Journal of Universal Computer Science, vol. 7, no. 9 (2001), 783-793
submitted: 22/10/00, accepted: 20/7/01, appeared: 28/9/01 © Springer Pub. Co.

Codifiable Languages and the Parikh Matrix Mapping[1]

Adrian Atanasiu
(Faculty of Mathematics, Bucharest University
Academiei 14, R-70109, Bucureşti, Romania
aadrian@pcnet.ro)

Carlos Martín-Vide
(Research Group on Mathematical Linguistics, Rovira i Virgili University
Pl. Imperial Tàrraco 1, 43005 Tarragona, Spain
cmv@correu.urv.es)

Alexandru Mateescu
(Faculty of Mathematics, Bucharest University
Academiei 14, R-70109, Bucureşti, Romania
alexmate@pcnet.ro)

Abstract: We introduce a couple of families of codifiable languages and investigate properties of these families as well as interrelationships between different families. We also develop an algorithm based on the Earley algorithm to compute the values of the inverse of the Parikh matrix mapping over a codifiable context-free language. Finally, an attributed grammar that computes the values of the Parikh matrix mapping is defined.

Key Words: the Parikh mapping, the Parikh matrix mapping, injectivity.
Category: F,G

1 Introduction

In this paper we continue the investigation started in [1] on the injectivity of the restriction of the Parikh matrix mapping to languages.

The Parikh mapping or the Parikh vector was introduced in [3]. The main result concerning this mapping is that the image by the Parikh mapping of a context-free language is always a semilinear set.

The Parikh matrix mapping is an extension of the Parikh mapping introduced in [2]. This extension is based on a special type of matrices. The classical Parikh vector appears in such a matrix as the second diagonal. All other entries above the main diagonal contain information about the order of letters in the original word. All matrices are upper triangular, with 1's on the main diagonal.

Two words with the same Parikh matrix always have the same Parikh vector, but the converse is not true. Hence, the Parikh matrix mapping gives more information about a word than the Parikh vector.

We start with some notations and definitions from the theory of formal languages. The set of all positive integers is denoted by N. Let Σ be an alphabet.

[1] Proceedings of the *First Japan-New Zealand Workshop on Logic in Computer Science*, special issue editors D.S. Bridges, C.S. Calude, M.J. Dinneen and B. Khoussainov.

The set of all words over Σ is Σ^* and the empty word is λ. If $w \in \Sigma^*$, then $|w|$ denotes the length of w. It should cause no confusion that sometimes we use also the customary notation, where vertical bars denote the absolute value of an integer.

In this paper we very often use "ordered" alphabets. An ordered alphabet is an alphabet $\Sigma = \{a_1, a_2, \ldots, a_k\}$ with a relation of order ("$<$") on it. If for instance $a_1 < a_2 < \ldots < a_k$, then we use the notation:

$$\Sigma = \{a_1 < a_2 < \ldots < a_k\}.$$

Let $a \in \Sigma$ be a letter. The number of occurrences of a in a word $w \in \Sigma^*$ is denoted by $|w|_a$.

The *shuffle* operation, denoted by $\sqcup\!\sqcup$, it is defined recursively by :

$$(au \sqcup\!\sqcup bv) = a(u \sqcup\!\sqcup bv) \cup b(au \sqcup\!\sqcup v),$$

and

$$(u \sqcup\!\sqcup \lambda) = (\lambda \sqcup\!\sqcup u) = \{u\},$$

where u ,$v \in \Sigma^*$ and a ,$b \in \Sigma$.

Let u, v be words over Σ. The word u is a scattered subword of v if there exists a word t such that $v \in u \sqcup\!\sqcup t$. If $u, v \in \Sigma^*$, then the number of occurrences of u in v as a scattered subword is denoted by $|v|_{scatt-u}$.

Partially overlapping occurrences of a word as a scattered subword are counted as distinct occurrences. For instance, $|acbb|_{scatt-ab} = 2$ and $|bacb|_{scatt-ab} = 1$.

