Parikh Matrices and Words over Tertiary Ordered Alphabet

Amrita Bhattacharjee
Department of Computer Science
Assam University Silchar
788011
India

Bipul Syam Purkayastha
Department of Computer Science
Assam University Silchar
788011
India

ABSTRACT
Parikh matrix is a numerical property of a word on an ordered alphabet. It is used for studying word in terms of its sub words. It was introduced by Mateescu et al. in 2000. Since then it has been being studied for various ordered alphabets. In this paper Parikh Matrices over tertiary alphabet are investigated. Algorithm is developed to display Parikh Matrices of words over tertiary alphabet. This algorithm proves a good tool for further investigation of Parikh Matrices of words over tertiary alphabet. A set of equations for finding tertiary words from the respective Parikh matrix is introduced. These equations are useful to find tertiary words from the respective Parikh matrix. Examples are given. Some examples of larger tertiary words are given with their Parikh matrices as result analysis. A distance is defined on classes of M-ambiguous words over tertiary ordered alphabet. It is named as stepping distance. One can compare words by this stepping distance.

Keywords
M-ambiguity, Parikh mapping, Parikh matrix, subword, word, Stepping distance

1. INTRODUCTION
Various techniques have been developed to solve complex problems of words over formal languages. Introduced in [1] by R.J.Parikh, Parikh mapping plays a very significant role in this context. The concept of Parikh matrix first introduced in [2] is an extension of Parikh Mapping. A word is a finite or infinite sequence of symbols taken from a finite set called alphabet. Parikh Matrix which is a triangular matrix can be associated with every word over an ordered alphabet. Main diagonal of this matrix takes only the value 1 and every entry below the main diagonal has the value 0 but the entries above the main diagonal provide information on the number of certain sub-words in. The concept of subword is the basic idea behind this interesting notion of Parikh matrix. Since the introduction of it a series of papers investigating these matrices has appeared studying various problems related to subwords. A few examples [3, 4, 5… 17] are cited which has used subword occurrences and Parikh matrix for solving the problems of word. Research on Parikh matrices are being extended from binary to ternary words. Recently tertiary ordered alphabet are also being investigated. In this paper an algorithm is introduced for showing Parikh matrices for tertiary words. Here a set of equations over tertiary alphabet is also introduced. From these equations one can find the word corresponding to a 5x5 Parikh matrix. Parikh Matrix faces some challenging problems. The most important of them is that it is not injective. Two words may have the same corresponding Parikh Matrix. This property known as M-ambiguity is a problem in the field of Parikh Matrix. The set of equations introduced in this paper helps us to find the M-ambiguous words. A distance defined on classes of M-ambiguous words is also introduced. One can compare the M-ambiguous words by this distance.

Organization of this paper is as follows. The following section 2 recapitulates the basic preliminaries of Parikh Matrix. Section 3 goes toward developing the algorithm for display Parikh Matrix of a sequence over tertiary alphabet In Section 3, result analysis is also presented. Section 4 gives proposed equations to find out the tertiary sequences corresponding to a given Parikh Matrix. In section 5 a distance between words namely stepping distance defined on classes of M- ambiguous words is introduced. The paper is concluded in Section 6 by summarizing the observations.

2. PRELIMINARY
Throughout this paper N will denote the set of natural numbers including 0. First some definitions are recalled.

Ordered alphabet: an ordered alphabet is a set of symbols $\Sigma = \{a_1, a_2, a_3, \cdots, a_n\}$ where the symbols are arranged maintaining a relation of order ("$<$") on it. For example if $a_1 < a_2 < a_3 < \cdots < a_n$, then one use notation: $\Sigma = \{a_1, a_2, a_3, \cdots, a_n\}$

Word: a word is a finite or infinite sequence of symbols taken from a finite set called an alphabet. Let $\Sigma = \{a_1, a_2, a_3, \cdots, a_n\}$ be the alphabet. The set of all words over Σ is Σ^*. The empty word is denoted by λ.

$|w|_{a_i}$: Let $a_i \in \Sigma = \{a_1, a_2, a_3, \cdots, a_n\}$ be a letter. The number of occurrences of a_i in a word $w \in \Sigma^*$ is denoted by $|w|_{a_i}$.

Sub -word: a word u is a sub- word of a word w, if there exists words $x_1 \cdots x_n$ and $y_0 \cdots y_n$, (some of them possibly empty), such that $u = x_1 \cdots x_n$ and $w = y_0 x_1 y_1 \cdots x_n y_n$.

