ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 11 Nov. 2016, Page No. 18795-18799

Parikh matrix on the Context-Free Grammar for Natural Languages

Amrita Bhattacharjee[1], Bipul Syam Purkayastha[2]
[1]Department of Computer Science,
Assam University Silchar 788011, India,
amritabhattacharjee10@gmail.com
[2]Department of Computer Science,
Assam University Silchar 788011, India,
bipulsyam@gmail.com

Abstract: ***In this paper Parikh Matrices over context-free languages are investigated. Context-free grammars for Natural languages are a developing area of investigation. Parikh matrix is a significant tool of Formal languages. Context-free language is a kind of formal language. Parikh matrix can be used in context-free language. A context-free grammar for Bengali language is also a developing area of investigation. As a case study in this paper Bengali letters, Bengali words and Bengali sentences are studied by using Parikh matrix.***

Keywords: Parikh matrix, Subword, M- ambiguous words, Formal grammar, Context-free grammar.

1. Introduction

Natural languages are those languages which are used by human beings either vocally or in written form in their day to day life for communication. Natural language and formal language are different to each other with respect to their configuration and utility. Much work has been done to establish the interrelation between natural language and formal language. A formal language is often defined by means of a formal grammar such as a regular grammar or context-free grammar. In this present work Parikh matrix, a tool of formal language is used in a Natural language. Parikh matrix is defined on formal language. The Parikh mapping or Parikh vector is an old and important tool in the theory of formal languages introduced by R.J.Parikh [1]. This notion is an important tool in the theory of formal languages. With the help of this tool properties of words can be expressed numerically. For example, for the word $w = abbaccac$ the Parikh vector is $(3,2,3)$. In 2001 Mateescu et al. [2] introduced the notion of Parikh matrix. With every word over an ordered alphabet, a Parikh Matrix can be associated and it is a triangular matrix. In recent decades many techniques have been developed to solve complex problems of words using Parikh Matrix. The notion of Parikh matrix is an extension of Parikh Mapping. We cite a few examples [3, 4, 5 …17] which have used Parikh matrix for solving the problems of word.

An ordered alphabet is a set of symbols $\Sigma = \{a_1, a_2, \cdots, a_n\}$ where the symbols are arranged maintaining a relation of order ("$<$") on it. For example if $\{a_1 < a_2 < \cdots < a_n\}$, then we use notation $\Sigma = \{a_1, a_2, \cdots, a_n\}$. With every word over an ordered alphabet, a Parikh matrix can be associated and it is a triangular matrix. All the entries of the main diagonal of this matrix is 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 w. As for example the tertiary word

$$\xi_1 = abd\underbrace{c\cdots c}_{25}\underbrace{a\cdots a}_{10}\underbrace{d\cdots d}_{15}\underbrace{b\cdots b}_{10}abcd$$

has the Parikh matrix

$$\Psi_{M_4}(\xi_1) = \begin{pmatrix} 1 & 12 & 123 & 148 & 523 \\ 0 & 1 & 12 & 37 & 412 \\ 0 & 0 & 1 & 26 & 401 \\ 0 & 0 & 0 & 1 & 17 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Natural Language Processing (NLP) is a theoretically inspired variety of computational techniques. It is used for analysing and representing naturally occurring texts at one or more levels of linguistic analysis. Bengali is a natural language. Bengali language is an emerging area of investigation of NLP. Many research works are going on in the field of Bengali language. In this paper study is done on the same field. The fact used in this paper is that a context-free grammar for Bengali language can be generated.

The paper is organized as follows. The following section 2 reviews the related works on Parikh Matrix and Bengali Language Processing. Section 3 goes toward reviewing the basic preliminaries of Parikh Matrix and computational linguistics. Section 4 gives representation of Bengali letters by Parikh matrices; In Section 5, representation of Bengali words by Parikh matrices is introduced; in section 6, representation of Bengali sentences by Parikh matrices are presented. We conclude the paper in Section 7 by summarizing the observations.

2. Preliminaries

Throughout this paper Z will denote the set of natural numbers including zero. First we recall some definitions.