Let $\Sigma = \{a_1 < a_2 < \ldots < a_k\}$ be an ordered alphabet. The Parikh mapping is a mapping:

$$\Psi : \Sigma^* \to N^k,$$

defined as:

$$\Psi(w) = (|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_k}).$$

The Parikh vector of w is $(|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_k})$. Note that the Parikh mapping Ψ is a morphism from the monoid $(\Sigma^*, \cdot, \lambda)$ to the monoid $(N^k, +, (0, 0, \ldots, 0))$. The mirror of a word $w \in \Sigma^*$, denoted $mi(w)$, is defined as: $mi(\lambda) = \lambda$ and $mi(b_1 b_2 \ldots b_n) = b_n \ldots b_2 b_1$, where $b_i \in \Sigma$, $1 \leq i \leq n$.

A word w is a *palindrome* iff $w = mi(w)$.

Let Σ and Δ be two alphabets such that $\Sigma \subset \Delta$. A *weak identity* is a morphism f from Δ^* to Δ^*, such that $f(a) = a$ for all $a \in \Sigma$ and $f(b) = \lambda$ for all $b \in \Delta - \Sigma$.

For more results and notions of formal languages, see [4].

Now we recall the notion of the Parikh matrix mapping.

Consider a special type of matrices, called *triangle matrices*. A triangle matrix is a square matrix $m = (m_{i,j})_{1 \leq i,j \leq k}$, such that $m_{i,j} \in N$, for all $1 \leq i, j \leq k$, $m_{i,j} = 0$, for all $1 \leq j < i \leq k$, and, moreover, $m_{i,i} = 1$, for all $1 \leq i \leq k$.

The set of all these matrices is denoted by $\mathcal{M}$.

Comment. The set of all triangle matrices of dimension $k \geq 1$ is denoted by $\mathcal{M}_k$. The set $\mathcal{M}_k$ is a monoid with respect to multiplication of matrices and has a unit which is the unit matrix of dimension k.

The notion of the Parikh matrix mapping was introduced in [2].

Definition 1.1 *Let $\Sigma = \{a_1 < a_2 < \ldots < a_k\}$ be an ordered alphabet, where $k \geq 1$. The Parikh matrix mapping, denoted Ψ_{M_k}, is the morphism:*

$$\Psi_{M_k} : \Sigma^* \to \mathcal{M}_{k+1},$$

defined as follows:

If $\Psi_{M_k}(a_q) = (m_{i,j})_{1 \leq i,j \leq (k+1)}$, then for each $1 \leq i \leq (k+1)$, $m_{i,i} = 1$, $m_{q,q+1} = 1$ and all other elements of the matrix $\Psi_{M_k}(a_q)$ are zero.

□

Example 1.1 *Let Σ be the ordered alphabet $\{a < b\}$ and assume that $w = abbab$. Note that $\Psi_{M_2}(w)$ is a 3×3 triangle matrix that can be computed as follows:*

$$\Psi_{M_2}(abbab) = \Psi_{M_2}(a)\Psi_{M_2}(b)\Psi_{M_2}(b)\Psi_{M_2}(a)\Psi_{M_2}(b) =$$

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} =$$

$$= \begin{pmatrix} 1 & 2 & 4 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{pmatrix}$$

However, if Σ is the ordered alphabet $\{a < b < c\}$ and $w' = babbc$, then one can easily verify that:

$$\Psi_{M_3}(w') = \Psi_{M_3}(babbc) = \Psi_{M_3}(b)\Psi_{M_3}(a)\Psi_{M_3}(b)\Psi_{M_3}(b)\Psi_{M_3}(c) =$$

$$= \begin{pmatrix} 1 & 1 & 2 & 2 \\ 0 & 1 & 3 & 3 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

□

The next theorem shows the basic property of the Parikh matrix mapping, see [2].

Notation Consider the ordered alphabet $\Sigma = \{a_1 < a_2 < \ldots < a_k\}$, where $k \geq 1$. We denote by $a_{i,j}$ the word $a_i a_{i+1} \ldots a_j$, where $1 \leq i \leq j \leq k$. □

Theorem 1.1 *Let $\Sigma = \{a_1 < a_2 < \ldots < a_k\}$ be an ordered alphabet, where $k \geq 1$ and assume that $w \in \Sigma^*$. The matrix $\Psi_{M_k}(w) = (m_{i,j})_{1 \leq i,j \leq (k+1)}$ has the following properties:*

(i) $m_{i,j} = 0$, for all $1 \leq j < i \leq (k+1)$,

(ii) $m_{i,i} = 1$, for all $1 \leq i \leq (k+1)$,

(iii) $m_{i,j+1} = |w|_{scatt-a_{i,j}}$, for all $1 \leq i \leq j \leq k$.