For example if $w = adbaabcacd$ is a word over the alphabet $\Sigma = \{a, b, c, d\}$ then $baca$ is a sub-word of w. Two occurrences of a sub-word are considered different if they differed by at least one position of some letter. In the word

$w = adbaabcacd$, the number of occurrences of the word $baca$ as a sub-word of w is $|w|_{baca} = 2$.

Triangle matrix: A triangle matrix is a square matrix $m = (m_{ij})_{1\le i,j\le n}$ such that:

1. $m_{ij} \in N \quad (1 \le i, j \le n)$,
2. $m_{ij} = 0$ for all $1 \le j < i \le n,$
3. $m_{ii} = 1 \quad (1 \le i \le n)$.

Parikh matrix: let $\Sigma = \{a_1 < a_2 < a_3 < \cdots < a_n\}$ be an ordered alphabet, where $n \ge 1$.The Parikh matrix mapping, denoted Ψ_{M_n} , is the homomorphism $\Psi_{M_n} : \Sigma^* \to M_{n+1}$ defined as follows:

If $\Psi_{M_n}(a_q) = (m_{ij})_{1\le i,j\le n+1}$

then $m_{i,i} = 1, m_{q,q+1} = 1$ and all other elements are zero.

Parikh matrix of a word:

Let $\Sigma=\{a_1 < a_2 < a_3 \cdots < a_n\}$ be an nth ordered alphabet. The Parikh matrix of $a_1, a_2, a_3 \cdots, a_n$ are as follows:

$$\Psi_{M_n}(a_1)=\begin{pmatrix}1&1&\cdots&0&0\\0&1&0&\cdots&0\\\vdots&\vdots&\cdots&\vdots&\vdots\\0&0&\cdots&1&0\\0&0&\cdots&0&1\end{pmatrix}_{(n+1)\times(n+1)}, \Psi_{M_n}(a_2)=\begin{pmatrix}1&0&\cdots&0&0\\0&1&1&\cdots&0\\\vdots&\vdots&\cdots&\vdots&\vdots\\0&0&\cdots&1&0\\0&0&\cdots&0&1\end{pmatrix}_{(n+1)\times(n+1)}$$

$$,\cdots \quad \Psi_{M_n}(a_n)=\begin{pmatrix}1&0&\cdots&0&0\\0&1&\cdots&0&0\\\vdots&\vdots&\cdots&\vdots&\vdots\\0&0&\cdots&1&1\\0&0&\cdots&0&1\end{pmatrix}_{(n+1)\times(n+1)}$$

Any word w over the n[th] order alphabet has a unique Parikh Matrix. This matrix is given by

$$\Psi_{M_n}(w)=\begin{pmatrix}1&|w|_{a_1}&|w|_{a_1a_2}&\cdots&|w|_{a_1a_2\cdots a_{n-2}}&|w|_{a_1a_2\cdots a_{n-1}}&|w|_{a_1a_2\cdots a_n}\\0&1&|w|_{a_2}&\cdots&|w|_{a_2\cdots a_{n-2}}&|w|_{a_2\cdots a_{n-1}}&|w|_{a_2\cdots a_n}\\\vdots&\vdots&\vdots&\cdots&\vdots&\vdots&\vdots\\\vdots&\vdots&\vdots&\cdots&\vdots&\vdots&\vdots\\0&0&0&\cdots&1&|w|_{a_{n-1}}&|w|_{a_{n-1}a_n}\\0&0&0&\cdots&0&1&|w|_{a_n}\\0&0&0&\cdots&0&0&1\end{pmatrix}_{(n+1)\times(n+1)}$$

Where $|w|_{a_1\cdots a_i}$ is the number of occurrences of $a_1 \cdots a_i$ in the word $w \in \Sigma^*$.Here $i \in [1, n]$.

Amiable or M-ambiguous words: Two words $\alpha, \beta \in \Sigma^* \ (\alpha \ne \beta)$ over the same alphabet Σ may have the same Parikh matrix. Then the words are called amiable or M-ambiguous.

