Discrete Mathematics 25 (1979) 51–63.

A LEFT PART THEOREM FOR GRAMMATICAL TREES

A. NIJHOLT

Vrije Universiteit, Department of Mathematics, P.O. Box 7161, Amsterdam, The Netherlands.

Received 14 September 1977
Revised 14 August 1978

A simple left part property for a set of grammatical trees is introduced. The class of left part grammars, a subclass of the class of context-free grammars, is defined. It is shown that the set of grammatical trees of a context-free grammar satisfies this left part property if and only if the context-free grammar is a left part grammar. Some properties of leftpart grammars are considered.

0. Introduction

We consider a global property of the derivation (or parse) trees of *context-free grammars.* This property of the derivation trees of context-free grammars can be considered as a restricted version of the left part property for the trees of *strict deterministic grammars* [3, 4, 5, 6]. In this paper it is shown that this left part property is satisfied by the set of grammatical trees of a *left part grammar*, a type of context-free grammar which we introduce here.

The class of left part grammars is a small extension of the class of *simple chain grammars* [10].

If a context-free grammar is *unambiguous* then each terminal string generated by this grammar has a unique derivation tree. Informally, our left part property requires that every prefix of such a terminal string has a unique "partial" tree. This notion of "partial" tree will be specified.

The aim of this paper is to present this left part property and the class of grammars for which the set of grammatical trees satisfies this property. Except for some informal remarks, in this short paper we will not be concerned with a parsing method for left part grammars. However, the reader who is familiar with simple chain grammars will have no difficulty in finding a very simple parsing method for the left part grammars.

To present the left part property and to describe grammatical trees we use the notations and definitions from [5]. For convenience we repeat, as far as necessary, some of these notions here. For more details the reader is referred to [5, 7].

Among others, an intuitive assumption on "translations" of prefixes of sentences which is discussed in [9] motivated us to introduce this left part property.

The organization of this paper is as follows. The remainder of this section is devoted to establish some definitions and notational conventions on trees, context-free grammars and grammatical trees. In Section 2 we present the left part property and we introduce the left part grammars. Moreover, some properties concerning relationships with other classes of grammars and with classes of languages are presented. In Section 3 we show that a context-free grammar is a left part grammar if and only if its set of grammatical trees satisfies the left property.

1. Preliminaries

1.1. *Trees*

To introduce the concepts of the theory of trees which we need here we will frequently refer to the tree T given in Fig. 1. This introduction goes along similar lines as in [5].

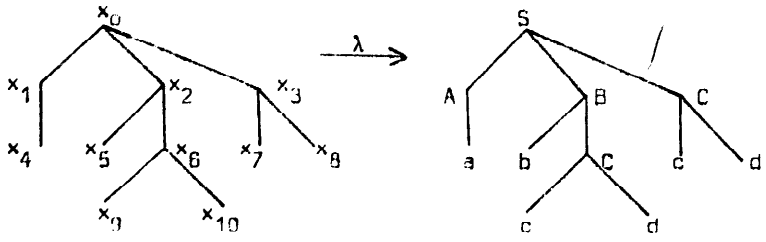

Fig. 1. Tree T and its labeling.

Tree T has *nodes* $(x_0, x_1, \ldots, x_{10})$ and it has a *root* (x_0). The relation of *immediate descendancy* is denoted by $\lceil$ (for example x_5 is an immediate descendant of x_2, $x_2 \lceil x_5$). The *transitive closure* of $\lceil$ is denoted by $\lceil^+$ and the *reflexive and transitive closure* by $\lceil^*$. If $x \lceil^* y$ then there is a *path* from x to y, which is the sequence of all nodes, including x and y, between x and y. For example, x_0, x_2, x_4, x_{10} is the path in T from x_0 to x_{10}. A *leaf* is a node x in T for which there is no y in T such that $x \lceil y$; in Fig. 1 the leaves are $x_4, x_5, x_9, x_{10}, x_7, x_8$, given here in the *left-right order*, which is in general, for a tree T with m leaves, denoted by $y_1, y_2, \ldots, y_m$. We introduce the binary relation $\lfloor$ as follows $x \lfloor y$ iff:

(i) x and y are not on the same path and

(ii) for some leaves y_i, y_{i+1} in the left-right order we have $x \lceil^* y_i$ and $y \lceil^* y_{i+1}$.

Thus, for instance, $x_4 \lfloor x_2$ and, by introducing transitive and reflexive-transitive closures of $\lfloor$ in an obvious way, $x_4 \lfloor^* x_8$.

Two trees T, T' are *structurally isomorphic*, $T \cong T'$, iff there is a bijection $g: T \to T'$ such that $x \lceil y$ iff $g(x) \lceil g(y)$ and $x \lfloor y$ iff $g(x) \lfloor g(y)$, that is, except for a possible labeling the trees are identical.

1.2. *Context-free grammars*

Let $G=(N, \Sigma, P, S)$ be a *context-free grammar* (CFG), where N is the set of *nonterminals*, Σ is the set of *terminals*, $V=N\cup\Sigma$, $P\subseteq N\times V^+$ is the set of *productions* and S is the *start symbol*. Elements of N will be denoted by the roman capitals $A, \ldots, S$; elements of Σ by the Roman smalls $a, b, c, \ldots,$; elements of V^* by the Greek smalls $\alpha, \beta, \gamma, \delta, \ldots$; elements of Σ^* by the Roman smalls u, v, w, x, y, z.

Instead of writing (A, α) in P we write $A\rightarrow\alpha$ in P. P is said to be *prefix-free* if $A\rightarrow\alpha$ and $A\rightarrow\alpha\beta$ in P implies $\beta=\varepsilon$(ε denotes the *empty string*).

