LEFTMOST SIMPLE MATRIX GRAMMARS

Ondřej Soukup
Doctoral Degree Programme (1), FIT BUT
E-mail: xsouku09@stud.fit.vutbr.cz

Supervised by: Alexander Meduna
E-mail: meduna@fit.vutbr.cz

Abstract: The recent study concentrates on the simple matrix grammars using the leftmost derivations. It defines several leftmost derivation modes and investigate their influence on the generative power of simple matrix grammars. We especially focus on the limiting of the number of components of simple matrix grammars.

Keywords: simple matrix grammars, leftmost derivations, generative power

1 INTRODUCTION

Simple matrix grammars were introduced in [1] by O. H. Ibarra in 1970. Several studies on their generative power and their leftmost variants were published during the next years. Later on, they start disappearing from the forefront. In this paper, we continue with the study of simple matrix grammars, namely with the leftmost simple matrix grammars introduced in [3] by H. A. Maurer.

Primarily, we refine the definition of leftmost simple matrix grammars. Three leftmost derivation modes of simple matrix grammars are defined and investigated. The definition of the derivation mode does not influence the definition of the grammar, only the way the derivation steps are performed may be changed. As we prove next, it can significantly influence the generative power without changing the definition of the used model. While, without leftmost derivations it is not even possible to generate all type 1 languages, with some of the defined leftmost derivation modes it is possible, and moreover, the number of necessary components is decreased to 2 or 3.

Due to the shortage of space, this contribution does not include whole proofs of all results.

2 PRELIMINARIES

We assume that the reader is familiar with formal language theory (see [5, 6]) especially with regulated grammars (see [4]). Let V be an alphabet (finite nonempty set). V^* is the set of all strings over V. Algebraically, V^* represents the free monoid generated by V under the operation of concatenation. The unit of V^* is denoted by ε. Set $V^+ = V^* - \{\varepsilon\}$. Algebraically, V^+ is thus the free semigroup generated by V under the operation of concatenation. For $w \in V^*$, $|w|$ denotes the length of w. The alphabet of w, denoted by $\mathrm{alph}(w)$, is the set of symbols appearing in w.

Let ρ be a relation over V^*. The transitive and transitive and reflexive closure of ρ are denoted ρ^+ and ρ^*, respectively. Unless we explicitly stated otherwise, we write $x \,\rho\, y$ instead $(x, y) \in \rho$.

The families of context-free, context-sensitive and recursively enumerable languages are denoted by **CF**, **CS** and **RE**, respectively.

3 DEFINITIONS AND EXAMPLES

In this section, we define simple matrix grammars and their leftmost variants.

Definition 1. Let $n \geq 1$. A *simple matrix grammar of degree n* (${}_nSMG$ for short) is an (n+3)-tuple

$$G_n = (N_1, N_2, \ldots, N_n, \Sigma, P, S), \text{where}$$

(1) $N_1, \ldots, N_n$ are finite nonempty pairwise disjoint sets of nonterminal symbols;

(2) Σ is a finite nonempty set of terminal symbols, $\Sigma \cap N_i = \emptyset$, for $1 \leq i \leq n$;

(3) S is not in $N_1 \cup \cdots \cup N_n \cup \Sigma$ and is called the *start symbol*;

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

   (a) $(S) \to (v)$, $v \in \Sigma^*$.
   (b) $(S) \to (v_1 v_2 \ldots v_n)$, $v_i \in (N_i \cup \Sigma)^*$, $\text{alph}(v_i) \cap N_i \neq \emptyset$, for $1 \leq i \leq n$.
   (c) $(A_1, A_2, \ldots, A_n) \to (v_1, v_2, \ldots, v_n)$, $A_i \in N_i$, $v_i \in (N_i \cup \Sigma)^*$, for $1 \leq i \leq n$.