3. PARIKH MATRIX OVER TERTIARY SEQUENCE

3.1 Parikh Matrix of a Word over Tertiary Alphabet

Let $\Sigma=\{a < b < c < d\}$ be an ordered tertiary alphabet. The Parikh matrix of

$$a=\begin{pmatrix}1&1&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{pmatrix}, b=\begin{pmatrix}1&0&0&0&0\\0&1&1&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{pmatrix}, c=\begin{pmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&1&0\\0&0&0&1&0\\0&0&0&0&1\end{pmatrix},$$

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

Any word over the tertiary alphabet has a unique Parikh Matrix. This matrix can be obtained by simple matrix product.

For example, the word $abcddabdc$ has the Parikh Matrix

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

which can be obtained from simple matrix product or by using the theory of Parikh Matrix which is like

$$\begin{pmatrix}1&|w|_a&|w|_{ab}&|w|_{abc}&|w|_{abcd}\\0&1&|w|_b&|w|_{bc}&|w|_{bcd}\\0&0&1&|w|_c&|w|_{cd}\\0&0&0&1&|w|_d\\0&0&0&0&1\end{pmatrix}$$

where $|w|_a$ denotes the number of scattered sub- word of a and $|w|_{ab}$ denotes the number of scattered sub- word of ab and so on.

To find the Parikh matrices for various words the above processes are used from the last decade. For smaller words these processes are well to do. But for larger words the above processes are time taking and clumsy. To overcome this problem the following algorithm is introduced. This algorithm gives instantly the Parikh matrix of a tertiary sequence however large the word may be. Results are verified.

3.2 Algorithm

The following pseudo code gives instantly the Parikh matrix of a tertiary sequence.

```
01  Initialize a word = ' w '
02  Set len = length of w
03  For i = 0 to len do
```

```
    Calculate total number of a, ab, abc, abcd in w .
    Calculate total number of b, bc, bcd in w .
    Calculate total number of c, cd in w .
    Calculate total number of d in w .
End // create a matrix (a_ij) of order M (= 5)
 For i = 0 to M do
    For j = 0 to M do
        If (i = j)
            a_ij = 1
        else If (i > j)
            a_ij = 0
        else
            If (i = 0 & j = 1)
              a_ij = total number of 'a'
            If (i = 0 & j = 2)
              a_ij = total number of 'ab'
            If (i = 0 & j = 3)
              a_ij = total number of 'abc'
            If (i = 0 & j = 4)
        a_ij = total number of 'abcd'
            If (i = 1 & j = 2)
              a_ij = total number of 'b'
            If (i = 1 & j = 3)
        a_ij = total number of 'bc'
            If (i = 1 & j = 4)
              a_ij = total number of 'bcd'
            If (i = 2 & j = 3)
              a_ij = total number of 'c'
            If (i = 2 & j = 4)
        a_ij = total number of 'cd'
            If (i = 3 & j = 4)
              a_ij = total number of 'd'
    End
```

3.3 Application of the Above Algorithm

Example 1 The tertiary word

$\xi_1 = abcd\underbrace{a\cdots a}_{100}\underbrace{b\cdots b}_{100}\underbrace{c\cdots c}_{100}\underbrace{d\cdots d}_{100}$ has the Parikh matrix

$$\Psi_{M_4}(\xi_1)=\begin{pmatrix} 1 & 101 & 10101 & 1010101 & 101010101 \\ 0 & 1 & 101 & 10101 & 1010101 \\ 0 & 0 & 1 & 101 & 10101 \\ 0 & 0 & 0 & 1 & 101 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Example 2 The tertiary word

$\xi_2 = abcd\underbrace{a\cdots a}_{10}\underbrace{b\cdots b}_{10}\underbrace{c\cdots c}_{10}\underbrace{d\cdots d}_{10}abcd$ has the Parikh matrix

$$\Psi_{M_4}(\xi_2)=\begin{pmatrix} 1 & 12 & 123 & 1234 & 12345 \\ 0 & 1 & 12 & 123 & 1234 \\ 0 & 0 & 1 & 12 & 123 \\ 0 & 0 & 0 & 1 & 12 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Example 3 The tertiary word

$\xi_3 = \underbrace{a\cdots a}_{11}\underbrace{b\cdots b}_{11}\underbrace{c\cdots c}_{11}\underbrace{d\cdots d}_{11}$ has the Parikh matrix

$$\Psi_{M_4}(\xi_3)=\begin{pmatrix} 1 & 11 & 121 & 1331 & 14641 \\ 0 & 1 & 11 & 121 & 1331 \\ 0 & 0 & 1 & 11 & 121 \\ 0 & 0 & 0 & 1 & 11 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Example 4 The tertiary word