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 we 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 $w = abaabcac$ is a word over the alphabet $\Sigma = \{a, b, c\}$ 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 $w = abaabcac$, the number of occurrences of the word $baca$ as a sub-word of w is $|w|_{baca} = 2$.

Parikh vector: The Parikh vector is a mapping $\Psi : \Sigma^* \to Z \times Z \times \cdots \times Z$ where $\Sigma = \{a_1, a_2, a_3, \cdots, a_n\}$ and Z is the set of natural numbers including 0, such that for a word w in Σ^*, $\Psi(w) = (|w|_{a_1}, |w|_{a_2}, |w|_{a_3}, \cdots, |w|_{a_n})$ with $|w|_{a_i}$ denoting the number of occurrences of the letter $a_i \in w$. For example, for the word $w = abaabcac$ the Parikh vector is $(4, 2, 2)$.

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

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

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 $\Psi_{M_n}(a_q) = (m_{i,j})_{1 \le i, j \le n+1}$ then $m_{i,i} = 1$, $m_{q,q+1} = 1$ and all other elements are zero.

M-ambiguous or Amiable 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.

The words *baaabaa* and *abab aaa* has the same Parikh Matrix $\begin{pmatrix} 1 & 5 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{pmatrix}$. So these two words are amiable.

M-unambiguous words: A word w is said to be M-unambiguous if there is no word w' for which $\Psi_{M_n}(w) = \Psi_{M_n}(w')$.

Formal grammars: A formal grammar of this type consists of:

- a finite set of terminal symbols,
- a finite set of nonterminal symbols,
- a finite set of production rules (left-hand side $\to$ right-hand side) where each side consists of a sequence of these symbols,
- a start symbol.

Context-free grammar: A context-free grammar *G* is defined by the 4-tuple: $G = \{V, \Sigma, R, S\}$ where

1. V is a finite set; each element $v \in V$ is called *a* non-terminal character or a variable. Each variable represents a different type of phrase or clause in the sentence. Variables are also sometimes called syntactic categories. Each variable defines a sub-language of the language defined by G.
2. Σ is a finite set of terminals, disjoint from V, which make up the actual content of the sentence. The set of terminals is the alphabet of the language defined by the grammar G.
3. R is a finite relation from V to $(V \cup \Sigma)^*$. The members of R are called the rules or productions of the grammar.
4. S is the start variable, used to represent the whole sentence (or program). It must be an element of V.

The asterisk represents the Kleene star operation.

3. Related works

Since the introduction of the notion of Parikh vector in 1966 [1] continuous research works are going on in this field. In this introductory paper [1] certain properties of context-free or type 2 grammars are investigated. In particular, questions regarding structure, possible ambiguity and relationship to finite automata are considered. Some important results are also presented. A sharpening of Parikh mapping namely Parikh matrix is introduced in [2] and this matrix representation gives more information than Parikh vector does. With the extension of Parikh matrix an interesting interconnection between mirror images of words and inverses of matrices are investigated in [3]. Researchers [4] have presented ratio property of two words. This property is a sufficient condition for the words uv & vu to have the same Parikh matrix. In this paper [5] universal languages for Parikh matrices is introduced and studied. In [6] M-unambiguity is studied both in terms of words and matrices and several criteria for M-unambiguity are provided in both cases. In this paper [7] palindromic amicable words are studied in the context of binary words. Researchers [8] have introduced subword condition. Various characterization and decidability results for languages subword conditions are discussed. In this paper [9] Parikh Matrices over tertiary alphabet are investigated. Algorithm is developed to display Parikh Matrices of words over tertiary alphabet. A set of equations for finding tertiary words from the respective Parikh matrix is introduced. In this paper [10] the notion of a subword history closely related to Parikh matrices is introduced and obtained a sequence of general results. A general inequality of Cauchy type for subword occurrences is established. In [11] a natural extension of Parikh matrix and a set of properties for this kind of matrices are investigated. The combination of Parikh matrix and this extension give a more powerful tool for the study of algebraic properties of words. Different characterizations of pairs of words having the same Parikh matrix are investigated in this paper [12]. In this paper [13] certain inequalities, as well as information about subword occurrences sufficient to determine the whole word uniquely are studied. Some algebraic considerations, facts about forbidden subwords, as well as some open problems are also included. Researchers [14] have investigated the numerical quantity $|w|_u$, the number of occurrences of a word u as a (scattered) subword of a word w. Parikh matrices recently introduced have these quantities as their entries. According to the main result in this paper, no entry in a Parikh matrix, no matter how high the dimension can be computed in terms of the other entries. In [15] Amiable words are investigated. It is shown that all the words having the same Parikh matrix can be obtained one from another by applying only two types of transformations. It is also shown that mirrors of two amiable words are also amiable. The paper [16] investigates some properties of the set of binary words having the same Parikh matrix. These words are named as amiable words. Investigations are done on the equivalence class. A characterisation theorem concerning a graph associated to an equivalence class of amiable words, and some basic properties of a rank distance is discussed. In this paper [17] ratio property of words are investigated. A relationship of ratio property with M- ambiguity is established. M-ambiguous words are formed by concatenating words satisfying ratio property. In the papers [24, 25] binary and ternary words corresponding to Parikh matrices are discussed. Equations, algorithms are used for finding M-ambiguous words. Graphical representations of formal words are introduced in [24, 25]. In the papers [19, 20, 21, 22 23] context-free grammar for natural language is studied.

