University "Alexandru Ioan Cuza" of Iaşi
Faculty of Computer Science

TECHNICAL REPORT

Languages attached to Parikh Matrices

Radu-Florian Atanasiu

TR 10-04, May 2010

ISSN 1224-9327

Universitatea "Alexandru Ioan Cuza" Iaşi
Facultatea de Informatică
Str. Berthelot 16, 6600-Iaşi, Romania
Tel. +40-32-201090, email: bibl@infoiasi.ro

Contents

1 Introduction

This report presents an approach of formal languages based on Parikh matrices. We will display several points of characterizing various aspects of languages using Parikh matrices and the Parikh matrix mapping. The results included in this report are mainly obtained in [Atanasiu, R., 2010].

Let us start with some basic notations and definitions.

Let Σ be a nonempty and finite alphabet and $\mathbb{N}$ the set of nonnegative integers (or natural numbers). Mainly, we will work with an alphabet like $\Sigma = \{a_1, a_2, \ldots, a_s\}$, for which we define an order relation $<$. Without loss of generality, we consider $a_i < a_{i+1}$ for all $1 \leq i \leq s-1$ (if not specified otherwise, we will consider this order relation as implicit on every alphabet we work with from now on). The set of all words over Σ is Σ^*; if λ is the empty word, then the set of nonempty sequences is $\Sigma^+ = \Sigma^* \setminus \{\lambda\}$. For $\alpha \in \Sigma^*$, $|\alpha|$ denotes the length of α. Besides, for any finite set A we denote $|A|$ the number of elements contained by A.

The mirror image of a word $\alpha \in \Sigma^*$, denoted $\mathrm{mi}(\alpha)$, is defined as: $\mathrm{mi}(\lambda) = \lambda$, $\mathrm{mi}(x_1 x_2 \ldots x_n) = x_n \ldots x_2 x_1$, where $x_i \in \Sigma$, $1 \leq i \leq n$. A word α is a *palindrome* if and only if $\alpha = \mathrm{mi}(\alpha)$.

The number of occurrences of a letter $a \in \Sigma$ in a word $\alpha \in \Sigma^*$ is denoted by $|\alpha|_a$. If $u, v \in \Sigma^*$, then the word u is a scattered subword of v if $u = \beta_1 \beta_2 \ldots \beta_r$ and $v = \gamma_0 \beta_1 \gamma_1 \ldots \gamma_{r-1} \beta_r \gamma_r$, for some $r \geq 1$ and $\beta_i, \gamma_j \in \Sigma^*$. We denote by $|\alpha|_u$ the number of occurrences of u in α as a scattered subword. For instance, for $\Sigma = \{a_1, a_2\}$, we have $|a_1 a_2 a_1 a_2|_{a_1 a_2} = 3$. For all $1 \leq i \leq j \leq s$ we denote $a_{i,j} = a_i a_{i+1} \ldots a_j$.

Next we will give the definition of the Parikh mapping:

Definition 1.1. *[Mateescu, A., Salomaa, A., Salomaa, K. & Yu, S., 2001] Let $\Sigma_s = \{a_1, a_2, \ldots, a_s\}$ be an ordered alphabet. The Parikh mapping is a mapping*

$$\Psi : \Sigma^* \longrightarrow \mathbb{N}^s$$

defined as:

$$\Psi(w) = (|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_s}) \text{ , for } w \in \Sigma^*.$$

We say that $(|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_s})$ is the Parikh vector of the word w.

It is important to notice that the Parikh mapping Ψ is a morphism from the monoid $(\Sigma^*, \cdot, \lambda)$ to the monoid $\left(\mathbb{N}^s, +, \underbrace{(0, 0, \ldots, 0)}_{s}\right)$.

We are now ready to formalize the definition of the Parikh matrix mapping:

Definition 1.2. *[Mateescu, A.* et al.*, 2001] Let $\Sigma = \{a_1, a_2, \ldots, a_s\}$ be an ordered alphabet and $\mathcal{M}_{s+1}$ be the multiplicative monoid of $(s+1)$ - dimensional upper-triangular matrices with nonnegative integer entries and unit diagonal. The Parikh matrix mapping, denoted Ψ_s, is the morphism*

$$\Psi_s : \Sigma^* \longrightarrow \mathcal{M}_{s+1}$$

defined by as follows:
if $k = 1, \ldots, s$ and $\Psi_s(a_k) = (m_{i,j})_{1 \leq i,j \leq s+1}$, then for each $1 \leq i \leq s+1$, $m_{i,i} = 1$, $m_{k,k+1} = 1$, all other elements of the matrix $\Psi_s(a_k)$ being 0.