□

2 Codifiable and partially codifiable languages

In this section we introduce the notion of codifiable and partially codifiable language.

Definition 2.1 *Let Σ be an alphabet with $card(\Sigma) = k$. A language $L \subseteq \Sigma^*$ is:*

(i) codifiable if for each order on Σ the Parikh matrix mapping $\Psi_M : L \longrightarrow \mathcal{M}_{k+1}$ is injective,
(ii) partially codifiable if there is at least one order on Σ such that the corresponding Parikh matrix mapping is injective in L.

Comment. Obviously, if a language L is codifiable, then L is also partially codifiable. A language that is not partially codifiable is referred to as a *non-codifiable language.*

Proposition 2.1 *If $\Sigma = \{a, b\}$, then a language $L \subseteq \Sigma^*$ is codifiable if and only if L is partially codifiable.*

Proof: For a binary alphabet, only two orders are possible. Assume that $\Sigma = \{a < b\}$ and $\Psi_M : L \longrightarrow \mathcal{M}_3$ is not injective. Let $\alpha, \beta \in L$ be two words such that $\Psi_M(\alpha) = \Psi_M(\beta)$. Note that $|\alpha|_a = |\beta|_a$, $|\alpha|_b = |\beta|_b$ and $|\alpha|_{scatt-ab} = |\beta|_{scatt-ab}$.

Since, for each binary word x, $|x|_{scatt-ba} = |x|_a|x|_b - |x|_{scatt-ab}$, see also [1], it follows that $|\alpha|_{scatt-ba} = |\beta|_{scatt-ba}$. Hence, $\Psi_{M,\circ}(\alpha) = \Psi_{M,\circ}(\beta)$ and therefore L is not codifiable on the ordered alphabet $\Sigma = \{b < a\}$, too. □

The Proposition 2.1 is trivial for the one-letter alphabet, but it is not true if $card(\Sigma) \geq 3$. This follows from the next example.

Example 2.1 *Consider the language:*

$$L = \{(ab)^n c(ba)^n | n \geq 0\} \cup \{(ba)^n c(ab)^n | n \geq 0\}.$$

The basic alphabet is $\Sigma = \{a, b, c\}$. Now consider the order $a < c < b$. The Parikh matrix mapping Ψ_M is not injective. For instance:

$$\Psi_M((ab)^n c(ba)^n) = \Psi_M((ba)^n c(ab)^n) = \begin{pmatrix} 1 & 2n & n & n^2 \\ 0 & 1 & 1 & n \\ 0 & 0 & 1 & 2n \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Now consider the order $a < b < c$ and note that the Parikh matrix mapping is injective:

$$\Psi_M((ab)^n c(ba)^n) = \begin{pmatrix} 1 & 2n & 2n^2 & \frac{n(n+1)}{2} \\ 0 & 1 & 2n & n \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Note that the only word from L with the same Parikh vector is $(ba)^n c(ab)^n$, but this word has another Parikh matrix mapping:

$$\Psi_M((ba)^n c(ab)^n) = \begin{pmatrix} 1 & 2n & 2n^2 & \frac{n(n-1)}{2} \\ 0 & 1 & 2n & n \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The next remark is a representation result. It shows that every language is the image by a weak identity of a partially codifiable language.

Remark 2.1 *Let $\Sigma = \{a_1 < a_2 < \ldots < a_n\}$ be an ordered alphabet and let $L \subseteq \Sigma^*$ be an arbitrary language. Let $\Sigma' = \Sigma \cup \{\sharp\}$, where $\sharp$ is a new letter.*

Then there exists a language $L' \subseteq \Sigma'^*$ and an order on Σ' such that:

(i) L' is partially codifiable on Σ', and
(ii) $h(L') = L$, where h is the weak identity defined by $h(a) = a$, for all $a \in \Sigma$, and $h(\sharp) = \lambda$.

To show the above properties, consider that the order on Σ' is $\Sigma' = \{a_1 < a_2 < \ldots < a_n < \sharp\}$.