4. Representation of Natural Language Letters by Parikh Matrices, Bengali Letters, A Case Study

A Natural Language is any one of the languages naturally used by human beings, whereas an artificial or man-made language is nothing but a computer programming language. 'Natural Language Processing' (NLP) is the computerized approach to process natural languages. Bengali is a Natural language. And it is possible to make a context-free grammar for Bengali. The matter of developing context-free grammar for Bengali is an emerging area of investigation. In the following research papers [19, 20, 21, 22 23] this problem is investigated. Depending on the above mentioned research works some results regarding context-free language are applied to Bengali language. Parikh matrix is one of the results concerning context-free language. In this paper Bengali alphabets are represented by Parikh matrices.

Bengali Alphabet is given below:

Swarbarna: A B C D E F G H I J K

Byanjanbarna: L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q s t u v w x y

It is an ordered alphabet. There are 50 letters in Bengali alphabet. In this paper an effort is given to use Parikh Matrices in Bengali letters. For this we enumerate each letter one by one maintaining the above sequence of the letters in the alphabet. That is we are taking A as the 1st letter, B as the 2nd letter, and so on. By this way we are taking L as the 12th letter, M as the 13th letter, and last one **y** as the 50st letter. By Parikh Matrices, A is defined as follows:

$$
\begin{array}{cc}
 & \begin{matrix} & 2^{nd} & & & & & & & \end{matrix} \\
\begin{matrix} 1^{st} \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} &
\begin{pmatrix}
1 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 1 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & \cdots & \cdots & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 1
\end{pmatrix}_{51x51}
\end{array}
$$

Similarly B and all other swarbarna s are defined.

Again L is defined as follows:

$$
\begin{array}{cc}
 & \begin{matrix} & & & & & 13^{th} & & & \end{matrix} \\
\begin{matrix} \\ \\ \\ \\ \\ 12^{th} \\ \\ \\ \\ \end{matrix} &
\begin{pmatrix}
1 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 1 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 0 & \cdots & 1 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 1
\end{pmatrix}_{51x51}
\end{array}
$$

Similarly M and all other byanjanbarna s are defined.

5. Representation of Bengali words by Parikh Matrices

In this section the way of representing Bengali words by Parikh matrices is shown. If we want to write Bj, by using Parikh Matrices, we have to find the matrix product of the two matrices corresponding to B and j.

$$
\begin{array}{cc}
 & \begin{matrix} & & 3^{rd} & & & & & & \end{matrix} \\
\begin{matrix} \\ 2^{nd} \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} &
\begin{pmatrix}
1 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 1 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & \cdots & \cdots & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 1
\end{pmatrix}_{51x51}
\end{array}
$$

$\cdot$

$$
\begin{array}{cc}
 & \begin{matrix} & & & & & & 37^{th} & & \end{matrix} \\
\begin{matrix} \\ \\ \\ \\ 36^{th} \\ \\ \\ \\ \\ \end{matrix} &
\begin{pmatrix}
1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & \cdots & \cdots & \cdots & 0 & 1 & 1 & \cdots & 0 \\
0 & \cdots & \cdots & \cdots & \cdots & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 1
\end{pmatrix}_{51x51}
\end{array}
$$