□

Definition 2. Let
$$G_n = (N_1, N_2, \ldots, N_n, \Sigma, P, S)$$
be an ${}_nSMG$, for some $n \geq 1$. Consider some rule $(S) \to (w) \in P$, then, $S \Rightarrow w$ is an *initial derivation step*. Consider any string $u_1 A_1 w_1 \ldots u_n A_n w_n$, where $v_i w_i \in (N_i \cup \Sigma)^*$, $A_i \in N_i$, and some rule r

$$r : (A_1, \ldots, A_n) \to (v_1, \ldots, v_n)$$

where $v_i \in (N_i \cup \Sigma)^*$, for $1 \leq i \leq n$. Then, G_n makes a *derivation step* by the rule r

$$w = u_1 A_1 w_1 \ldots u_n A_n w_n \Rightarrow u_1 v_1 w_1 \ldots u_n v_n w_n$$

The transitive and transitive and reflexive closures are defined as stated in the section 2. Subsequently,

$$L(G_n) = \{x \mid S \Rightarrow^* x, x \in \Sigma^*\}$$

is the *language generated by* G_n. The *family of all languages generated by* ${}_nSMG$s is denoted by ${}_n\mathbf{SM}$. □

Definition 3. Consider G_n from the previous definition and the derivation step performed by the rule r. Giving additional restrictions we define three *modes of leftmost derivations*:

(1) $A_i \notin \text{alph}(u_i)$, for $1 \leq i \leq n$.

(2) If
$$w = u'_1 B_1 w'_1 u'_2 B_2 w'_2 \ldots U'_n B_n W'_n$$
   where $u'_i, w'_i \in (N_i \cup \Sigma)^*$, $B_i \in N_i$, and for some $j \leq n$: $|u'_i| = |u_i|$, $i < j$, $|u'_j| < |u_j|$, then, in P, there is no applicable rule
$$(B_1, B_2, \ldots, B_n) \to (x_1, x_2, \ldots, x_n)$$

(3) $N_i \cap \text{alph}(u_i) = \emptyset$, for $1 \leq i \leq n$.

If the derivation step is performed by the leftmost mode k derivations, we write ${}_k\Rightarrow$. ${}_nSMG$ using leftmost mode k derivations is denoted by ${}_k\ell_nSMG$ and the family of all languages of ${}_k\ell_nSMG$s by ${}_k\ell_n\mathbf{SM}$. □

Definition 4. Let
$$G_n = (N_1, N_2, \ldots, N_n, \Sigma, P, S)$$
be an ${}_nSMG$, for some $n \geq 1$. Let R be the set of all rules (not only from P) of the following form
$$(A_1, A_2, \ldots, A_j, \ldots, A_n) \rightarrow (v_1, v_2, \ldots, A_j, \ldots, v_n)$$
where $A_i \in N_i$, $v_i \in (N_i \cup \Sigma)^*$, for $1 \leq i \leq n$, $1 \leq j \leq n$, for all $A_j \in N_j$. For brevity, we denote R
$$(A_1, A_2, \ldots, A_{j-1}, -, A_{j+1}, \ldots, A_n) \rightarrow (v_1, v_2, \ldots, v_{j-1}, -, v_{j+1} \ldots, v_n)$$
□

4 RESULTS

In this section, we investigate the generative power of simple matrix grammars with the defined derivation modes with respect to the number of components. Mode 1 and mode 3 of leftmost derivations were already studied before, therefore, due to the shortage of space, only the achieved results with references are briefly stated.

Context-free grammar is in fact ${}_1SMG$, therefore, the next corollary can be trivially proven.