Example 1.3. *Let $\Sigma_3 = \{a, b, c\}$ and $w = abacb$. Then $s = 3$ and $\Psi_3(w)$ is a 4×4 upper triangular matrix that can be computed as follows:*

$$\Psi_3(abacb) = \Psi_3(a)\Psi_3(b)\Psi_3(a)\Psi_3(c)\Psi_3(b)$$
$$= \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 2 & 3 & 1 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

If $|\Sigma| = s$ is fixed, we will denote $\Psi_s(\alpha)$ also by M_α. Also, for simplicity we will denote by Σ_s such an alphabet with an order relation defined as in the previous paragraphs.

A first remark regarding the properties of the Parikh matrix mapping is also a very important one:

Remark 1.4. *The Parikh matrix mapping is not injective. For instance, for $\Sigma_3 = \{a, b, c\}$, the words "*cab*" and "*acb*" have the same image through the Parikh matrix mapping:*

$$\Psi_3(cab) = \Psi_3(acb) = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

A natural question after this remark is: can one determine a natural criterion (non-numerical) for two words to have the same Parikh matrix mapping? Several conditions were discovered, but which do not cover all the existing cases, except for binary alphabets. In fact, it should be pointed out that concerning Parikh matrices, there are immense differences in the phenomena that arise in the case of binary alphabets, compared to higher-order alphabets.

Going further now, we will give a theorem that follows naturally from the definition:

Theorem 1.5. *[Mateescu, A.* et al.*, 2001] Consider $\Sigma_s = \{a_1, a_2, \ldots, a_s\}$ and $w \in \Sigma^*$. The matrix $M_w = \Psi_s(w) = (m_{i,j})_{1 \leq i,j \leq s+1}$ has the following properties:*

- $m_{i,j} = 0$ *for all* $1 \leq j < i \leq s + 1$,
- $m_{i,i} = 1$ *for all* $1 \leq i \leq s + 1$,
- $m_{i,j+1} = |w|_{a_{i,j}}$ *for all* $1 \leq i \leq j \leq s$.

This theorem practically points out the structure of a Parikh matrix. It says that given a word, we can find its corresponding Parikh matrix by calculating the number of appearances only of those subwords composed of consecutive letters in ascending order (bearing in mind the ordering of the alphabet). For instance, the Parikh matrix computed in Example 1.3 for the word $w = abacb$ is:

$$\Psi_3(w) = \begin{pmatrix} 1 & |w|_a & |w|_{ab} & |w|_{abc} \\ 0 & 1 & |w|_b & |w|_{bc} \\ 0 & 0 & 1 & |w|_c \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 1 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

It is obvious that for any words u, v and w such that $w = uv$ we have that $\Psi_s(w) = \Psi_s(u) \cdot \Psi_s(v)$, which is equivalent to writing $M_w = M_u \cdot M_v$.

Now it is easy to make the following remark:

Remark 1.6. *[Mateescu, A.* et al.*, 2001] For any word $w \in \Sigma_s^*$, the Parikh matrix $\Psi_s(w)$ has the Parikh vector as the second diagonal, i.e.*
$(m_{1,2}, m_{2,3}, \ldots, m_{s,s+1}) =$
$(|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_s})$.

One might ask if the properties of the Parikh mapping are imported to the Parikh matrix mapping. The answer is generally "NO", and to illustrate this we will analyze perhaps the most famous and used property of the Parikh mapping:

> "If L is a context-free language, then its image through the Parikh mapping is a semilinear set."

Now, what if we take the context-free language $L = \{a^n b^n \mid n \geq 1\}$ over $\Sigma_2 = \{a, b\}$? We get:

$$\Psi_2(a^n b^n) = \begin{pmatrix} 1 & n & n^2 \\ 0 & 1 & n \\ 0 & 0 & 1 \end{pmatrix}$$

Hence, $\Psi_2(L)$ cannot be a semilinear set, therefore the property does not hold for the Parikh matrix mapping.

Having encoded words by matrices means that we have to draw some advantages from this algebraic theory. The road was opened with a series of properties that link the matrices theory to the mirror of the words. These early results represented the startup of the theory towards several directions, such as the study of injectivity of the Parikh matrix mapping or the relation to languages - to mention only such two possibilities.