=

$$\begin{array}{cc}
 & \begin{array}{ccccccccc} & & 3^{rd} & & & & 37^{th} & & \end{array} \\
\begin{array}{c} \\ 2^{nd} \\ \\ \\ \\ 36^{th} \\ \\ \\ \\ \end{array} & \left(\begin{array}{ccccccccc}
1 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 1 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & \cdots & \cdots & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 1
\end{array}\right)_{51x51}
\end{array}$$

Bengali language has 50 letters. Making “Sandhi bicched” that is resolution of Bengali words into respective letters we can use Parikh Matrices to Bengali words. As for example we can take the following word R¢h . This word can be resolute as R¢h = R + h + C. By matrix product of the Parikh matrices of the letters R, h and C we can get the Parikh matrix of the word R¢h as follows:

$$\begin{array}{cc}
 & \begin{array}{ccccccccc} & & & 4^{th} & & 19^{th} & & 35^{th} & \end{array} \\
\begin{array}{c} \\ \\ 3^{rd} \\ \\ 18^{th} \\ \\ 34^{th} \\ \\ \\ \end{array} & \left(\begin{array}{ccccccccc}
1 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 1 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 1 & 1 & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
0 & 0 & 0 & \cdots & \cdots & 1 & \cdots & \cdots & 0 \\
0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 1 & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 1
\end{array}\right)_{51x51}
\end{array}$$

In this section we are giving some more examples for clear understanding. For convenience the 51x51 matrices are represented as follows:

-Q±d¤l£ = Q + K + d + E + l + D

$$a_{i,i} = 1,\ a_{4,5} = 1, a_{5,6} = 1, a_{11,12} = 1, a_{17,18} = 1, a_{30,31} = 1, a_{38,39} = 1;$$

$a_{i,j} = 0$ for the rest of the entries of the 51x51 matrix.

¢h-hL¡eec = h + C + h + H + L + B + e + e + c

$$a_{i,i} = 1,\ a_{2,3} = 2, a_{3,4} = 1, a_{8,9} = 1, a_{12,13} = 1, a_{31,32} = 2, a_{29,30} = 1, a_{34,35} = 2;$$

$a_{i,j} = 0$ for the rest of the entries of the 51x51 matrix.

6. Representation of Bengali sentences by Parikh Matrices

For further investigation on the use of Parikh matrix on Bengali language the following simple sentence is taken. For further investigation on the use of Parikh matrix on Bengali language the following simple sentence kc¤ i¡m -R-m

is taken. . This sentence is written by permutation of the separate words as follows:

kc¤ i¡m -R-m

i¡m -R-m kc¤

-R-m i¡m kc¤

i¡m kc¤ -R-m

kc¤ -R-m i¡m

-R-m kc¤ i¡m

All the above sentences are having the same meaning. Difference is that some of them are frequently used in the Bengali language and some are not. We can take a line as a whole. We use Parikh matrix in the above Bengali lines as follows:

For the word kc¤ = k + c + E the Parikh matrix is $a_{i;i} = 1$; $a_{5;6} = 1$; $a_{29;30} = 1$; $a_{37;38} = 1$ and 0 for the rest of the entries of the 51 x 51 matrix.

For the word i¡m = i + B + m the Parikh matrix is $a_{i,\ i} = 1$; $a_{2,3} = 1$; $a_{35,36} = 1$; $a_{39,40} = 1$; and 0 for the rest of the entries of the 51 x 51 matrix.

For the word -R-m = R + H + m + H the Parikh matrix is $a_{i,i} = 1$; $a_{2,3} = 3$; $a_{5,6} = 1$; $a_{8,9} = 1$; $a_{29,30} = 1$; $a_{34,35} = 1$; $a_{34,36} = 1$; $a_{35,36} = 1$; $a_{37,38} = 1$; $a_{39,40} = 1$; and 0 for the rest of the entries of the 51 x 51 matrix..