Corollary 1. ${}_i\ell_1\mathbf{SM} = \mathbf{CF}$, for $i \in \{1,2,3\}$. □

4.1 Mode 1

In [1] and [2], it was proved, ${}_1\ell_n\mathbf{SM} \subset {}_1\ell_{n+1}\mathbf{SM}$, thus it is not possible to find some limit k of the number of components, such that ${}_1\ell_i\mathbf{SM} \subseteq {}_1\ell_k\mathbf{SM}$, for any $i \geq 0$. More precisely, it was proved
$$\mathbf{CF} = {}_1\ell_1\mathbf{SM} \subseteq {}_1\ell_2\mathbf{SM} \subseteq {}_1\ell_3\mathbf{SM} \subseteq \cdots \subset \mathbf{CS}$$

4.2 Mode 2

Lemma 1. *Consider any recursively enumerable language L. Then, there exists ${}_2\ell_3SMG$ G, where $L(G) = L$.*

Proof. Let $L \subseteq \Sigma^*$ be any recursively enumerable language. L can be represented as $L = h(L_1 \cap L_2)$, where $h : T^* \rightarrow \Sigma^*$ is a morphism and L_1 and L_2 are two context-free languages. Then, there exist context-free grammars $G_i = (N_i, T, P_i, S_i)$, where $L(G_i) = L_i$, for $i = 1, 2$. Without any loss of generality, assume that $N_1 \cap N_2 = \emptyset$. Let $T = \{a_1, \ldots, a_n\}$ and $0, 1, \overline{0}, \overline{1}, S_3, F, X \notin (N_1 \cup N_2 \cup T \cup \Sigma)$ be the new symbols. Consider the morphisms

(1) $c : a_i \mapsto 10^i; \overline{c} : a_i \mapsto \overline{10}^i$;

(2) $\pi_1 : N_1 \cup T \mapsto N_1 \cup \Sigma \cup \{0,1\}$, $\begin{cases} A \mapsto A, & A \in N_1, \\ a \mapsto h(a)c(a), & a \in T; \end{cases}$

(3) $\pi_2 : N_2 \cup T \mapsto N_2 \cup \{\overline{0}, \overline{1}\}$, $\begin{cases} A \mapsto A, & A \in N_2, \\ a \mapsto \overline{c}(a), & a \in T; \end{cases}$

(4) $o : \overline{a} \mapsto a, a \in \{\overline{0}, \overline{1}\}$;

(5) $t_1 : \Sigma \cup \{0,1\} \rightarrow \{0,1,\varepsilon\}$, $\begin{cases} a \mapsto a, & a \in \{0,1\}, \\ a \mapsto \varepsilon, & a \notin \{0,1\}; \end{cases}$

(6) $t_2 : \Sigma \cup \{0,1\} \rightarrow \Sigma \cup \{\varepsilon\}$, $\begin{cases} a \mapsto \varepsilon, & a \in \{0,1\}, \\ a \mapsto a, & a \notin \{0,1\}. \end{cases}$

Finally, let $G = (N_1', N_2', N_3', \Sigma, P, S)$ be ${}_2\ell_3SMG$, where $S \notin N_1' \cup N_2' \cup N_3'$ and

- $N_1' = N_1 \cup \{0, 1\}$
- $N_2' = N_2 \cup \{\bar{0}, \bar{1}\}$
- $N_3' = \{S_3, F, X\}$

Construct P as follows. Initially, set $P = \emptyset$. Perform (1) through (5), given next:

(1) add $(S) \rightarrow (S_1 S_2 S_3)$ to P;

(2) for each $(A_1) \rightarrow (w_1) \in P_1$ and for each $(A_2) \rightarrow (w_2) \in P_2$, add $(A_1, A_2, S_3) \rightarrow (\pi_1(w_1), \pi_2(w_2), F)$ to P;

(3) for each $(A_1) \rightarrow (w_1) \in P_1$, add $(A_1, -, S_3) \rightarrow (\pi_1(w_1), -, S_3)$ to P;

(4) for each $(A_2) \rightarrow (w_2) \in P_2$, add $(-, A_2, S_3) \rightarrow (-, \pi_2(w_2), S_3)$ to P;

(5) add

(a) $(0, \bar{0}, F) \rightarrow (\varepsilon, \varepsilon, F)$,
(b) $(1, \bar{1}, F) \rightarrow (\varepsilon, \varepsilon, F)$,
(c) $(0, \bar{1}, F) \rightarrow (\varepsilon, \varepsilon, X)$,
(d) $(1, \bar{0}, F) \rightarrow (\varepsilon, \varepsilon, X)$,
(e) $(0, \bar{0}, F) \rightarrow (\varepsilon, \varepsilon, \varepsilon)$ to P.