We will start by noticing that Parikh matrices are nonsingular, thus any Parikh matrix M_w (over any ordered alphabet Σ_s) has an inverse matrix M_w^{-1}. This comes as no surprise, since the Parikh matrices of order higher than two is a noncommutative group with respect the operation of standard multiplication, and having the unit element the unit matrix I_{s+1}.

Definition 1.7. *[Mateescu, A.* et al.*, 2001] Let w be a word over the alphabet Σ_s and $M_w = (m_{i,j})_{1 \leq i,j \leq s+1}$ its Parikh matrix. The alternate Parikh matrix of w, denoted $\overline{M_w}$, is the matrix $(m'_{i,j})_{1 \leq i,j \leq s+1}$, where $m'_{i,j} = (-1)^{i+j} m_{i,j}$, for all $1 \leq i, j \leq s+1$.*

Before going further, please note that a word w and its mirror $\mathrm{mi}(w)$ always have the same Parikh vector, but this is not true for Parikh matrices. Only palindromes have this property.

Theorem 1.8. *[Mateescu, A.* et al.*, 2001] $[M_w]^{-1} = \overline{M}_{\mathrm{mi}(w)}$, for all words $w \in \Sigma_s^*$.*

Corollary 1.9. *[Mateescu, A.* et al.*, 2001] Let w be a word over Σ_s and $M_w = (m_{i,j})_{1 \leq i,j \leq s+1}$ its Parikh matrix. Assume that $[M_w]^{-1} = (m'_{i,j})_{1 \leq i,j \leq s+1}$. Then $|mi(w)|_{a_{i,j}} = |m'_{i,j+1}|$[1], for all $1 \leq i, j \leq s$.*

If Theorem 1.8 gives a method for computing the inverse of a Parikh matrix, Corollary 1.9 establishes a sharp relationship between the Parikh matrices of words and those of their mirrors. This relation will prove very useful when characterizing certain classes of words having the same Parikh matrix.

Example 1.10. *Let $\Sigma_3 = \{a, b, c\}$ and the word $w = cbbaa$. Obviously,* $\mathrm{mi}(w) = aabbc$. *We have that:*

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

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

By Theorem 1.8 the inverse matrix of M_{cbbaa} is:

[1] by $|m'_{i,j+1}|$ we understand the absolute value of $m'_{i,j+1}$

$$[M_{cbbaa}]^{-1} = \overline{M}_{aabbc} = \begin{pmatrix} 1 & -2 & 4 & -4 \\ 0 & 1 & -2 & 2 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

There has been another way of describing the Parikh matrix of the mirror of a word, by reversing the alphabet's ordering. Thus, we denote by $<^\circ$ the *dual order* of the order $<$, defined by $a <^\circ b$ iff $b < a$. In the same scope we define the *dual ordered alphabet*, $\Sigma_{s,\circ} = \{a_s < a_{s-1} < \ldots < a_1\}$ for an ordered alphabet $\Sigma_s = \{a_1 < a_2 < \ldots < a_s\}$. The Parikh matrix associated to a word $w \in \Sigma_s^*$ with respect to the dual order on Σ_s is denoted by $M_{s,\circ}(w)$.

Now we introduce the reverse of a triangle matrix: let $M = (m_{i,j})_{1 \leq i,j \leq s}$ be such a matrix. The *reverse* of M, denoted $M^{(rev)}$, is the matrix $M^{(rev)} = (m'_{i,j})_{1 \leq i,j \leq s}$, where $m'_{i,j} = m_{s+1-j,s+1-i}$, for all $1 \leq i < j \leq s$.

One easy way to obtain $M^{(rev)}$ is to reverse in M all diagonals that are parallel to the main diagonal.

Example 1.11. *For* $M = \begin{pmatrix} 1 & 1 & 2 & 6 \\ 0 & 1 & 3 & 4 \\ 0 & 0 & 1 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$, $M^{(rev)} = \begin{pmatrix} 1 & 5 & 4 & 6 \\ 0 & 1 & 3 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$.

Theorem 1.12. *[Mateescu, A.* et al.*, 2001]* $[M_w]^{-1} = \overline{M}_{w,\circ}^{(rev)}$*, for all words* $w \in \Sigma_s^*$.

The next corollary shows the connection between words and their mirror:

Corollary 1.13. *[Mateescu, A.* et al.*, 2001]* $M_{\mathrm{mi}(w)} = M_{w,\circ}^{(rev)}$*, for all words* $w \in \Sigma_s^*$.

As one may see, this method of computing the Parikh matrix of a word's mirror is highly similar to the previous one.