Then the matrix product of all the Parikh matrices of the words is taken. This gives the Parikh matrix of the line kc¤ i¡m -R-m as $a_{i;i} = 1$; $a_{2;3} = 2$; $a_{5;6} = 1$; $a_{8;9} = 1$; $a_{29;30} = 1$; $a_{34;35} = 1$; $a_{35;36} = 1$; $a_{37;38} = 1$; $a_{39;40} = 2$ and 0 for the rest of the entries of the 51 x 51 matrix.

The Parikh matrix of the line i¡m -R-m kc¤ as $a_{i;i} = 1$; $a_{2;3} = 2$; $a_{5,6} = 1$; $a_{8,9} = 1$; $a_{29,30} = 1$; $a_{34,35} = 1$; $a_{35,36} = 1$; $a_{37,38} = 1$; $a_{39,40} = 2$ for the rest of the entries of the 51 x 51 matrix.

The Parikh matrix of the line -R-m i¡m kc¤ as $a_{i,i} = 1$; $a_{2,3} = 2$; $a_{5,6} = 1$; $a_{8,9} = 1$; $a_{29,30} = 1$; $a_{34,35} = 1$; $a_{34,36} = 1$; $a_{35,36} = 1$; $a_{37,38} = 1$; $a_{39;40} = 2$ and 0 for the rest of the entries of the 51 x 51 matrix.

The Parikh matrix of the line i¡m kc¤ -R-m as $a_{i;i} = 1$; $a_{2;3} = 2$; $a_{5;6} = 1$; $a_{8;9} = 1$; $a_{29;30} = 1$; $a_{34;35} = 1$; $a_{35;36} = 1$; $a_{37;38} = 1$; $a_{39;40} = 2$ and 0 for the rest of the entries of the 51 x 51 matrix.

The Parikh matrix of the line kc¤ -R-m i¡m as $a_{i;i} = 1$; $a_{2;3} = 2$; $a_{5;6} = 1$; $a_{8;9} = 1$; $a_{29;30} = 1$; $a_{34;35} = 1$; $a_{34;36} = 1$; $a_{35;36} = 1$; $a_{37;38} = 1$; $a_{39;40} = 2$ and 0 for the rest of the entries of the 51 x 51 matrix.

The Parikh matrix of the line -R-m kc¤ i¡m as $a_{i;i} = 1$; $a_{2;3} = 2$; $a_{5;6} = 1$; $a_{8;9} = 1$; $a_{29;30} = 1$; $a_{34;35} = 1$; $a_{34;36} = 1$; $a_{35;36} = 1$; $a_{37;38} = 1$; $a_{39;40} = 2$ and 0 for the rest of the entries of the 51 x 51 matrix.

7. Conclusion

Depending on the previous works done on the context-free grammar for Bengali language by various researchers Parikh matrix is applied on Bengali language. The Parikh matrix of every Bengali letter is a 51x51 matrix. All the entries of the main diagonal of this matrix is 1 and every entry below the main diagonal has the value 0 but the entries above the main diagonal provide information on the concerning Bengali letter . Every word is a matrix product of these matrices. The entries above the main diagonal provide information on the concerning Bengali word and Bengali sentence. It is an effort to use the tool Parikh matrix to Bengali language. With the advance of development of context-free grammar for Bengali language this effort will result more fruitful. Many tools of context-free grammar can be used to Bengali language. Various results of Parikh matrix can also be applied to Bengali language.

References

[1] R.J.Parikh, (1966) "On the context-free languages" *Journal of the Association for Computing Machinery*, Vol No.13 pp 570-581.

[2] A. Mateescu, A. Salomaa, K. Salomaa, S.Yu, (2001) "A sharpening of the Parikh Mapping" *Theoret. Informetics Appl.,* Vol No. 35 pp 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, (2009) "On Parikh matrices" *Int. J. Found. Comput. Sci.* Vol.20 No.2 pp211-219.

[5] C. Ding, A. Salomaa, (2006) "On some problems of Mateescu concerning sub word occurences" *Fundamenta Informaticae* Vol No. 72 pp1-15.

[6] V.N. Serb˘anut˘a, (2006) "Injectivity of the Parikh matrix mappings revisited" *Fundamenta Informaticae* Vol No. XX pp1–19, IOS Press.