The relation $\Rightarrow\subseteq V^*\times V^*$ is defined as follows. For any $\alpha, \beta\in V^*$, $\alpha\Rightarrow\beta$ iff $\alpha=\alpha_1A\alpha_2$, $\beta=\alpha_1\beta_1\alpha_2$ and $A\rightarrow\beta_1$ is in P for some $A\in N$ and $\alpha_1, \alpha_2, \beta_1\in V^*$. If $\alpha_1\in\Sigma^*$ or $\alpha_2\in\Sigma^*$ we write $\alpha\Rightarrow_l\beta$ and $\alpha\Rightarrow_r\beta$ respectively. Transitive and reflexive-transitive closures of these relations are defined in the usual way. If $\alpha_0\Rightarrow\alpha_1\cdots\Rightarrow\alpha_n$ then this sequence is said to be a *derivation* of α_n from α_0.

If $\alpha\in V^+$, then $L(\alpha)=\{w\in\Sigma^+|\alpha\overset{*}{\Rightarrow}w\}$. The *language* of G, denoted by $L(G)$, is the set $L(S)$. If $\alpha\in V^*$, then $|\alpha|$, the *length* of α, denotes the number of symbols in α. If $\alpha\in V^*$, then $^{(n)}\alpha$ denotes α if $|\alpha|<n$ and otherwise a prefix of α of length n.

FIRST $(\alpha)=\{a\in\Sigma|\alpha\overset{*}{\Rightarrow}a\phi$ for some $\phi\in V^*\}$. Notice that $P\subseteq N\times V^+$, hence there are no productions $A\rightarrow\varepsilon$, i.e. the CFG's are assumed to be *ε-free*. Moreover, we assume in this paper that the CFG's are *reduced* and *cycle-free* [1].

1.3. *Grammatical trees*

Let T be a tree. Then every node x of T has a *label* $\lambda(x)$, for instance in Fig. 1 x_3 has label C. We will be concerned with grammatical trees, therefore $\lambda(x)\in V$, where $V=N\cup\Sigma$ for a given CFG $G=(N, \Sigma, P, S)$. The *root-label* of tree T is denoted by rt (T) (in Fig. 1 rt $(T)=S$) and the *frontier* of tree T is the concatenation of the labels of the leaves (in the left-right order) of T, notation: fr (T). In Fig. 1 fr $(T)=abcdcd$. We write $T=T'$ when $T\cong T'$ and T and T' have the same labeling. In this case the corresponding nodes in T and T' will be treated as identical. The productions in P are elementary subtrees (see Fig. 2 for a production $A\rightarrow X_1X_2\cdots X_n$).

Formally, T is said to be a *grammatical tree* iff

(i) for every elementary subtree T' of T there exists a production in P corresponding to T', and

(ii) fr $(T)\in\Sigma^*$.

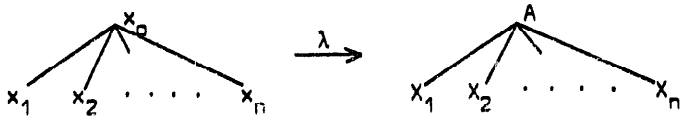

Fig. 2. An elementary subtree.

The set of grammatical trees for a CFG G is denoted by J_G; $J_G(A) = \{T \in J_G | \mathrm{rt}(T) = A\}$ and trees in $J_G(S)$ are the *derivation trees* of G. The correspondence between derivations of G and grammatical trees of G will be clear.

2. Left part grammars

Let $G = (N, \Sigma, P, S)$ be a CFG. Informally the left part property says that for each $A \in N$ and for each prefix u of $w = uv \in L(A)$ u *uniquely* determines the "left part" (up to the first symbol of v) of the grammatical tree which corresponds to the derivation of w from A. Clearly such a property can only be satisfied (take for instance $v = \varepsilon$ and $A = S$) by grammatical trees for which the CFG is *unambiguous*, that is, each sentence (element of $L(S)$) has a unique derivation tree. The following definition of left part is from Harrison and Havel [5].

Definition 2.1. Let T be a grammatical tree of some grammar G. For any $n \geq 0$ we define ${}^{(n)}T$, the *left n-part* of T (or the *left part* when n is understood) as follows. Let $(x_1, \ldots, x_m)$ be the sequence of all leaves in T (from the left to the right). Then ${}^{(n)}T = \{x \in T | x \lfloor^* \lceil^* x_n\}$ if $n \leq m$ and ${}^{(n)}T = T$ if $n > m$. ${}^{(n)}T$ is considered to be a tree under the same relations $\lceil$, $\lfloor$ and the same labeling λ as T.

For instance, in Fig. 1 ${}^{(3)}T$ is the subtree with the nodes x_0, x_1, x_2, x_4, x_5, x_6 and x_9. In the following definition we introduce our simple left part property for a set of grammatical trees.

Definition 2.2. Let $J \subseteq J_G$ for some CFG G. J is said to satisfy the *left part property* iff for any $n > 0$ and $T, T' \in J$ if $\mathrm{rt}(T) = \mathrm{rt}(T')$ and ${}^{(n)}\mathrm{fr}(T) = {}^{(n)}\mathrm{fr}(T')$ then ${}^{(n)}T = {}^{(n)}T'$.

This definition is illustrated in Fig. 3, where two trees T and T' in a set $J \subseteq J_G$ are given with their labeling.
In Fig. 3 we have ${}^{(2)}T = {}^{(2)}T'$. However, since ${}^{(3)}T \neq {}^{(3)}T'$ and ${}^{(3)}\mathrm{fr}(T) = {}^{(3)}\mathrm{fr}(T')$ we may conclude that J does not satisfy the left part property.