$\xi_4 = abd\underbrace{c\cdots c}_{25}\underbrace{a\cdots a}_{10}\underbrace{d\cdots d}_{15}\underbrace{b\cdots b}_{10}abcd$ has the Parikh matrix

$$\Psi_{M_4}(\xi_4)=\begin{pmatrix} 1 & 12 & 123 & 148 & 523 \\ 0 & 1 & 12 & 37 & 412 \\ 0 & 0 & 1 & 26 & 401 \\ 0 & 0 & 0 & 1 & 17 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

4. A SET OF EQUATIONS FOR FINDING WORDS OVER TERTIARY ALPHABET FROM THE RESPECTIVE PARIKH MATRIX

A set of equations is proposed for finding the tertiary sequences corresponding to a given 5x5 Parikh matrix. Let $\Sigma=\{a<b<c<d\}$ be a tertiary ordered alphabet and

$$\Psi_{M_4}(\varsigma)=\begin{pmatrix} 1 & A & E & H & J \\ 0 & 1 & B & F & I \\ 0 & 0 & 1 & C & G \\ 0 & 0 & 0 & 1 & D \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{-----------(I)}$$

be a Parikh matrix i.e. $|w|_a = A$, $|w|_b = B$, $|w|_c = C$ and $|w|_d = D$ and so on. Then $\varsigma \in \Sigma^*$ is a tertiary sequence corresponds to the above matrix if ς can be represented in the following form:

$$\varsigma = a^{x_1} b^{y_1} c^{z_1} d^{t_1} a^{x_2} b^{y_2} c^{z_2} d^{t_2} \cdots$$
$$\cdots a^{x_{A+B+C+D}} b^{y_{A+B+C+D}} c^{z_{A+B+C+D}} d^{t_{A+B+C+D}} .$$

The Parikh Matrix corresponds to this word if and only if x_i = either 0 or 1, y_j = either 0 or 1, z_k = either 0 or 1 and t_l = either 0 or 1, is a solution of the following system of equations:

$$\sum_{i=1}^{A+B+C+D} x_i = A \quad (1)$$

$$\sum_{j=1}^{A+B+C+D} y_j = B \quad (2)$$

$$\sum_{k=1}^{A+B+C+D} z_k = C \quad (3)$$

$$\sum_{l=1}^{A+B+C+D} t_l = D \quad (4)$$

$$\sum_{i=1}^{A+B+C+D} x_i \sum_{j=i}^{A+B+C+D} y_j = E \quad (5)$$

$$\sum_{j=1}^{A+B+C+D} y_j \sum_{k=j}^{A+B+C+D} z_k = F \quad (6)$$

$$\sum_{k=1}^{A+B+C+D} z_k \sum_{l=k}^{A+B+C+D} t_l = G \quad (7)$$

$$\sum_{i=1}^{A+B+C+D} x_i \sum_{j=i}^{A+B+C+D} y_j \sum_{k=j}^{A+B+C+D} z_k = H \quad (8)$$

$$\sum_{j=1}^{A+B+C+D} y_j \sum_{k=j}^{A+B+C+D} z_k \sum_{l=k}^{A+B+C+D} t_l = I \quad (9)$$

$$\sum_{i=1}^{A+B+C+D} x_i \sum_{j=i}^{A+B+C+D} y_j \sum_{k=j}^{A+B+C+D} z_k \sum_{l=k}^{A+B+C+D} t_l = J \quad (10)$$

Example 1:

For clear understanding the example of the following Parikh matrix is taken. Let

$$\Psi_{M_4}(\varsigma_1) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

be a Parikh matrix. Here $|w|_a = 1, |w|_b = 1, |w|_c = 0, |w|_d = 0$ and so on. Then $\varsigma_1 \in \Sigma^*$ is a tertiary sequence corresponds to the above matrix. Here $A+B+C+D = 1+1+0+0 = 2$. So ς_1 can be represented in the following form: $\varsigma_1 = a^{x_1} b^{y_1} c^{z_1} d^{t_1} a^{x_2} b^{y_2} c^{z_2} d^{t_2}$