[7] A. Atanasiu, C. M.Vide, A. Mateescu, (2001) "On the injectivity of the Parikh matrix mapping" *Fundam. Informa*. Vol No.46 pp 1-11.

[8] A. Salomaa et al, (2006) "Subword conditions and subword histories" *Information and Computation* Vol No.204 pp1741–1755.

[9] Amrita Bhattacharjee, Bipul Syam Purkayastha, (2014) "Parikh Matrices and Words over Tertiary Ordered Alphabet" *Int. J. of Computer Applications,* Vol 85 No.4 pp10–15.

[10] A. Mateescu, A. Salomaa, S. Yu, (2004) "Subword histories and Parikh matrices" *J. Comput. Syst. Sci.,* Vol No.68 pp 1–21.

[11] T.-F. S¸ erb˘anut¸˘a, (2004) "Extending Parikh matrices" *Theoretical Computer Science*, Vol No.310 pp 233–246.

[12] S. Fosse, G. Richmomme, (2004) "Some characterisations of Parikh matrix equivalent binary words" *Information Processing Letters*. Vol.92 No.2 pp77–82.

[13] A. Salomaa, (2005) "Connections between subwords and certain matrix mappings" *Theoretical Computer Science,* Vol No.340 pp 188–203.

[14] A. Salomaa, (2006) "Independence of certain quantities indicating subword occurences" *Theoretical Computer Science.* Vol.362 No.1 pp222–231.

[15] Adrian Atanasiu, Radu Atanasiu, Ion Petre, (2008) "Parikh matrices and amiable words" *Theoretical Computer Science Vol No.*390 pp 102–109.

[16] A. Atanasiu, (2007) "Binary amiable words" *Int. J. Found. Comput. Sci* Vol.18 No.2 pp387-400.

[17] Amrita Bhattacharjee, Bipul Syam Purkayastha, Application of ratio property in searching of M-ambiguous words and its generalization, 3rd International Conference on Soft Computing for Problem Solving ,AISC (SocProS 2013) 258, pp857-865. December 27-29, Greater Noida Extension Centre of IIT Roorkee, India.

[18] J. E. Hopcroft, Rajeev Motwani and J. D. Ullman, "Introduction to Automata Theory Languages and Computation," Pearson Education Publishing Company, Second Edition (book style)

[19] Apurbalal Senapati, Utpal Garain "Bangla Morphological Analyzer using Finite Automata" ISI @FIRE MET 2012

[20] K. M. Azharul Hasan, Al-Mahmud, Amit Mondal, Amit Saha "RECOGNIZING BANGLA GRAMMAR USING PREDICTIVE PARSER" IJCSIT Vol 3 December 2011.

[21] Muhammad Nasimul Haque and Mumit Khan "Parsing Bangla using LFG: An Introduction" (IJCSIT) Vol 3, No 6, Dec 2011

[22] M. A. Karim (Old Dominion University, USA), M. Kaykobad (Bangladesh University of Engineering & Technology, Bangladesh) and M. Murshed (Monash University, Australia) chapter 7 "Parsing Bangla Grammar Using Context-free Grammar".

[23] M.S. Hasan ,Amit Mondal, Amit Saha "A context-free grammar and its predicative parser for Bangla grammar recognition" IEEE Xplore , computer and information technology,(ICCIT), 2010, 13th International conference.

[24] Amrita Bhattacharjee, Bipul Syam Purkayastha, "Some Alternative Ways To Find M-Ambiguous Binary Words Corresponding To A Parikh Matrix", International Journal of Computational Science, AIRCC, 2014, 4(1), 53-64.

[25] Amrita Bhattacharjee, Bipul Syam Purkayastha, Parikh Matrices and words over ternary alphabet, 4th International Conference on Soft Computing for Problem Solving, AISC (SocProS 2014) **335**,135-145. Springer India, December 26-28, National Institute of Technology, Silchar, India.

[26] **Author Profile**

Amrita Bhattacharjee received M. Sc. Degree in Pure Mathematics from Gauhati University and at present a Ph.D. Research scholar of Assam University.