Now consider an enumeration of the language L:

$$L = \{w_i \mid i \geq 0\}, \text{ where } w_i \in \Sigma^* \text{ for all } i \geq 0.$$

Define the language L' as:

$$L' = \{w_i \sharp^i \mid i \geq 0\}.$$

Note that if $x, y \in L'$ such that $x \neq y$, then $|x|_\sharp \neq |y|_\sharp$ and hence L' is a partially codifiable language.

Also, it is easy to see that $h(L') = L$.

□

Notations. We denote by $\mathcal{CO}$ the class of all codifiable languages by $\mathcal{NCO}$ the class of non-codifiable languages.

Proposition 2.2 *$\mathcal{NCO}$ is closed under: union, catenation, Kleene star, λ-free morphisms.*

$\mathcal{NCO}$ is not closed under general morphisms.

Proof. The positive assertions are easy to be verified. To show that $\mathcal{NCO}$ is not closed under general morphisms, consider the language $L = \{\alpha \in \{a, b\} |\; |\alpha|_a = |\alpha|_b\}$ and the morphism $h(a) = a,\ h(b) = \lambda$. Then $h(L) = a^*$, which obviously is a codifiable language.

□

Proposition 2.3 *$\mathcal{CO}$ is not closed under union and general morphisms.*

Proof. Consider the languages $L_1 = \{(ab)^n(ba)^n | n \geq 0\}, L_2 = \{(ba)^n(ab)^n | n \geq 0\}$. The alphabet is $\Sigma = \{a, b\}$. The Parikh matrix mapping Ψ_M is injective for both languages, since for the order $a < b$ we obtain:

$$\Psi_M((ab)^n(ba)^n) = \Psi_M((ba)^n(ab)^n) = \begin{pmatrix} 1 & 2n & 2n^2 \\ 0 & 1 & 2n \\ 0 & 0 & 1 \end{pmatrix}$$

(for the reverse order the Parikh matrix mapping remains injective, see Proposition 2.1).

But $L_1 \cup L_2 \notin \mathcal{CO}$ (see also Example 2.1).

Now consider the language $L = \{(acb)^n(bca)^n|n \geq 0\} \cup \{(bca)^n(acb)^n|n \geq 0\}$ over the alphabet $\Sigma = \{a, b, c\}$. It is easy to see that $L \in \mathcal{CO}$. Let h be the morphism defined by $h(a) = a,\ h(b) = b,\ h(c) = \lambda$. Then $h(L) = L_1 \cup L_2 \notin \mathcal{CO}$. □

Comment. The family $\mathcal{CO}$ is closed under intersection with arbitrary languages and the family $\mathcal{NCO}$ is closed under union with arbitrary languages.

From the above results it follows that:

Proposition 2.4 *Let $L_1 \in \mathcal{CO},\ L_2 \in \mathcal{NCO}$ be two languages. Then:*

$$L_1L_2 \in \mathcal{NCO}, \quad L_1 - L_2 \in \mathcal{CO}, \quad L_2 - L_1 \in \mathcal{NCO}.$$

□

Definition 2.2 *Let Σ be a binary ordered alphabet. Two words $\alpha, \beta \in \Sigma^*$ are called palindromicly amicable if the next two assertions hold:*

(i) $\alpha = mi(\alpha),\ \beta = mi(\beta)$, i.e., α and β are palindromes,
(ii) α and β have the same Parikh vector, i.e., $\Psi(\alpha) = \Psi(b)$.

For two words $x, y \in \Sigma^*$, we define the relation $\equiv_{pa}$ as follows:

$x \equiv_{pa} y$ iff there are $\alpha, \beta \in \Sigma^+$ palindromicly amicable such that $x = u\alpha v,\ y = u\beta v$, where u and v are words.

The reflexive and transitive closure of $\equiv_{pa}$ is denoted by $\equiv_{pa}^*$.

Note that the relation $\equiv_{pa}^*$ is a congruence.

In [1], the following theorem is proved

Theorem 2.1 *If $x, y \in \Sigma^*$, where $\Sigma = \{a < b\}$, then:*

$$\Psi_M(x) = \Psi_M(y) \text{ if and only if } x \equiv_{pa}^* y.$$

□

If α is a word, then the equivalence class of α is denoted by $\hat{\alpha}$, i.e., $\hat{\alpha} = \{\beta | \alpha \equiv_{pa}^* \beta\}$.