This introductive section presented some basic notions and early results, aiming to familiarize the reader with the concepts of *Parikh matrix mapping* and *Parikh matrix*.

2 Amiability and Languages

In the following we will investigate several languages related to the Parikh matrice mapping and integrate them into the Chomsky hierarchy.

2.1 The Parikh Matrix Mapping of Languages

Suppose that the alphabet Σ_s is minimal for a language $L \subseteq \Sigma_s^*$, meaning that for all $a \in \Sigma_s$ there is a $w \in L$ with $|w|_a \neq 0$.

Theorem 2.1. *Let $\Sigma_2 = \{a, b\}$ be an alphabet and $M \in \mathcal{M}_3$ a fixed Parikh matrix, where $M = (m_{ij})_{1 \leq i,j \leq 3}$. Then the language*

$$L = \{\alpha \mid \alpha \in \Sigma_2^*,\ |\alpha|_{ab} = m_{13}\}$$

is regular.

Proof. Let $k \stackrel{def}{=} m_{13}$. By emphasizing the position of each letter b, any word $\alpha \in L$ can be written as

$$\alpha = b^t a^{x_1} b a^{x_2} b \dots a^{x_n} b a^{x_{n+1}}$$

with $t \geq 0$, $x_1 \geq 1$, $x_i \geq 0$, for $2 \leq i \leq n+1$ and

$$nx_1 + (n-1)x_2 + \dots + x_n = k. \tag{1}$$

For each fixed n, equation (1) has a finite number of nonnegative integer solutions. Moreover, the number of values n for which the equation has distinct solutions is finite ($n \leq k$ can be considered an upper bound).

Therefore the set

$$S(k) = \{(x_1, x_2, \dots, x_n) \in \mathbb{Z}_+^n \mid nx_1 + (n-1)x_2 + \dots + x_n = k\}$$

is finite. For each solution $(x_1, x_2, \dots, x_n) \in S(k)$, the language

$$L_{x_1,\dots,x_n} = b^* a^{x_1} b a^{x_2} b \dots a^{x_n} b a^*$$

is obviously regular.

As

$$L = \bigcup_{(x_1,\dots,x_n) \in S(k)} L_{x_1,\dots,x_n}$$

it will mean that $L \in \mathcal{L}_3$. □

In fact it is easy to see that if we consider an alphabet Σ_s of arbitrary cardinality (though greater than 2) and if we fix one element from the third diagonal, the resulting language is regular. This can be formalized as a corollary:

Corollary 2.2. *Let $\Sigma_s = \{a_1, a_2, \dots, a_s\}$ be an alphabet and $M \in \mathcal{M}_{s+1}$ a fixed Parikh matrix, where $M = (m_{ij})_{1 \leq i,j \leq s+1}$. Then, given a fixed k, with $1 \leq k \leq s-1$ the language*

$$L_{a_k a_{k+1}} = \{\alpha \mid \alpha \in \Sigma_s^*,\ |\alpha|_{a_k a_{k+1}} = m_{k,k+2}\}$$

is regular.

Remark 2.3. *In the previous corollary and theorem, we did not use the whole matrices, but only specific entries of them. Otherwise the resulting alphabets would be finite, thus obviously regular.*

Now we can extend the initial theorem to a multiple letter alphabet:

Theorem 2.4. *Let $\Sigma_s = \{a_1, a_2, \dots, a_s\}$ an ordered alphabet and $M \in \mathcal{M}_{s+1}$ a fixed Parikh matrix. Then the following language is regular:*

$$L = \{\alpha \mid \alpha \in \Sigma_s^*, M_\alpha \text{ and } M \text{ are identical on the third diagonal}\}$$

Proof. The hypothesis gives the information that the specified language is infinite, due to the fact that no restriction is imposed over the number of a_i-s (for $1 \leq i \leq s$) in the words belonging to the language. However, as no restriction is placed on the number $|\alpha|_{a_i a_{i+j}}$, where $j \geq 2$, it means that the language is the intersection of many smaller languages, namely:

$$L = L_{a_1 a_2} \cap L_{a_2 a_3} \cap \ldots \cap L_{a_{s-1} a_s}$$

where by $L_{a_i a_{i+1}}$ we denote the language $L_{a_i a_{i+1}} = \{\alpha \mid \alpha \in \Sigma_s^*, |\alpha|_{a_i a_{i+1}} = m_{i,i+2}\}$, as defined in Corollary 2.2. Taking into account that each of the languages $L_{a_i a_{i+1}}$ is regular and that s is finite, it follows that L is regular too. □