Clearly not for every CFG G we have that J_G satisfies the left part property. We introduce the left part grammars, a subclass of the context-free grammars which is defined in such a way that CFG G is a left part grammar iff J_G satisfies the left property. The definition of left part grammars is an adapted version of the definition of *simple chain grammars* which was first introduced in [10]. In Section 1.2 we defined the prefix-free property for a set of productions P. We say a set of productions is *prefix* (1) iff for each pair $A \to \beta$, $A \to \beta\gamma$ in P, where $\gamma \neq \varepsilon$, and for strings $\alpha \in V^*$ and $w \in T^*$, if $S \overset{*}{\Rightarrow}_l wA\alpha$, then $\mathrm{FIRST}(\gamma) \cap \mathrm{FIRST}(\alpha) = \emptyset$. To avoid an empty α we add, if necessary, the production $S' \to S \perp$ to P, where S' is a new start symbol and $\perp$ is an endmarker, $\perp \notin V$.

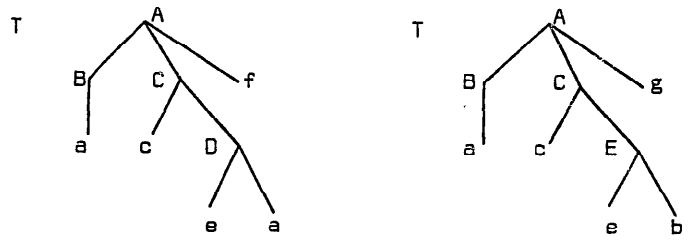

Fig. 3. Two trees, ${}^{(3)}T \neq {}^{(3)}T'$

Definition 2.3. A CFG $G=(N, \Sigma, P, S)$ is said to be a *left part grammar* iff P is prefix (1) and $\mathrm{FIRST}(X) \cap \mathrm{FIRST}(Y) = \emptyset$ for each pair $A \to \alpha X \phi$, $A \to \alpha Y \psi$ in P with $X \neq Y$.

The following definition introduces chains, a concept which turns out to be useful in formulating properties of context-free grammars.

Definition 2.4. Let $G=(N, \Sigma, P, S)$ be a CFG, let $X_0 \in V$. The set of *chains* of X_0, denoted by $\mathrm{CH}(X_0)$ is defined by

$$\mathrm{CH}(X_0) = \{X_0 X_1 \cdots X_n \in N^* \Sigma \mid X_0 \Rightarrow_l X_1 \psi_1 \Rightarrow_l \cdots \Rightarrow_l X_n \psi_n,\ \psi_i \in V^*,\ 1 \leq i \leq n\}$$

Notice that the definition of $\mathrm{CH}(X_0)$ is such that each chain in $\mathrm{CH}(X_0)$ ends with a terminal. If $X_0 \in \Sigma$, then X_0 is the only chain in $\mathrm{CH}(X_0)$. We use the following notations and conventions. If $\pi = X_0 X_1 \cdots X_n$, then $l(\pi) = X_n$, that is $l(\pi)$ denotes the last element of a chain. Hence, for each chain π, $l(\pi) \in \Sigma$.

Let $X \in V$. X is said to be *chain-independent* if for each pair π_1, π_2 in $\mathrm{CH}(X)$, $\pi_1 \neq \pi_2$, we have $l(\pi_1) \neq l(\pi_2)$. Clearly, if X is chain-independent, then $\mathrm{CH}(X)$ is a finite set. Also it follows that each terminal is chain-independent. If each element of V is chain-independent then V is said to be chain-independent. Let $X, Y \in V, X \neq Y$. X and Y are said to be *mutually chain-independent* if for each pair $\pi_1 \in \mathrm{CH}(X)$ and $\pi_2 \in \mathrm{CH}(Y)$ we have $l(\pi_1) \neq l(\pi_2)$; notation $X \not\equiv Y$. Notice that $a \not\equiv b$ for each pair a, b in Σ such that $a \neq b$.

Lemma 2.1. *Let $G=(N, \Sigma, P, S)$ be a* CFG. *For each pair $A \to \alpha X \phi$, $A \to \alpha Y \psi$ in P, where $X \neq Y$, we have* $\mathrm{FIRST}(X) \cap \mathrm{FIRST}(Y) = \emptyset$ *iff* $X \not\equiv Y$.

Proof. Trivial.

Lemma 2.2. *Let $G=(N, \Sigma, P, S)$ be a* CFG. *If* $\mathrm{FIRST}(X) \cap \mathrm{FIRST}(Y) = \emptyset$ *for each pair $A \to \alpha X \phi$, $A \to \alpha Y \psi$ in P with $X \neq Y$ then V is chain-independent.*

Proof. Assume that V is not chain-independent. Hence, there exist $A \in N$ and $\pi_1, \pi_2 \in \mathrm{CH}(A)$ such that $\pi_1 \neq \pi_2$ and $l(\pi_1) = l(\pi_2)$. Let $\pi_1 = X_0 X_1 \cdots X_n$ and $\pi_2 = Y_0 Y_1 \cdots Y_m$, where $X_0 = Y_0 = A$ and $X_n = Y_m$. Then there exists $i \geq 0$ such

that $X_0X_1 \cdots X_i = Y_0Y_1 \cdots Y_i$, there exists a derivation $A \overset{*}{\Rightarrow}_l X_i\psi_i$ for some $\psi_i' \in V^*$ and there exist productions $X_i \to X_{i+1}\psi_{i+1}$, $X_i \to Y_{i+1}\psi_{i+1}'$, for some ψ_{i+1}, $\psi_{i+1}' \in V^*$ and such that $X_{i+1} \neq Y_{i+1}$. Since $\text{FIRST}(X_{i+1}) \cap \text{FIRST}(Y_{i+1}) = \emptyset$ according to the "if"-part of the lemma we have $l(\pi_1) \neq l(\pi_2)$. Contradiction.