The Parikh Matrix corresponds to this word if and only if x_i = either 0 or 1, y_j = either 0 or 1, z_k = either 0 or 1 and t_l = either 0 or 1, is a solution of the following system of equations:

$$\sum_{i=1}^{2} x_i = 1 \quad (1)$$

$$\sum_{j=1}^{2} y_j = 1 \quad (2)$$

$$\sum_{k=1}^{2} z_k = 0 \quad (3)$$

$$\sum_{l=1}^{2} t_l = 0 \quad (4)$$

$$\sum_{i=1}^{2} x_i \sum_{j=i}^{2} y_j = 0 \quad (5)$$

$$\sum_{j=1}^{2} y_j \sum_{k=j}^{2} z_k = 0 \quad (6)$$

$$\sum_{k=1}^{2} z_k \sum_{l=k}^{2} t_l = 0 \quad (7)$$

$$\sum_{i=1}^{2} x_i \sum_{j=i}^{2} y_j \sum_{k=j}^{2} z_k = 0 \quad (8)$$

$$\sum_{j=1}^{2} y_j \sum_{k=j}^{2} z_k \sum_{l=k}^{2} t_l = 0 \quad (9)$$

$$\sum_{i=1}^{2} x_i \sum_{j=i}^{2} y_j \sum_{k=j}^{2} z_k \sum_{l=k}^{2} t_l = 0 \quad (10)$$

Now from (1), one gets, $x_1 + x_2 = 1$ from (2), one gets, $y_1 + y_2 = 1$ from (5), one gets.

$$x_1\left(y_1 + y_2\right) + x_2 y_2 = 0 \Rightarrow x_1\left(1\right) + x_2 y_2 = 0 \text{ [from (2)]}$$
$$\Rightarrow y_2 = 0 \quad \text{[using (1)]}$$

$\therefore y_1 = 1$ again from (5) one has

$$x_1\left(y_1 + y_2\right) + x_2 y_2 = 0 \Rightarrow x_1\left(1+0\right) + x_2 \cdot 0 = 0$$
$$\Rightarrow x_1 = 0$$

So the word $\varsigma_1 = a^{x_1} b^{y_1} c^{z_1} d^{t_1} a^{x_2} b^{y_2} c^{z_2} d^{t_2}$ is

$\varsigma_1 = a^0 b^1 c^0 d^0 a^1 b^0 c^0 d^0 = ba$

Example 2: For the Parikh matrix

$$\Psi_{M_4}\left(\varsigma_2\right) = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Here $|w|_a = 1,\ |w|_b = 1,\ |w|_c = 1,\ |w|_d = 1$ and so on. Then $\varsigma_2 \in \Sigma^*$ is a tertiary sequence corresponds to the above matrix. Here $A + B + C + D = 1+1+1+1 = 4$. So ς_2 can be represented in the following form:

$$\varsigma_1 = a^{x_1} b^{y_1} c^{z_1} d^{t_1} a^{x_2} b^{y_2} c^{z_2} d^{t_2} a^{x_3} b^{y_3} c^{z_3} d^{t_3} a^{x_4} b^{y_4} c^{z_4} d^{t_4}$$

The Parikh Matrix corresponds to this word if and only if x_i = either 0 or 1, y_j = either 0 or 1, z_k = either 0 or 1 and t_l = either 0 or 1, is a solution of the system of equations (I) . One gets the solution of the equations as $x_4 = 1, y_3 = 1, z_2 = 1 \text{and } t = 1$

And remaining all others is 0. Then the word is

$$\varsigma_2 = a^{x_1} b^{y_1} c^{z_1} d^{t_1} a^{x_2} b^{y_2} c^{z_2} d^{t_2} a^{x_3} b^{y_3} c^{z_3} d^{t_3} a^{x_4} b^{y_4} c^{z_4} d^{t_4}$$
$$\Rightarrow \varsigma_2 = a^0 b^0 c^0 d^1 a^0 b^0 c^1 d^0 a^0 b^1 c^0 d^0 a^1 b^0 c^0 d^0$$
$$= dcba$$

This is how one can use the proposed set of equations to find the corresponding word from a 5x5 Parikh matrix.

5. DEFINITION OF STEPPING DISTANCE ON CLASSES OF M-AMBIGUOUS WORDS

For convenience it is assumed that the symbols a, b, c, d are lying on a straight line. So to describe the distant between two symbols on the straight line one has to either step forward or step backward. For this reason this distance is named as stepping distance. Using this notion of stepping distance one can compare M-ambiguous words.

