

$$[p, Z, r] \to a[q, \gamma, r] \quad \text{provided } \gamma \neq \lambda, \quad \text{and}$$

$$[p, Z, r] \to a \quad \text{provided } \gamma = \lambda.$$

(2) If $p \in Q_U$ and $\delta(p, a, Z) = \{(p_i, \gamma_i) \mid 1 \leq i \leq m\}$, then P contains

$$[p, Z, r] \to \langle p, a, Z, r \rangle \quad \text{for each } r \in Q.$$

In addition, for each $r \in Q$ and each i $(1 \leq i \leq m)$, P contains

$$\langle p, a, Z, r \rangle \to a[p_i, \gamma_i, r] \quad \text{provided } \gamma_i \neq \lambda, \quad \text{and}$$

$$\langle p, a, Z, r \rangle \to a \quad \text{provided } \gamma_i = \lambda.$$

(iii) For each $[p, \gamma, q]$ in N, if $\gamma = Z_1 Z_2 \ldots Z_k$, where each Z_i $(1 \leq i \leq k)$ is in Γ, then

$$[p, \gamma, q] \to [p, Z_1, r_1][r_1, Z_2, r_2] \ldots [r_{k-1}, Z_k, r_k]$$

with $r_k = q$ is in P for each $r_1, r_2, \ldots, r_{k-1}$ in Q.

Roughly speaking, G simulates M as follows. Consider a computation tree of M on input w, in which there is a path from the root,

$$(q_0, uav, Z_0) \vdash_M^* (p, av, Y_1 Y_2 \ldots Y_n) \quad \text{with } w = uav. \tag{2.3}$$

Then there exists a leftmost derivation tree for G generating w, in which there is a path from the root,

$$S \Rightarrow [q_0, Z_0, q] \Rightarrow^* u[p, Y_1, q_1][q_1, Y_2, q_2] \ldots [q_{n-1}, Y_n, q_n]. \tag{2.4}$$

Suppose $\delta(p, a, Y_1) = \{(p_i, \gamma_i) \mid 1 \leq i \leq m\}$. Suppose further that p is existential and that (2.3) expands into

$$(p, av, Y_1 Y_2 \ldots Y_n) \vdash_M (p_i, v, Z_1 \ldots Z_k Y_2 \ldots Y_n),$$

where $\gamma_i = Z_1 \ldots Z_k$ with each Z_i in Γ. Then by (ii)(1) and (iii), G guesses $r_1, \ldots, r_{k-1}$ so that (2.4) expands into

$$u[p, Y_1, q_1][q_1, Y_2, q_2] \ldots [q_{n-1}, Y_n, q_n]$$

$$\Rightarrow ua[p_i, \gamma_i, q_1][q_1, Y_2, q_2] \ldots [q_{n-1}, Y_n, q_n] \tag{2.5}$$

$$\Rightarrow ua[p_i, Z_1, r_1] \ldots [r_{k-1}, Z_k, q_1][q_1, Y_2, q_2] \ldots [q_{n-1}, Y_n, q_n]. \tag{2.6}$$

When p is universal, by (ii)(2) and (iii), the portion (2.5) must have taken place universally for every (p_i, γ_i).

Equation (2.3) is accepting if and only if (2.4) generates w. Thus $L(M) = L(G)$. □

Very unfortunately the above proof fails to prove an analogue of Corollary 2.3, since the number of alternations in G increases in proportion to the number of universal moves made by M, according to the necessity of the guessing portion (2.6).

Combining Lemmas 2.2 and 2.4, we have the following theorem.

Theorem 2.5. *The class of ACFLs is precisely the class of APDA languages.*

References

[1] M. Behzad, G. Chartrand and L. Lesniak-Foster, *Graphs and Digraphs* (Prindle, Weber & Schmidt, Boston, MA, 1979).

[2] A.K. Chandra, D.C. Kozen and L.J. Stockmeyer, Alternation, *J. Assoc. Comput. Mach.* **28** (1981) 114-133.

[3] A.K. Chandra and L.J. Stockmeyer, Alternation, in: *Proc. 17th FOCS* (1976) 88-108.

[4] S. Ginsburg, S. Greibach and M. Harrision, One-way stack automata, *J. Assoc. Comput. Mach.* **14** (1967) 389-418.

[5] E.M. Gurari and O.H. Ibarra, (Semi)alternating stack automata, *Math. Systems Theory* **15** (1982) 211-224.

[6] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages, and Computation* (Addision-Wesley, Reading, MA, 1979).

[7] J. Hromkovič, On the power of alternation in automata theory, *J. Comput. System Sci.* **31** (1985) 28-39.

[8] D. Kozen, On parallelism in Turing machines, in: *Proc. 17th FOCS* (1976) 89-97.

[9] R. Ladner, R.J. Lipton and L.J. Stockmeyer, Alternating pushdown and stack automata, *SIAM J. Comput.* **13** (1984) 135-155.

[10] R. Ladner, L.J. Stockmeyer and R.J. Lipton, Alternation bounded auxiliary pushdown automata, *Inform. and Control* **62** (1984) 93-108.

[11] A. Salomaa, *Formal Languages* (Academic Press, New York, 1973).

[12] G. Slutzki, Alternating tree automata, *Theoret. Comput. Sci.* **41** (1985) 305-318.