From Definition 2.3 and the two lemmas the following corollary is now self-evident.

Corollary 2.1. *A CFG $G = (N, \Sigma, P, S)$ is a left part grammar iff*

(i) *P is prefix(1), and*

(ii) *$X \neq Y$ for each pair $A \to \alpha X\phi$, $A \to \alpha Y\psi$ in P with $X \neq Y$.*

iff

(i) *P is prefix(1), and*

(ii) *V is chain-independent, and*

(iii) *$X \neq Y$ for each pair $a \to \alpha X\phi$, $A \to \alpha Y\psi$ in P with $X \neq Y$ and $\alpha \neq \varepsilon$.*

Remark. If we replace in Definition 2.3 (or in Corollary 2.1) prefix (1) by prefix-free then a definition of the *simple chain grammars* is obtained. The reader who is familiar with their parsing method [10, 2] will notice that this method also will work for left part grammars if one symbol of look-ahead is allowed in cases that there is doubt whether a reduction has to be made or a symbol has to be shifted to the pushdown stack.

Examples. CFG G_1 with only productions $P_1\{S \to aAc, S \to aAd, A \to aA, A \to b\}$ is a left part grammar. The same holds for CFG G_2 with $P_2 = \{S \to Ac, A \to a, A \to ab\}$. A more complicated example is CFG G_3 with $P_3 = \{S \to E\perp, E \to (T)*(E), E \to (T), T \to F+T, T \to F, F \to E, F \to a\}$. CFG G_4 with $P_4 = \{S \to aSA, S \to aA, A \to bbd, A \to b, A \to c, S' \to S\perp\}$ is not a left part grammar since it is not prefix (1).

Notice that the class of simple chain grammars is a *proper* subclass of the left part grammars, since G_2 is not a simple chain grammar. G_2 is not even an LR(0) grammar [1]. With the aid of the property that each simple chain grammar is an LR(0) grammar (which will be proved somewhere else), one can easily verify that the class of left part grammars is a proper subclass of the LR(1) grammars.

In the remainder of this section we prove some results concerning relationships with other classes of grammars and with classes of languages.

The class of prefix-free deterministic languages[1] has been studied by Harrison and Havel [4, 5, 6]. The class of prefix-free deterministic languages is a subclass of the class of deterministic languages. For each prefix-free deterministic language there exist a *strict deterministic grammar* [4, 5, 6]. There is a nontrivial hierarchy of strict deterministic grammars and their languages according to their degree [4].

[1] A deterministic language is a language which can be accepted by a deterministic pushdown automaton [1]. A language L is said to be prefix-free iff $u \in L$ and $uv \in L$ implies $v \in \varepsilon$.

We will turn our attention to the simplest class in this hierarchy, the *strict deterministic grammars of degree* 1. The following definition is a reformulation of Theorem 3.1 of [6].

Definition 2.5. CFG $G=(N, \Sigma, P, S)$ is a strict deterministic grammar of degree 1 iff P is prefix-free and, if $A \rightarrow \alpha X \phi$ and $A \rightarrow \alpha Y \psi$ are in P (hence, α, ϕ and ψ in V^*, X and Y in V), where $X \neq Y$, then X and Y are in Σ.

Simple deterministic grammars [8] are grammars in *Greibach normal form* [2] (GNF, for short) which have the property that for all $a \in \Sigma$, $C \in N$ and $\alpha, \beta \in V^*$,

$$C \rightarrow a\alpha \text{ and } C \rightarrow a\beta \text{ in } P \text{ implies } \alpha = \beta.$$

It follows immediately that the class of simple deterministic grammars is properly included in the class of strict deterministic grammars of degree 1. However, their families of languages coincide [6].

Directly from Definition 2.3, 2.6 and the remark following Corollary 2.1 it follows that the strict deterministic grammars of degree 1 are (properly) included in the class of simple chain grammars, which in its turn is a proper subclass of the class of left part grammars. As a last result of this section we show that the left part grammars with prefix-free production set (hence, the simple chain grammars) generate exactly the class of simple deterministic languages. This is done by presenting a new transformation to GNF which can be used for *non-left recursive* grammars[3], and which, when used for a simple chain grammar, yields a simple deterministic grammar. In the following definition some preliminaries are introduced.

Definition 2.6. Let $G=(N, \Sigma, P, S)$ be a CFG. Define

$$[N]=\{[A\alpha] \mid A \in N, \alpha \in V^* \text{ and } A \rightarrow \alpha\beta \text{ in } P \text{ for some } \beta \in V^*\}$$

and homomorphism $\xi:[N]^* \rightarrow [N]^*$ by letting $\xi([A\alpha])$ is

(i) ε if $A \rightarrow \alpha$ is in P,
(ii) $[A\alpha]$ if $A \rightarrow \alpha\beta$ is in P, where $\beta \neq \varepsilon$.

Now we are sufficiently prepared to present the algorithm.

Algorithm 2.1. Let $G=(N, \Sigma, P, S)$ be an ε-free and non-left-recursive CFG which satisfies the conditions that P is prefix-free and it has no useless symbols. G is transformed to a CFG $G'=(N', \Sigma, P', [S])$ in GNF such that $L(G')=L(G)$.

Method. Set $P'=\emptyset$. N' will contain all symbols of $[N]$ which appear in the productions introduced below.

[2] A CFG $G=(N, \Sigma, P, S)$ is said to be in Greibach normal form iff $P \subseteq N \times \Sigma V^*$, i.e., each production has the form $A \rightarrow a\alpha$ ($A \in N$, $a \in \Sigma$ and $\alpha \in V^*$).