The class $\mathcal{CO}$ of codifiable languages can be divided into two subclasses:

(i) The class $\mathcal{SCO}$ of *strong codifiable languages*. A language L is in $\mathcal{SCO}$ iff for any $w \in L,\ card(\hat{w}) = 1$.
(ii) The class $\mathcal{WCO}$ of *weak codifiable languages*. A language L is in $\mathcal{WCO}$ iff for all $w \in L$ with $card(\hat{w}) > 1$, it follows that $\hat{w} \cap L = \{w\}$.

Example 2.2 *The language $L = \{a^nb^n|n \geq 0\}$ is a strong codifiable language. A word a^nb^n has no other words palindromicly amicable to it. Thus, a matrix $\begin{pmatrix} 1 & n & n^2 \\ 0 & 1 & n \\ 0 & 0 & 1 \end{pmatrix}$ defines only one word in $\{a < b\}^*$, namely $a^nb^n \in L$.*

Also, all thin languages are strong codifiable.

□

Example 2.3 *The language $L = \{a^i b^2 a^j | 0 \leq i \leq j\}$ is weak codifiable. For instance, if $w = ab^2a^3$, then $\hat{w} = \{ab^2a^3, ba^2ba^2\}$, and thus:*

$$\Psi_M(ab^2a^3) = \Psi_M(ba^2ba^2) = \begin{pmatrix} 1 & 4 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}.$$

Note that $\hat{w} \cap L = \{w\} = \{ab^2a^3\}$.

□

Obviously, $\mathcal{CO} = \mathcal{SCO} \cup \mathcal{WCO}$.

Conjecture: The family $\mathcal{SCO}$ contains only languages where the Parikh vector mapping is injective.

It is not known yet whether there are languages over Σ that are strong codifiable for a peculiar order on Σ and weak codifiable for another order.

□

3 Context-free languages and the Parikh matrix mapping

In this section we present some problems concerning context-free languages and the Parikh matrix mapping.

Let Σ be an ordered alphabet with k letters and $L \subseteq \Sigma^*$ be a codifiable context-free language. Thus:

(i) for each matrix $X \in \mathcal{M}_{k+1}$ there is at most one word $w \in L$ with $\Psi_M(w) = X$ and

(ii) there is a context-free grammar $G = (V_N, \Sigma, S, P)$ with $L(G) = L$.

The following problem, $(P1)$, is important: having a matrix $X \in \mathcal{M}_{k+1}$, Is there a word $w \in L$ such that $\Psi_M(w) = X$?

Note that the above problem $(P1)$ is obviously decidable. For a given matrix $X \in \mathcal{M}_{k+1}$, the set $F_X = \Psi_M^{-1}(X)$ is a finite set. Note that $(P1)$ has a solution if and only if $L \cap F_X \neq \emptyset$. Since L is a context-free language, the last condition is decidable.

In the sequel we present a different method of a smaller complexity. The method is based on the Earley algorithm.

We present the method only for the binary alphabet $\Sigma = \{a < b\}$. It is easy to extend this method to the general case.

Note that each matrix $X \in \mathcal{M}_3$, $X = \begin{pmatrix} 1 & i & k \\ 0 & 1 & j \\ 0 & 0 & 1 \end{pmatrix}$, is completely determined by the vector (i, j, k). If $X = \Psi_M(\alpha)$, where $\alpha \in \Sigma^*$, then $i = |\alpha|_a$, $j = |\alpha|_b$, $k = |\alpha|_{scatt-ab}$.

A generalised Earley configuration is a quadruple:

$$[A \longrightarrow \alpha.\beta, n, (i, j, k), \gamma]$$

where:

- $A \longrightarrow \alpha\beta \in P$;
- $1 \leq n \leq i + j$;
- (i, j, k) corresponds to a matrix from $\mathcal{M}_3$;
- $\gamma \in \Sigma^*$ is a possible prefix of w.

The algorithm enumerates the sets of all configurations I_m $(0 \leq m \leq i+j+1)$. Its formal definition is:

Input:

A context-free grammar $G = (V_N, \{a, b\}, S, P)$ and a matrix:

$$X = \begin{pmatrix} 1 & x & z \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix} \in \mathcal{M}_3.$$

The initial step (the construction of I_0):

The set I_0 contains the configuration $[S' \longrightarrow .\$S, 1, (0, 0, 0), \lambda]$, where $S', \$ \notin V_N$.