Every derivation of G starts with the application of the rule from (1). Next, G simulates the leftmost derivations of both G_1, G_2, respectively, by the rules from (2) through (4). The rules from (2) simulates applications of some rules simultaneously in both G_1 and G_2, while the rules from (3) and (4) only in G_1 or G_2, respectively, until any rule from (2) is applied.

$$S \;{}_2\!\Rightarrow S_1 S_2 S_3 \;{}_2\!\Rightarrow^* w_1 w_2 F$$

Then, the rules from (5) are the only applicable, thus, without any loss of generality, we suppose $\text{alph}(w_1) = \{0, 1\} \cup \Sigma$ and $\text{alph}(w_2) = \{\bar{0}, \bar{1}\}$. Additionally, since the nonterminals from $\{0, 1, \bar{0}, \bar{1}\}$ are erased one by one simultaneously from both w_1, w_2, respectively, the derivation, where $|t_1(w_1)| \neq |w_2|$, is obviously not terminating.

Notice, hereafter, there is always applicable rule for any combination of leftmost nonterminals, however, application of the rules from (5c) and (5d) inserts the symbol X and blocks the derivation.

Consider two possible cases:

(1) $t_1(w_1) = t_1(o(w_2))$. Then, $t_2(w_1) \in L$ and by the sequence of applications of the rules from (5a) and (5b) and finally the rule from (5e)

$$w_1 w_2 F \;{}_2\!\Rightarrow^* w0\bar{0}F \;{}_2\!\Rightarrow w, w = t_2(w_1)$$

(2) $t_1(w_1) \neq t_1(o(w_2))$. Then, $t_2(w_1) \notin L$ and $w_1 = uav$, $w_2 = ua'v'$, where $a \neq o(a')$. By the rules from (5a) and (5b)

$$w_1 w_2 F \;{}_2\!\Rightarrow^* ava'v'F$$

Next, the rule from (5c) or (5d) is the only applicable, which blocks the derivation and no terminal string can be generated.

We covered all possibilities and the lemma holds. □

Corollary 2. ${}_2\ell_3$**SM** = **RE** □

4.3 MODE 3

In [3], the next corollary was proved.

Corollary 3. ${}_3\ell_2\mathbf{SM} = \mathbf{RE}$. □

5 CONCLUSION

This study is part of more extensive work, which is currently being prepared and will cover all defined leftmost derivation modes in greater detail. Additionally, it seems the presented result can be improved, which is going to be the aim of the future effort.

ACKNOWLEDGEMENT

This work was supported by the BUT FIT grant FIT-S-14-2299.

REFERENCES

[1] Ibarra, O.H.: Simple matrix languages. Information and Control 17, 359–394 (1970)

[2] Kuich, W., Maurer, H.A.: Tuple languages. In: International Computing Symposium, Bonn. pp. 881–891 (1970)

[3] Maurer, H.A.: Simple matrix languages with a leftmost restriction. Information and Control 23, 128–139 (1973)

[4] Meduna, A., Zemek, P.: Regulated Grammars and Their Transformations. Faculty of Information Technology, Brno University of Technology (2010)

[5] Rozenberg, G., Salomaa, A. (eds.): Handbook of Formal Languages, Vol. 1: Word, Language, Grammar. Springer, New York (1997)

[6] Salomaa, A.: Formal Languages. Academic Press, London (1973)