[3] A CFG is said to be non-left-recursive if there is no $A \in N$ such that $A \overset{+}{\Rightarrow} A\alpha$, for some $\alpha \in V^*$.

(i) For each $SX_1 \cdots X_n \in \mathrm{CH}(S)$, let $[S] \to X_n \xi([X_{n-1}X_n][X_{n-2}X_{n-1}] \cdots [SX_1])$ be in P'.

(ii) For each $A \to \alpha X_0 \phi$ in P, where $\alpha \neq \varepsilon$ and $X_0X_1 \cdots X_n \in \mathrm{CH}(X_0)$, let $[A\alpha] \to X_n \xi([X_{n-1}X_n] \cdots [X_0X_1][A\alpha X_0])$ be in P'.

(iii) Remove all useless symbols.

The following two claims are used in the proof of correctness of algorithm.

Claim 2.1. *Let $A \to \alpha X_0 \phi$ be in P, $\alpha \neq \varepsilon$ or $A = S$ and let $X_0X_1 \cdots X_n \in \mathrm{CH}(X_0)$, where $n \geq 0$. Then, for each X_i, $i \geq 0$, if $X_i \overset{m}{\Rightarrow}_l y$[4], where $y \in \Sigma^*$, then $[A\alpha] \overset{*}{\Rightarrow} y\xi([X_{i-1}X_i] \cdots [A\alpha X_0])$.*

Proof. Suppose $m = 0$, then $y = X_i \in \Sigma$. Then, since $A \to \alpha X_0 \phi$ in P and $X_0 \cdots X_{i-1}X_i \in \mathrm{CH}(X_0)$ we have by construction that

$$[A\alpha] \to y\xi([X_{i-1}X_i] \cdots [A\alpha X_0]), \quad \alpha \neq \varepsilon \text{ or } A = S.$$

Now let $m > 0$ and assume the claim holds for all $m' < m$ (induction hypothesis). Then, if $X_i \to Y_1Y_2 \cdots Y_q$ is the first production which is used, we have the following derivation:

$$X_i \Rightarrow Y_1Y_2 \cdots Y_q \overset{*}{\Rightarrow} y_1y_2 \cdots y_q = y$$

where

$$Y_j \overset{m_j}{\Rightarrow} y_j, \quad 1 \leq j \leq q, \quad y_j \in \Sigma^* \quad \text{and} \quad m_j < m.$$

From the induction hypothesis it follows that

$$[A\alpha] \overset{*}{\Rightarrow} y_1\xi([X_iY_1][X_{i-1}X_i] \cdots [X_0X_1][A\alpha X_0]) \qquad (*)$$

and also

$$[X_iY_1 \cdots Y_{j-1}] \overset{*}{\Rightarrow} y_j\xi([X_iY_1 \cdots Y_{j-1}Y_j]), \quad 1 < j \leq q. \qquad (**)$$

From $(*)$ and $(**)$ we obtain

$$[A\alpha] \overset{*}{\Rightarrow} y\xi([X_{i-1}X_i] \cdots [X_0X_1][A\alpha X_0]),$$

which was to be proved.

Claim 2.2. *Let $[A\alpha] \overset{m}{\Rightarrow} w$, then $A \overset{*}{\Rightarrow} \alpha w$.*

[4] We write $\overset{n}{\Rightarrow}$ for $(\Rightarrow)^n$, i.e., $\Rightarrow$ composed with itself $n-1$ times. Similar notation is used for $\Rightarrow_l$ and $\Rightarrow_r$.

Proof. The proof is by induction on m. If $m = 1$ then, since G' is in GNF, we have $w \in \Sigma$, hence $[A\alpha] \to w$ is in P'. Then, by construction, there exists a production $A \to \alpha X_0$ in P such that $X_0 X_1 \cdots X_n \in \mathrm{CH}(X_0)$, $n \geq 0$, $X_n = w$ and $[A\alpha] \to w$ is obtained from

$$[A\alpha] \to w\xi([X_{n-1}w] \cdots [X_0X_1][A\alpha X_0]) = [A\alpha] \to w.$$

Hence, there exists a derivation $A \Rightarrow \alpha X_0 \overset{*}{\Rightarrow} \alpha w$

Now let $m > 1$. The first step of the derivation is done with a production of the form

$$[A\alpha] \to a\xi([X_{n-1}X_n] \cdots [X_0X_1][A\alpha X_0]),$$

where $X_n = a$, $X_0X_1 \cdots X_n$ is in $\mathrm{CH}(X_0)$ and $w = ax$ for some $x \in \Sigma^*$. Then,

$$[A\alpha] \Rightarrow a\xi([X_{n-1}X_n] \cdots [X_0X_1][A\alpha X_0]) \overset{*}{\Rightarrow} ax_nx_{n-1} \cdots x_1x_0 = ax,$$

such that

(i) if $\xi([X_{i-1}X_i]) \neq \varepsilon$, then $[X_{i-1}X_i] \overset{m_i}{\Rightarrow} x_i$, otherwise $x_i = \varepsilon$, $1 \leq i \leq n$, and

(ii) if $\xi([A\alpha X_0]) \neq \varepsilon$, then $[A\alpha X_0] \overset{m_0}{\Rightarrow} x_0$, otherwise $x_0 = \varepsilon$.

Since $m_i < m$, $0 \leq i \leq n$, we obtain

$$X_{i-1} \overset{*}{\Rightarrow} X_ix_i, \qquad 1 \leq i \leq n, \tag{*}$$

and

$$A \overset{*}{\Rightarrow} \alpha X_0x_0. \tag{**}$$

From (*) and (**) it follows that $A \overset{*}{\Rightarrow} \alpha a x_n x_{n-1} \cdots x_1x_0$, hence $A \overset{*}{\Rightarrow} \alpha w$, which was to be proved.