Theorem 2.5. *Let $L \in \mathcal{L}_1$ be a context-sensitive language over the alphabet $\Sigma_s = \{a_1, a_2, \ldots, a_s\}$ and $M \in \mathcal{M}_{s+1}$ a Parikh matrix. It is decidable whether there exists $w \in L$ having $M_w = M$.*

Proof. Without reducing the generality we can presume that $\lambda \notin L$. Let $G = (V_N, V_T, S, P)$ be a type 1 grammar in the Kuroda normal form[1] such that $L(G) = L$. P will not contain any production of the form $A \longrightarrow \lambda$. We propose the following algorithm:

Algorithm 2.6.

1. *From M we extract the Parikh mapping (the Parikh vector)$\Psi = (x_1, x_2, \ldots, x_s)$ (from the second diagonal, parallel with the main diagonal).*
2. *Let $n = \sum_{i=1}^{s} x_i$, $V_0 = \{S\}$, $i := 1$.*
3. *$V_i = V_{i-1} \cup \{\alpha \mid \exists \beta \in V_{i-1},\ \beta \overset{G}{\Longrightarrow} \alpha, \quad |\alpha| \leq n,\ |\alpha|_{a_i} \leq x_i,\ 1 \leq i \leq s\}$.*
4. *If $V_i \neq V_{i-1}$ then $i := i + 1$ and goto 3.*
5. *$W = V_i \cap \Sigma_s^*$.*
6. *If $W = \emptyset$ then NO, Stop.*
7. *Let $x \in W$. If $M_x = M$ then YES, Stop.*
8. *$W := W \setminus \{x\}$ and goto 6.*

It can be seen that the two loops ($2 - 4$ and $6 - 8$) end after a finite number of steps. All the words from L which have the Parikh mapping Ψ are obtained when exiting the first loop. The second loop explores the set W (which is finite) and specifies the Parikh matrix mapping for each word. If there exists a word whose Parikh matrix mapping is equal to M, then the algorithm ends with the YES message. Otherwise the message is NO. □

[1] A grammar is in **Kuroda normal form** iff all production rules are of the form

$$AB \to CD \text{ or}$$
$$A \to BC \text{ or}$$
$$A \to B \text{ or}$$
$$A \to a,$$

where A, B, C, D are nonterminal symbols and a is a terminal symbol.

2.2 Amiable Extensions of Chomsky languages

Let L be a language over a finite alphabet Σ_s. We will extend this language by adding all the words that are amiable with the elements of L.

Definition 2.7. *[Mateescu, A. & Salomaa, A., 2004] The* matrix extension / amiable extension *of a language L over the alphabet Σ_s consists of all words matrix equivalent to some word in L:*

$$[L] = \{w' \mid w' \sim_a w, w \in L\}.$$

Two languages K and L are matrix equivalent (amiable) *if their matrix extensions coincide: $[K] = [L]$.*

Clearly, we have:

$$[L] = \bigcup_{\alpha \in L} [\alpha]$$

where $[\alpha] = \{w \mid w \sim_a \alpha\}$.

The typical language-theoretic problems that arise are the following:

- Decide the matrix equivalence problem for languages belonging to a specific family of languages $\mathcal{L}$.
- Decide whether or not the matrix extension of a language in $\mathcal{L}$ belongs to another specific family $\mathcal{L}'$ (possibly $\mathcal{L} = \mathcal{L}'$).

For a finite language the matrix (amiable) extension is always finite, making both problems decidable. The following result shows what happens for the case of infinite regular languages.

Theorem 2.8. *[Mateescu, A. & Salomaa, A., 2004] Each of the following three cases is possible for an infinite regular language L.*

1. *$[L]$ is regular.*
2. *$[L]$ is context-free but not regular.*
3. *$[L]$ is not context-free.*

Now we will investigate several properties of the amiable extension, regardless of the class of languages.

Proposition 2.9.

1. *$[[L]] = [L]$;*
2. *$[L_1 \cup L_2] = [L_1] \cup [L_2]$;*
3. *$[L_1][L_2] \subseteq [L_1 L_2]$;*

Proof. 1. This result follows directly from the definition.

2. Let $w \in [L_1 \cup L_2]$. Then there exists $\alpha \in L_1 \cup L_2$ with $M_w = M_\alpha$. If $\alpha \in L_1$, then $w \in [L_1]$; otherwise $w \in [L_2]$. In any case it follows that $w \in [L_1] \cup [L_2]$.
   The inverse implication is similar.