The iterative step (the construction of I_{m+1}, $0 \leq m \leq x + y$):

- If $[S' \longrightarrow .\$S, 1, (i, j, k), \gamma] \in I_m$, then $[S' \longrightarrow \$.S, 1, (i, j, k), \gamma] \in I_m$, and all its closures are introduced in I_{m+1}.
- For each configuration $[A \longrightarrow \alpha.a\beta, n, (i, j, k), \gamma] \in I_m$ with $i < x$, the new configuration $[A \longrightarrow \alpha a.\beta, n, (i + 1, j, k), \gamma a]$ and all its closures are introduced in I_{m+1}.
- For each configuration $[A \longrightarrow \alpha.b\beta, n, (i, j, k), \gamma] \in I_m$ with $j < y$ and $k+i \leq z$, the new configuration $[A \longrightarrow \alpha b.\beta, n, (i, j+1, k+i), \gamma a]$ and all its closures are introduced in I_{m+1}.

Note that there are two possible closures of a configuration:

(i) If $[A \longrightarrow \alpha.B\beta, n, (i, j, k), \gamma] \in I_{m+1}$, then $[B \longrightarrow .u, m + 1, (i, j, k), \gamma]$ and its closures will be added to I_{m+1}, for all productions $B \longrightarrow u \in P$.
(ii) If $[A \longrightarrow \alpha., n, (i, j, k), \gamma] \in I_{m+1}$, then we enumerate all configurations $[B \longrightarrow u.Av, p, (i_1, j_1, k_1), \gamma_1] \in I_n$. For each such configuration, we add to I_{m+1} the configuration $[B \longrightarrow uA.v, p, (i, j, k), \gamma]$ and its closures.

The final step:

If $[S' \longrightarrow \$S., 1, (x, y, z), w] \in I_{x+y+1}$, then w is the word from L with $\Psi_M(w) = X$; otherwise, for X there is no $\alpha \in L$ such that $\Psi_M(\alpha) = X$.

Theorem 3.1 *Assume that* $X = \begin{pmatrix} 1 & x & z \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix} \in \mathcal{M}_3$. *If there is a word* $w \in L$ *with* $\Psi_M(w) = X$, *then* $[S' \longrightarrow \$S., 1, (x, y, z), w] \in I_{x+y+1}$; *otherwise,* I_{m+1} *contains no such configuration.*

Proof. If one ignores the last two components of a configuration in the construction we made above, the original Earley algorithm is obtained. In this case $w \in \{a,b\}^*$, $|w| = n$, is a word from L iff $[S' \longrightarrow \$S., 1] \in I_{n+1}$.

The fourth component builds the new Parikh matrix after a new letter is encountered. Namely, if the m-th letter is a, then:

$$\begin{pmatrix} 1 & i & k \\ 0 & 1 & j \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & i+1 & k \\ 0 & 1 & j \\ 0 & 0 & 1 \end{pmatrix}.$$

This means that (i, j, k) is transformed in $(i + 1, j, k)$. If the integer $i + 1$ is greater than the number x of a from the final expected word, then this configuration fails.

In a similar way, if the m-th letter is b, then:

$$\begin{pmatrix} 1 & i & k \\ 0 & 1 & j \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & i & k+i \\ 0 & 1 & j+1 \\ 0 & 0 & 1 \end{pmatrix}.$$

This means that (i, j, k) is transformed in $(i, j + 1, k + i)$. The condition $j + 1 \leq y$ is necessary to bound the number of b to a maximum y.

The last component of a configuration keeps the last letter encountered (a or b). If the algorithm succeeds, then the word w is found and its length is $x + y$. □

Corollary 3.1 *If* $[A \longrightarrow \alpha.\beta, n, (i, j, k), \gamma] \in I_{m+1}$, *then* $\Psi_M(\gamma) = \begin{pmatrix} 1 & i & k \\ 0 & 1 & j \\ 0 & 0 & 1 \end{pmatrix}$.

□

Comment. Note that the above algorithm has the same complexity as the Earley algorithm. Hence it is a deterministic polynomial time algorithm.