Theorem 2.1. *Algorithm* 2.1, *when applied to a simple chain grammar* G', *yields a simple deterministic grammar* G' *such that* $L(G') = L(G)$.

Proof. That $L(G') \subseteq L(G)$ is an immediate consequence of Claim 2.2, where $\alpha = \varepsilon$ and $A = S$. Next we show $L(G) \subseteq L(G')$. From Claim 2.1 it follows that if $A \to \alpha C\phi$ and $C \overset{*}{\Rightarrow} x$, then

$$[A\alpha] \overset{*}{\Rightarrow} x\xi([A\alpha C]).$$

Notice that the claim holds for $\alpha = \varepsilon$ and $A = S$. If $w \in L(G)$, then there exists a derivation $S \overset{*}{\Rightarrow} w$. If $w \in \Sigma$, then $[S] \to w$ in P', hence, $w \in L(G')$. Otherwise, let $S \to Z_1 Z_2 \cdots Z_n$ be the first production which is used in this derivation. Then $w = z_1 z_2 \cdots z_n \in \Sigma^*$, where $Z_i \overset{*}{\Rightarrow} z_i$, $1 \leq i \leq n$. It follows that

$$[S] \overset{*}{\Rightarrow} z_1[SZ_1], \tag{*}$$

and

$$[SZ_1 Z_2 \cdots Z_i] \overset{*}{\Rightarrow} z_{i+1}\xi([SZ_1 Z_2 \cdots Z_i Z_{i+1}]), \qquad 1 \leq i \leq n. \tag{**}$$

From (*) and (**) it follows that

$$[S] \overset{*}{\Rightarrow} z_1 z_2 \cdots z_n = w,$$

hence, $L(G) \subseteq L(G')$ and we conclude $L(G') = L(G)$. We show that G' is a simple deterministic grammar. Firstly, it is clear that G' is in GNF. Now assume that there exist $a \in \Sigma$, $C \in N'$ and $\alpha, \beta \in (N' \cup \Sigma)^*$ such that $C \to a\alpha$ and $C \to a\beta$ are in P' and $\alpha \neq \beta$. Consider case (i) of the algorithm, hence, $C = [S]$. By construction, there exist $\pi_1, \pi_2 \in \mathrm{CH}(S)$ with $l(\pi_1) = l(\pi_2) = a$, and since $\alpha \neq \beta$, $\pi_1 \neq \pi_2$ which contradicts the properties of a simple chain grammar. Further, consider case (ii) of the algorithm, hence, C is of the form $[A\alpha]$, $\alpha \neq \varepsilon$. Similar observations as in the preceding case lead to a contradiction with the simple chain grammar properties. We may conclude that G' is indeed a simple deterministic grammar.

Note. As remarked above, Algorithm 2.1 can be used to transform any non-left-recursive grammar to a CFG in GNF. Such a non-left-recursive grammar should be, at least for the form in which we present the algorithm here, prefix-free, but for arbitrary non-left-recursive grammars this can be assumed without loss of generality. For example, if there exist productions $A \to \alpha$ and $A \to \alpha\beta$ then one can replace these productions by $A \to \alpha$, $A \to H_\alpha \beta$ and $H_\alpha \to \alpha$.

From Theorem 2.1 it follows that the class of simple deterministic languages is included in the class of left part languages. This inclusion is proper. A trivial example is the language $\{a, ab\}$ which has left part grammar $S \to a|ab$, and, since the language is not prefix-free, it has no simple deterministic grammar. More interesting, however, is the CFG G with productions

$$S \to aSA|aA, \qquad A \to bd|b|c.$$

Obviously, G is a left part grammar. However, the language generated by G is not a simple deterministic language, since $L(G)$ cannot be generated by an ε-free

LL(1)-grammar (see Aho and Ullman [1]). Since each simple deterministic grammar is an ε-free LL(1)-grammar, the proper inclusion follows[5].

3. The left part property

It will be clear from Definition 2.3 that, strictly speaking, we do not really need the concept of a chain to describe left part grammars. However, thinking in terms of chains will sometimes be helpful. Moreover, it is sometimes handsome to use this and the related concepts in proofs on left part grammars. From Definition 2.2 and Definition 2.3 we can now achieve the main result of this paper.

Theorem 3.1 (Left Part Theorem). *Let $G=(N, T, P, S)$ be a* CFG. *The set J_G of all grammatical trees of G satisfies the left part property iff G is a left part grammar.*