3. Let $w \in [L_1][L_2]$; there exists $w_1 \in [L_1]$, $w_2 \in [L_2]$ so that $w = w_1 w_2$. By definition, there are $\alpha_1 \in L_1$, $\alpha_2 \in L_2$ such that $M_{w_i} = M_{\alpha_i}$, $i = 1, 2$.
   Then

   $M_w = M_{w_1 w_2} = M_{w_1} M_{w_2} = M_{\alpha_1} M_{\alpha_2} = M_{\alpha_1 \alpha_2}$, following that $w \in [L_1 L_2]$.

   The inverse inclusion is not true, as it is shown in the following example:
   $L_1 = \{ab\}$, $L_2 = \{ba\}$. Then $L_1 L_2 = \{abba\}$ and $[L_1 L_2] = \{abba, baab\}$, but $[L_1][L_2] = L_1 L_2 = \{abba\} \subset [L_1 L_2]$.

□

Corollary 2.10.

1. $L_1 \subseteq L_2 \implies [L_1] \subseteq [L_2]$,

2. $[L]^* \subseteq [L^*]$.

Proof. 1. $L_1 \subseteq L_2 \implies L_2 = L_1 \cup L_2$.
Therefore $[L_2] = [L_1 \cup L_2] = [L_1] \cup [L_2]$, following that $[L_1] \subseteq [L_2]$.

2. Using the third point from Proposition 2.9, the inclusion $[L]^i \subseteq [L^i]$ follows by induction.

□

Proposition 2.11. $[L_1 \cap L_2] \subseteq [L_1] \cap [L_2]$.

Proof. For $L_1 \cap L_2 = \emptyset$ the statement is obvious.
Let us assume that $L_1 \cap L_2 \neq \emptyset$. Let $w \in [L_1 \cap L_2]$. Then $\exists \alpha \in L_1 \cap L_2$ with $w \sim_a \alpha$.
From $\alpha \in L_1$ and $w \sim_a \alpha$ it follows that $w \in [L_1]$.
From $\alpha \in L_2$ and $w \sim_a \alpha$ it follows that $w \in [L_2]$.
Therefore $w \in [L_1] \cap [L_2]$.
The inverse inclusion is not true. The example given below emphasizes this. Let $\Sigma_3 = \{a, b, c\}$ be an alphabet and the languages $L_1 = \{abba, ac\}$, $L_2 = \{baab, ac\}$ defined on Σ_3.
Then $[L_1 \cap L_2] = \{ac, ca\}$, and $[L_1] \cap [L_2] = \{abba, baab, ac, ca\}$. □

Proposition 2.12. *$L \subset [L]$ iff $[L] \cap [\overline{L}] \neq \emptyset$, where $\overline{L} = \Sigma_s^* \setminus L$.*

Proof. "$\Longrightarrow$": Let $w \in [L] \setminus L$. Due to the fact that $w \notin L$, $w \in \overline{L} \subseteq [\overline{L}]$ follows.
But $w \in [L]$, thus $w \in [L] \cap [\overline{L}]$.
"$\Longleftarrow$": Let $w \in [L] \cap [\overline{L}]$. We presume that $L = [L]$. We will get:

- $\exists \alpha \in L, \quad w \sim_a \alpha$;
- $\exists \beta \in \overline{L}, \quad w \sim_a \beta$.

From the above two statements it follows that $\alpha \sim_a \beta$.
But $\alpha \in L$ and $L = [L]$, therefore $\beta \sim_a \alpha$ implies $\beta \in L$, which is a contradiction. □

Corollary 2.13. $L = [L] \iff \overline{L} = [\overline{L}]$.

Now, going a bit more in depth towards analyzing the structures defined earlier we are able to prove some more algebraic properties. An obvious one is given below:

Proposition 2.14. *$L_{\Sigma_s} = \{[L] | L \subseteq \Sigma_s^*\}$ forms a Boolean algebra (with the $\bigcup$ and $\bigcap$ operators).*

If we look at the amiability classes from the languages point of view, we can say that an amiability class acts like an atom for an amiable extended language $[L]$. First we denote an amiability class by $L_\alpha = \{w | w \sim_a \alpha\}$. Now it appears natural to write $[L]$ as an unique union of languages. More specific: $[L] = \bigcup_{\alpha \in L} L_\alpha$. We will also define $\mathcal{A}_L$ as the set of all the atoms of L.