Example 3.1 *Let us suppose that* $L = \{a^i bba^j | 0 \leq i \leq j\} \in \mathcal{WCO}$ *is a context-free language generated by the grammar with the rules:*

$$S \longrightarrow aSa, \quad S \longrightarrow Sa, \quad S \longrightarrow bb.$$

We solve the problem $(P1)$ *for the matrix* $X = \begin{pmatrix} 1 & 4 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$.

Thus, the sets of configurations $I_0, \ldots, I_7$, *iteratively generated, are depicted in Figure 1.*

□

I_0 $[S' \longrightarrow .\$S, 1, (0,0,0), \lambda]$

I_1
$[S' \longrightarrow \$.S, 1, (0,0,0), \lambda]$
$[S \longrightarrow .aSa, 1, (0,0,0), \lambda]$
$[S \longrightarrow .Sa, 1, (0,0,0), \lambda]$
$[S \longrightarrow .bb, 1, (0,0,0), \lambda]$

I_2
$[S \longrightarrow a.Sa, 1, (1,0,0), a]$
$[S \longrightarrow .aSa, 2, (1,0,0), a]$
$[S \longrightarrow .Sa, 2, (1,0,0), a]$
$[S \longrightarrow .bb, 2, (1,0,0), a]$
$[S \longrightarrow b.b, 1, (0,1,0), b]$

I_3
$[S \longrightarrow a.Sa, 2, (2,0,0), aa]$
$[S \longrightarrow .aSa, 3, (2,0,0), aa]$
$[S \longrightarrow .Sa, 3, (2,0,0), aa]$
$[S \longrightarrow .bb, 3, (2,0,0), aa]$
$[S \longrightarrow b.b, 2, (1,1,1), ab]$
$[S \longrightarrow bb., 1, (0,2,0), bb]$
$[S' \longrightarrow .\$S, 1, (0,2,0), bb]$
$[S \longrightarrow S.a, 1, (0,2,0), bb]$

I_4
$[S \longrightarrow a.Sa, 3, (3,0,0), a^3]$
$[S \longrightarrow .aSa, 4, (3,0,0), a^3]$
$[S \longrightarrow .Sa, 4, (3,0,0), a^3]$
$[S \longrightarrow .bb, 4, (3,0,0), a^3]$
$[S \longrightarrow b.b, 3, (2,1,2), a^2b]$
$[S \longrightarrow bb., 2, (1,2,2), ab^2]$
$[S \longrightarrow aS.a, 1, (1,2,2), ab^2]$
$[S \longrightarrow S.a, 2, (1,2,2), ab^2]$
$[S \longrightarrow Sa., 1, (1,2,0), b^2a]$
$[S' \longrightarrow \$S., 1, (1,2,0), b^2a]$
$[S \longrightarrow S.a, 1, (1,2,0), b^2a]$

I_5
$[S \longrightarrow a.Sa, 4, (4,0,0), a^4]$
$[S \longrightarrow .aSa, 5, (4,0,0), a^4]$
$[S \longrightarrow .Sa, 5, (4,0,0), a^4]$
$[S \longrightarrow .bb, 5, (4,0,0), a^4]$
$[S \longrightarrow aSa., 1, (2,2,2), ab^2a]$
$[S \longrightarrow Sa., 2, (2,2,2), ab^2a]$
$[S \longrightarrow Sa., 1, (2,2,0), b^2a^2]$
$[S \longrightarrow aS.a, 1, (2,2,0), b^2a^2]$
$[S \longrightarrow S.a, 1, (2,2,0), b^2a^2]$
$[S' \longrightarrow \$S., 1, (2,2,0), b^2a^2]$
$[S' \longrightarrow \$S., 1, (2,2,2), ab^2a]$
$[S \longrightarrow aS.a, 1, (2,2,2), ab^2a]$
$[S \longrightarrow S.a, 1, (2,2,2), ab^2a]$

I_6
$[S \longrightarrow Sa., 1, (3,2,2), ab^2a^2]$
$[S \longrightarrow aSa., 1, (3,2,0), b^2a^3]$
$[S \longrightarrow Sa., 1, (3,2,0), b^2a^3]$
$[S \longrightarrow aSa., 1, (3,2,2), ab^2a^2]$
$[S \longrightarrow S.a, 1, (3,2,0), b^2a^3]$
$[S \longrightarrow S.a, 1, (3,2,2), ab^2a^2]$
$[S' \longrightarrow \$S., 1, (3,2,2), ab^2a^2]$
$[S' \longrightarrow \$S., 1, (3,2,0), b^2a^3]$
$[S' \longrightarrow \$S., 1, (3,2,2), ab^2a^2]$