Proof. Let G be a left part grammar. To prove: J_G satisfies the left part property. Assume J_G does not satisfy the left part property. Hence there exist $n>0$ and trees T_1 and T_2 in J_G with $\mathrm{rt}(T_1)=\mathrm{rt}(T_2)$, ${}^{(n)}\mathrm{fr}(T_1)={}^{(n)}\mathrm{fr}(T_1)$ and ${}^{(n)}T_1\neq{}^{(n)}T_2$. Suppose $n=1$, then ${}^{(1)}T_1\neq{}^{(1)}T_2$ and ${}^{(1)}\mathrm{fr}(T_1)={}^{(1)}\mathrm{fr}(T_2)$ hence, since $\mathrm{rt}(T_1)=\mathrm{rt}(T_2)$ we must conclude that V is not chain-independent. Contradiction. Suppose $n>1$. For T_1 and T_2 we can choose n such that ${}^{(n-1)}T_1={}^{(n-1)}T_2$ and ${}^{(n)}T_1\neq{}^{(n)}T_2$. Let T_1 be labeled by λ_1 and T_2 by λ_2. The restriction of λ_1 to ${}^{(n-1)}T_1$ which is equal to the restriction of λ_2 to ${}^{(n-1)}T_2$ is denoted by λ. We use the same convention for the relations $\lceil_1$, $\lfloor_1$ on T_1 and $\lceil_2$, $\lfloor_2$ on T_2. Let the leaves of ${}^{(n)}T_1$ have a left-right order $x_1, x_2, \ldots, x_n$. Since ${}^{(n)}\mathrm{fr}(T_1)={}^{(n)}\mathrm{fr}(T_2)$ we have the same order and labels for the leaves of ${}^{(n)}T_2$. Since ${}^{(n-1)}T_1={}^{(n-1)}T_2$, the path in T_1 from the root of T_1 to x_{n-1} is the same (including the labeling) as the path in T_2 from the root of T_2 to x_{n-1}. Let this path be $p=(y_0, y_1, \ldots, y_m)$, where y_0 is the root, $y_m=x_{n-1}$ and $y_0\lceil y_1\lceil\cdots\lceil y_m$. Since ${}^{(n)}T_1\neq{}^{(n)}T_2$ there exist nodes y_i and y_j on p $(0\leqslant i, j<m)$ such that

(a) $y_i\lceil_1^* x_n$ in T_1 and not $y_{i+1}\lceil_1^* x_n$ in T_1,
(b) $y_j\lceil_2^* x_n$ in T_2 and not $y_{j+1}\lceil_2^* x_n$ in T_2.

First we show that $i=j$. Suppose $i>j$ (the case $i<j$ is symmetric). See also Fig. 4. Since T_1 and T_2 are grammatical trees and since we have no ε-productions there exist $\lambda(y_i)\to\beta\lambda(y_{i+1})$ and $\lambda(y_i)\to\beta\lambda(y_{i+1})\phi$ in P, for some $\phi\in V^+$ and $\beta\in V^*$.

Notice that $\phi\neq\varepsilon$ since $\lambda_1(x_n)\in\mathrm{FIRST}(\phi)$. Tree T_1 corresponds with a derivation $\mathrm{rt}(T_1)\overset{*}{\Rightarrow}_l w\lambda(y_i)\alpha_1\Rightarrow_l w\beta\lambda(y_{i+1})\phi\alpha_1\overset{*}{\Rightarrow}_l{}^{(n-1)}\mathrm{fr}(T_1)\phi\alpha_1\overset{*}{\Rightarrow}_l\mathrm{fr}(T_1)$, for some $w\in\Sigma^*$ and $\alpha_1\in V^*$.

[5] Moreover, the place of the left part languages in the hierarchy of LL(k)-languages becomes interesting. However, a proper treatment requires a rather technical discussion on the role of ε-productions, and therefore we omit more detailed comparisons.

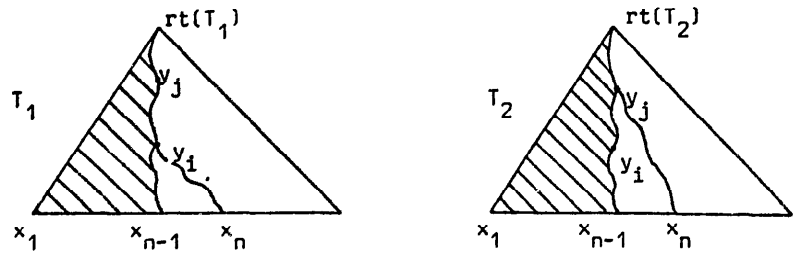

Fig. 4. Trees T_1 and T_2.

Tree T_2 corresponds with a derivation

$$\mathrm{rt}(T_2)\overset{*}{\Rightarrow}_l w\lambda(y_i)\alpha_2\Rightarrow_l w\beta\lambda(y_{i+1})\alpha_2\overset{*}{\Rightarrow}_l {}^{(n-1)}\mathrm{fr}\,(T_2)\alpha_2\overset{*}{\Rightarrow}_l \mathrm{fr}\,(T_2),$$

for some $w\in\Sigma^*$ and $\alpha_2\in V^*$.

Since $\lambda_1(x_n)=\lambda_2(x_n)$ we have that $\mathrm{FIRST}(\alpha_2)\cap\mathrm{FIRST}(\phi)\neq\emptyset$. Since the CFG is reduced and since $\mathrm{rt}(T_1)=\mathrm{rt}(T_2)$ it immediately follows that if P contains $\lambda(y_i)\to\beta\lambda(y_{i+1})$ and $\lambda(y_i)\to\beta\lambda(y_{i+1})\phi$ then P is not prefix (1). Therefore we must conclude that $i=j$.

We proceed with i. There are corresponding nodes, z_1 in T_1 and z_2 in T_2, which will again be treated as identical, hence we omit the indexes, such that $y_i \lceil_1 z$, $y_i \lceil_2 z$, $y_{i+1} \lfloor_1 z$, $y_{i+1} \lfloor_2 z$ and $z \lceil_1^* x_n$ and $z \lceil_2^* x_n$. ${}^{(n)}T_1$ is obtained by adding in an obvious way the path from y_i to x_n to ${}^{(n-1)}T_1$. ${}^{(n)}T_2$ is obtained in an analogous way. Hence there are paths $y_i \lceil_1 z \lceil_1 \cdots \lceil_1 x_n$ and $y_i \lceil_2 z \lceil_2 \cdots \lceil_2 x_n$. Since ${}^{(n-1)}T_1={}^{(n-1)}T_2$ and ${}^{(n)}T_1\neq{}^{(n)}T_2$ these labeled paths are different. Since T_1 and T_2 are grammatical trees there exist productions

$$\lambda(y_i)\to\beta\lambda(y_{i+1})\lambda_1(z)\psi_1 \quad\text{and}\quad \lambda(y_i)\to\beta\lambda(y_{i+1})\lambda_2(z)\psi_2,$$

for some β, ψ_1 and ψ_2 in V^*. If $\lambda_1(z)=\lambda_2(z)$ then V is not chain-independent. If $\lambda_1(z)\neq\lambda_2(z)$, then the necessary condition that $\lambda_1(z)\not\equiv\lambda_2(z)$ is not satisfied. We must conclude that also the case $n>1$ leads to a contradiction. This concludes the "*if*"-part of the proof.