Proposition 2.15. *The set of all the languages characterized by the same set of Parikh matrices (given by a language L) is a join-semilattice.*

Proof. We denote the specified set by $\mathcal{L}_L = \{W | W \subseteq [L], [W] = [L]\}$. The order relation is the inclusion.
In order to prove the proposition we will now analyze when $\mathcal{L}_L$ becomes a lattice and moreover, a bounded one.
First, we will check if any two elements have a supremum. Let $W_1, W_2 \in \mathcal{L}_L$. These languages have a supremum, namely $W_{12} = W_1 \bigcup W_2$. We have the guarantee that this is the least upper bound of W_1 and W_2, from the set's union properties.
Next we will see that any two elements do not necessarily have an infimum. The idea of the proof is this: if L contains at least two words, and if among these words there exist an $\alpha \in L$ such that $|L_\alpha| > 1$, then we can choose W_1 and W_2 such that $W_1 \cap W_2 \notin \mathcal{L}_L$. In this case there is no infimum ($\emptyset$ is not included in $\mathcal{L}_L$ unless L is void), therefore $\mathcal{L}_L$ is not a lattice.
In order to illustrate the above, let's consider the language $L = \{\alpha_1, \alpha_2, \dots, \}$. Each word α_i $(i \geq 1)$ generates a class of amiable words L_{α_i}.

$$\begin{aligned} L_{\alpha_1} &= \{\alpha_{11}, \alpha_{12}, \alpha_{13}, \ldots\} \\ L_{\alpha_2} &= \{\alpha_{21}, \alpha_{22}, \alpha_{23}, \ldots\} \\ &\vdots \end{aligned}$$

Now we can choose for instance a W_1 such that it contains all the α_{j1} elements and a W_2 that contains all the α_{j2} elements ($j \geq 1$). The condition for their intersection to be in $\mathfrak{L}_L$ is to contain at least a word from each L_{α_j}. Therefore they don't have a meet.

Basically, if for any $\alpha \in L$ we have that $|L_\alpha| = 1$ (the atoms that form $[L]$ are singletons) then $\mathfrak{L}_L$ is a lattice. In other words, if $|\mathfrak{L}_L| = 1$ then the lattice conditions are met. In this case, the greatest and the least element would coincide with the only element of the lattice, namely $L = [L]$. □

Next we will look what happens to the extensions of various Chomsky-classified languages over a two-letter alphabet.

Theorem 2.16. *Let $\Sigma = \{a, b\}$ be an alphabet and L a language over this alphabet.*

1. *If $L \in \mathfrak{L}_i$, with $1 \leq i \leq 3$, then $[L] \in \mathfrak{L}_1$.*
2. *$\mathfrak{L}_0$ is closed under the $[\,]$ extension.*

Proof. Let $G = (N, \{a, b\}, S, P)$ the grammar that generates the language L. We will enrich this grammar by adding two new non-terminal symbols and a new set of productions. Let these new terminals be A and B and consider the following set of productions denoted by Q:

$$\begin{aligned} BA &\rightarrow ABX_2 \mid Y_1AB \\ AB &\rightarrow Y_2BA \mid BAX_1 \\ A &\rightarrow a \\ B &\rightarrow b \\ X_iZ &\rightarrow ZX_i \\ ZY_i &\rightarrow Y_iZ \\ X_iY_i &\rightarrow \lambda \end{aligned}$$

where $1 \leq i \leq 2$ and $Z \in \{A, B\}$. Let's also change the set of productions P, by replacing the terminals a and b with the new non-terminals A and B: $P' = \{X \rightarrow [Y]_N | X \rightarrow Y \in P\}$, where $[Y]_N$ is the string Y having the terminals a and b replaced with the non-terminals A and B respectively.

We obtain an alternative grammar: $G' = (N', \{a, b\}, S, P' \cup Q)$, where $N' = N \cup \{A, B\}$ (we consider that the non-terminals A and B did not appear previously in N).

Let $\alpha = \alpha_1 ab \alpha_2 ba \alpha_3$ a word belonging to the language L. The grammar generates this word by using the productions from P' and then the last two productions from Q, which transform the non-terminals A and B into the the terminals a and b respectively. It can be easily seen that the grammar also generates the word $\alpha' = \alpha_1 ba \alpha_2 ab \alpha_3$, which is amiable with α. Iteratively, all the words amiable with α are being generated. We are sure that no other words outside $[L]$ are being generated due to the construction of the grammar:

1. First the rules of the original grammar are used, thus guaranteeing that a word from L is generated if we additionally apply the last two rules from Q.
2. Next we either choose to apply the last two rules from Q or to use the other productions from Q, which flip an equal number of AB-s and BA-s (to be noted that if no corresponding BA is found for an AB, then the same positions are flipped and the original word is obtained in the end).