I_7
$[S \longrightarrow Sa., 1, (4,2,2), ab^2a^3]$
$[S' \longrightarrow \$S., 1, (4,2,2), ab^2a^3]$
$[S \longrightarrow S.a, 2, (4,2,2), ab^2a^3]$
$[S \longrightarrow S.a, 1, (4,2,0), b^2a^4]$
$[S \longrightarrow Sa., 1, (4,2,0), b^2a^4]$
$[S' \longrightarrow \$S., 1, (4,2,0), b^2a^4]$

Figure 1

Theorem 3.2 *Let $\Sigma = \{a < b\}$ be an ordered binary alphabet and consider $L \subseteq \Sigma^*$ a context-free language. There exists a context-free attributed grammar G such that, for each $w \in L$, G computes the Parikh matrix $\Psi_M(w)$.*

Proof. Let $G' = (V_N, \Sigma, S, P)$ be a context-free grammar in the Greibach normal form such that $L(G') = L$. Since Σ is a binary alphabet, the rules from P are of the form: $A \longrightarrow a\alpha,\ A \longrightarrow b\beta,\ A \longrightarrow \lambda$, where $\alpha, \beta \in \Sigma^*$.

Consider a new starting symbol S' and the attributes n_a, n_b, n_{ab}.

The rules from P are extended with the attributes as follows:

$A \longrightarrow a\alpha \qquad n_a := n_a + 1;$
$A \longrightarrow b\beta \qquad n_b := n_b + 1, \quad n_{ab} := n_{ab} + n_a.$
$A \longrightarrow \lambda \qquad$ no attributes.

The first rule is:

$S' \longrightarrow S \qquad n_a := 0,\ n_b := 0,\ n_{ab} := 0.$

It is easy to see that after a leftmost derivation the result is:
$S' \Longrightarrow^* w$ with $|w|_a = n_a,\ |w|_b = n_b,\ |w|_{scatt-ab} = n_{ab}$.

□

Example 3.2 *Consider the language $L = \{a^n b^n | n \geq 0\}$ that is generated by the following context-free grammar in the Greibach normal form:*

$$S \longrightarrow aSB|\lambda, \quad B \longrightarrow b$$

The attributed grammar is:
$S' \longrightarrow S \qquad n_a := 0,\ n_b := 0,\ n_{ab} := 0.$
$S \longrightarrow aSB \qquad n_a := n_a + 1;$
$B \longrightarrow b \qquad n_b := n_b + 1,\ n_{ab} := n_{ab} + n_a;$
$S \longrightarrow \lambda.$
Consider the word aabb with the leftmost derivation:
$S' \Longrightarrow S_{(0,0,0)} \Longrightarrow aSB_{(1,0,0)} \Longrightarrow aaSBB_{(2,0,0)} \Longrightarrow aaBB_{(2,0,0)} \Longrightarrow$
$\Longrightarrow aabB_{(2,1,2)} \Longrightarrow aabb_{(2,2,4)}$.

Hence $\Psi_M(aabb) = \begin{pmatrix} 1 & n_a & n_{ab} \\ 0 & 1 & n_b \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 4 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$.

□

Remark 3.1 *Note that the grammar G is not necessarily an unambiguous grammar.*

Also, note that G is not an attributed grammar in the classical sense. However, one can define an attributed grammar in the classical sense having the same property.

4 Conclusion

We found new properties related to the injectivity of the Parikh matrix mapping. However, most of these properties are proved for binary alphabets. It remains to investigate which of these properties can be extended to alphabets with more than two letters.

References

1. Atanasiu, A., Martín-Vide, C., Mateescu, A., *On the injectivity of the Parikh matrix mapping*, to appear.
2. Mateescu, A., Salomaa, A., Salomaa, K., Yu, S., *A sharpening of the Parikh mapping*, to appear in *RAIRO*.
3. Parikh, R.J., *On the context-free languages*, *Journal of the Association for Computing Machinery*, *13* (1966), 570 – 581.
4. Rozenberg, G., Salomaa, A. (eds.), *Handbook of Formal Languages*, Springer, Berlin, 1997.