"*only-if*". Let G be a CFG such that J_G satisfies the left part property. Assume that G is not a left part grammar, then, according to Corollary 2.1. there are three possibilities.

(i) V is not chain-independent. Then there is $A\in N$ and $\pi_1,\pi_2\in\mathrm{CH}(A)$, $\pi_1\neq\pi_2$ such that $l(\pi_1)=l(\pi_2)$. Then we can construct trees T_1 and T_2 in J_G with $\mathrm{rt}(T_1)=\mathrm{rt}(T_2)=A$ and where the first leaf of each of the trees has label $l(\pi_1)$. Let the path (and the labeling) from the root of T_1 to the first leaf of T_1 be according to π_1 and the path (and the labeling) from the root of T_2 to the first leaf of T_2 be according to π_2, then ${}^{(1)}\mathrm{fr}\,(T_1)={}^{(1)}\mathrm{fr}\,(T_2)$ and ${}^{(1)}T_1\neq{}^{(1)}T_2$. Contradiction.

(ii) Suppose there exist productions $A\to\alpha X\phi$ and $A\to\alpha Y\psi$ in P, $X\neq Y$, $\alpha\neq\varepsilon$ and X and Y are not mutually chain-independent. Let $w\in L(\alpha)$, where $|w|=n-1$.

Let $\pi_1 \in \mathrm{CH}(X)$, $\pi_2 \in \mathrm{CH}(Y)$ and $l(\pi_1) = l(\pi_2)$. Obviously there exist trees T_1 and T_2 in J_G with $\mathrm{rt}(T_1) = \mathrm{rt}(T_2) = A$, ${}^{(n-1)}\mathrm{fr}(T_1) = {}^{(n-1)}\mathrm{fr}(T_2) = w$ and ${}^{(n-1)}T_1 = {}^{(n-1)}T_2$. By adding paths corresponding to the chains π_1 and π_2 to ${}^{(n-1)}T_1$ and to ${}^{(n-1)}T_2$ respectively we obtain a situation such that ${}^{(n)}\mathrm{fr}(T_1) = {}^{(n)}\mathrm{fr}(T_2)$ and ${}^{(n)}T_1 \neq {}^{(n)}T_2$. Contradiction.

(iii) Suppose P is not prefix (1). Then there exist productions $A \to \beta$ and $A \to \beta\gamma$, $\gamma \neq \varepsilon$ and there is $a \in T$, $w \in T^*$ and $\alpha \in V^*$ such that $S \overset{*}{\Rightarrow} wA\alpha$ and $a \in \mathrm{FIRST}(\gamma) \cap \mathrm{FIRST}(\alpha)$. Also in this case we can construct trees T_1 and T_2 in J_G, $\mathrm{rt}(T_1) = \mathrm{rt}(T_2) = S$. Let $w_1 \in L(\beta)$ and let $|ww_1|$ be $n-1$. Then we can construct T_1 and T_2 such that ${}^{(n)}\mathrm{fr}(T_1) = {}^{(n)}\mathrm{fr}(T_2) = ww_1a$ and where ${}^{(n)}T_1 \neq {}^{(n)}T_2$, since ${}^{(n)}T_1$ is obtained from ${}^{(n-1)}T_1$ by adding the (rightmost) path from the node corresponding to ${}^{(1)}\alpha$ to the nth leaf of T_1, and ${}^{(n)}T_2$ is obtained by adding to ${}^{(n-1)}T_1 (= {}^{(n-1)}T_2)$ the path from the node corresponding to A to the nth leaf of T_2. Since ${}^{(n)}T_1 \neq {}^{(n)}T_2$ we have again a contradiction with the left part property. This concludes the "only if"-part of the proof.

With this theorem we conclude this section and this paper.

Acknowledgments

The author gratefully acknowledges some suggestions and corrections of a referee.

References

[1] A.V. Aho and J.D. Ullman, The Theory of Parsing, Translation and Compiling. Vol. 1 and 2 (Prentice-Hall, Englewood Cliffs, NJ, 1972 and 1973).

[2] F.L. DeRemer, Simple LR(k) grammars, Comm. ACM 14 (1971) 453–460.

[3] M. Geller, M.A. Harrison and I.M. Havel, Normal forms of deterministic grammars, Discr. Math. 16 (1976) 313–322.

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

[5] M.A. Harrison and I.M. Havel, On the parsing of deterministic languages, J. Assoc. Comput. Mach. 21 (1974) 525–548.

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

[7] D.E. Knuth, The Art of Computer Programming, Vol. 1 (Addison-Wesley, Reading, MA, 1968).

[8] A.J. Korenjack and J.E. Hopcroft, Simple deterministic languages, in: IEEE Conference Record of the Seventh Annual Symposium on Switching and Automata Theory (1966), 34–46.

[9] J. Kràl, Bottom up versus top down syntax analysis revised, Research report U VT 10–11/74, Inst. of Computation Technique, Technical University of Prague (1974).

[10] A. Nijholt, Simple chain grammars, in Proceedings 4th Int. Coll. on Automata. Languages and Programming (1977) 352–364.