As G' is context-sensitive, it results that both statements are proven. □

As a conclusion, we have seen that there are various language-theoretic aspects that can be dealt using Parikh matrices. Moreover, we have extended the concept of amiability to languages, by extending in a natural manner the concept of Parikh matrix mapping.

3 Conclusions

The relatively young subject of *Parikh matrix mappings* and, intrinsically of *Parikh matrices*, has given the basic technical tool for investigations concerning the number of occurrences of a word u as a scattered subword of a word w. *Parikh matrices* are a step forward into obtaining a suitable mathematical characterization of words. Theorists' goal is to find a model that is efficient enough to be computed, but at the same time complex enough to describe every single aspect of words. *Parikh matrices* do not claim to fully characterize them, however, they arrive with a multitude of properties that add up to those already-known of both words and words relations.

This report contains a formal analysis of the relation between *Parikh matrices* and languages. In this sense we work with *amiable extensions* of languages. Several properties are shown and further studied. We were also able to discuss the relation with classes of languages from the Chomsky hierarchy.

Bibliography

Atanasiu, A. (2007). Binary amiable words. *Int. J. Found. Comput. Sci.*, **18**, 387–400.

Atanasiu, A. and Atanasiu, R. and Petre, I. (2008). Parikh matrices and amiable words. *Theoretical Computer Science vol. 390, no. 1, pp. 102-109*.

Atanasiu, R. (2010). Parikh matrix mapping and languages. *Int. J. Found. Comput. Sci.*, **In Press, Accepted Manuscript**.

Atanasiu, R. and Manea, F. (2010). How to decide if a matrix is Parikh. *Work in Progress*.

Fossé, S. and Richomme, G. (2004). Some characterisations of Parikh matrix equivalent binary words. *Inf. Processing Letters Vol. 92(2), 77-82*.

Garey, M.R. and Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman & Co Ltd.

Kuroda, S.-Y. (1964). Classes of languages and linear-bounded automata. *Information and Control*, **7**, 207–223.

Lothaire, M. (1983). Combinatorics on words. *Encyclopedia of Mathematics*, **17**.

Mateescu, Al. (2004). Algebraic aspects of Parikh matrices. In *Theory Is Forever*, 170–180.

Mateescu, Al. and Salomaa, A. (2004). Matrix indicators for subword occurrences and ambiguity. *Int. J. Found. Comput. Sci. 15, 277-292*.

Mateescu, Al. and Rozenberg, G. and Salomaa, A. (1998). Shuffle on trajectories: Syntactic constraints. *Theor. Comput. Sci.*, **197**, 1–56.

Mateescu, Al. and Salomaa, A. and Salomaa, K. and Yu, S. (2001). A sharpening of the Parikh mapping. *Theoret. Informatics Appl. 35, 551-564*.

Mateescu, Al. and Salomaa, A. and Yu, S. (2004). Subword histories and Parikh matrices. *J. Comput. Syst. Sci.*, **68**, 1–21.

Parikh, R.J. (1966). On context-free languages. *J. ACM*, **13**, 570–581.

Rozenberg, G. & Salomaa, A., eds. (1997). *Handbook of formal languages, vol. 1-3*. Springer-Verlag New York, Inc., New York, NY, USA.

Salomaa, A. (2003). Counting (scattered) subwords. *Bulletin of the EATCS*, **81**, 165–179.

Salomaa, A. (2005b). On the injectivity of Parikh matrix mappings. *Fundam. Inform.*, **64**, 391–404.

Salomaa, A. (2009). Characteristic words for Parikh matrices. In *Automata, Formal Languages, and Related Topics*, 117–127.

Salomaa, A. (2010). Criteria for matrix equivalence of words. *Theoretical Computer Science*, **411**, 1818 – 1827.

Salomaa, A. and Yu, S. (2006). Subword conditions and subword histories. *Inf. Comput.*, **204**, 1741–1755.

Salomaa, A. and Yu, S. (2010). Subword occurrences, Parikh matrices and lyndon images. *Int. J. Found. Comput. Sci. 21, 91-11*.

Sipser, M. (1996). *Introduction to the Theory of Computation*. Course Technology.