Let $\alpha = a_1 a_2 a_3 \ldots a_n;\ a_i \in \{a, b, c, d\}$ and

$$\beta = b_1 b_2 b_3 \ldots b_n;\ b_i \in \{a, b, c, d\}$$

be two M- ambiguous words. The stepping distance defined on the class of M-ambiguous words over

$\Sigma = \{a, b, c, d\}$ is defined as $d_S\left(\alpha, \beta\right) = \sum_{i=1}^{n}\left(a_i +_S b_i\right)$. Where $+_S : \Sigma \times \Sigma \rightarrow \{0, 1, 2, 3\}$ is defined by

$$a +_S a = 0,$$
$$b +_S b = 0,$$
$$c +_S c = 0,$$
$$d +_S d = 0,$$
$$a +_S b = 1 = b +_S a,$$
$$b +_S c = 1 = c +_S b,$$
$$c +_S d = 1 = d +_S c,$$
$$a +_S c = 2 = c +_S a,$$
$$b +_S d = 2 = d +_S b,$$
$$a +_S d = 3 = d +_S a.$$

For example, the stepping distance $d_S\left(\alpha, \beta\right)$ between the following two M- ambiguous words $\alpha = abcddcbadcba$, $\beta = abdccbddacba$ is 12.

6. CONCLUSION

To find the Parikh matrices of tertiary words by general existing processes are time consuming and clumsy. This paper presents an algorithm to build the Parikh matrix of a word over tertiary alphabet instantly. This algorithm can help in further investigations of words over tertiary ordered alphabet. A generalized approach to this algorithm for n^{th} ordered alphabet still waits investigation. A set of equations to determine the amiable words corresponding to a particular Parikh matrix over tertiary alphabet is introduced. These equations give a new arena for investigation in this field .It awaits investigation to extend the equations for more than four symbols. A type of distant between M- ambiguous words namely Stepping distance is introduced. Using Stepping distance one can compare M- ambiguous words.

7. REFERENCES

[1] R.J.Parikh: On the context-free languages. *Journal of the Association for Computing Machinery*, 13 (1966), 570-581.

[2] A. Mateescu, A. Salomaa, K. Salomaa, S.Yu: A sharpening of the Parikh Mapping. *Theoret. Informetics Appl.*, 35 (2001), 551-564.

[3] A. Mateescu, A. Salomaa, K. Salomaa, S.Yu: On an extension of the Parikh mapping, *T.U.C.S Technical Report No 364*.

[4] K. G. Subramanian, A. M. Huey, A. K. Nagar: On Parikh matrices, *Int. J. Found. Comput. Sci.* 20(2) (2009)211-219.

[5] C. Ding, A. Salomaa: On some problems of Mateescu concerning sub word occurrences, *Fundamenta Informaticae* 72(2006) 1-15.

[6] V.N. Serbˇanutˇa, Injectivity of the Parikh matrix mappings revisited, *Fundamenta Informaticae* XX (2006) 1–19, IOS Press.

[7] A. Atanasiu, C. M.Vide, A. Mateescu, On the injectivity of the Parikh matrix mapping, *Fundam. Informa.*46 (2001) 1-11.

[8] A. Salomaa et al. Subword conditions and subword histories. Information and Computation 204 (2006) 1741–1755.

[9] A. Mateescu, A. Salomaa: Matrix indicators for subword occurrences and ambiguity. *Int. J. Found. Comput. Sci*, 15 (2004), 277–292.

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

[11] T.-F. S¸ erbˇanut¸ˇa: Extending Parikh matrices. *Theoretical Computer Science*, 310 (2004), 233–246.

[12] S. Fosse, G. Richmomme, Some characterizations of Parikh matrix equivalent binary words, *Information Processing Letters*. 92 (2) (2004) 77–82.

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

[14] A. Salomaa: Connections between subwords and certain matrix mappings. *Theoretical Computer Science*, 340 (2005), 188–203.

[15] A. Salomaa, Independence of certain quantities indicating subword occurrences, *Theoretical Computer Science.* 362 (1) (2006) 222–231.

[16] A. Salomaa et al. Subword conditions and subword histories. *Information and Computation* 204 (2006) 1741–1755.

[17] Adrian Atanasiu, Radu Atanasiu, Ion Petre: Parikh matrices and amiable words. *Theoretical Computer Science* 390 (2008) 102–109.

[18] A. Atanasiu, Binary amiable words, *Int. J. Found. Comput. Sci* 18 (2) (2007) 387-400.