Chapter 6

Conclusion

Formal language is the area of interesting study having diverse scope, for which the advancement in this field of knowledge is also encouraging. Formal language has its impact on the subjects like Linguistic, Mathematics and Computer Science etc. and vice versa.

First chapter is mainly discussed on the introduction of the present study comprising of motivation, statement of the problem, objectives, data and methodology and some definitions associated with the study. Review of literature in connection with the present study is discussed in the second chapter. A finite state automaton is a very important notion for the study of regular languages. The third chapter of this study is based on finite state automata. Finite state automata and state transition table are discussed in the context of regular languages. An algorithm to construct deterministic finite state automata is introduced.

The fourth chapter of the present study is based on quite recent topic namely Parikh matrix. As mentioned before with every word over an ordered alphabet, a Parikh Matrix can be associated and it is a triangular matrix. The Parikh matrix of $w \in \Sigma^*$ is as follows:
Any word w over the n^{th} order alphabet $\Sigma = \{a_1, a_2, a_3, \cdots, a_n\}$ has a unique

Parikh matrix. This matrix is given by $\Psi_{M_n}(w)$.

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

where $|w|_{a_i}$ is the number of occurrences of a_i in the word. Here $i \in [1, n]$. 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 the word. To find the Parikh matrices for various words numerous processes are being used since the last decade. For smaller words these processes are helpful. But for larger words the above processes are time taking and clumsy. To overcome this problem present study presents one algorithm for finding the Parikh matrix corresponding to a word. This helps us to find Parikh matrix of a binary word instantly, however big or small. This algorithm is extended to ternary and tertiary alphabets.

M-ambiguity is the pain in the neck of the tool Parikh matrix. M-ambiguous words are the main research problem of Parikh matrices. In this present study an algorithm is shown to find the M- ambiguous words of a binary ordered word. This algorithm is also extended to ternary and tertiary alphabet.

A system to represent binary words in a two dimensional field is introduced. There are some relations among the representations of M-ambiguous words in the two dimensional field. The area covered by line depicted by the word and the x-axis and the line parallel to the y-axis through the point $(|w|_a, 0)$ is the same for all M-ambiguous words corresponding to a 3×3 Parikh matrix. The number of squares covered by the word line, y-axis and the line parallel to the x-axis through the point $(0, |w|_b)$ is the same as the number $|w|_{ab}$. It is seen that if one word corresponding to a Parikh matrix is known then those words which have the same area covered below the word line and with the same Parikh matrix are all M-ambiguous words. Then method of graphical representation of ternary sequences is introduced.

Sets of equations by which one can calculate the M-ambiguous words of a Parikh matrix are also introduced. One can find M-ambiguous words by using this set of equations. Three sets of equations are developed for binary, ternary and tertiary ordered alphabets. For binary words the numbers of equations in the set are three. For ternary words the numbers of equations in the set are six. For tertiary words the numbers of equations in the set are ten.

The condition for a 4×4 matrix to be a Parikh matrix is discussed in terms of theorem and results.

In this research work a type of distance function between M-ambiguous words namely Stepping distance is introduced. Using Stepping distance M-ambiguous words can be compared. To reduce the problem of ambiguity of words corresponding to a particular Parikh matrix ambiguity reduction factor is considered. Using this reduction factor the problem of ambiguity can be handled to some extent.

This study presents generalization of ratio property and weak ratio property. Some lemmas proved for ternary sequences are extended for tertiary words. Using those lemmas certain M-ambiguous words are formed from words satisfying ratio property by concatenating them in certain ways.

The root of the natural language grammar can be traced to the ancient days' study on natural language. As for example, over two thousand years ago the great Indian grammarian Panini wrote his grammar of Sanskrit. After the development of computer science and Turing machine Chomsky developed new types of grammars namely– Unrestricted grammars, Context-sensitive grammars, Context-free grammars, Regular grammars. The fifth chapter is based on the concept of connecting natural languages to formal language. 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 research papers [58, 67, 44, 65, 5] this area is studied. Parikh matrix is applied to Bengali language in the present study. The Parikh matrix of every Bengali letter is a 51×51 matrix. Parikh matrix will become a very useful tool for mathematisation of Bengali language if the context-free grammar for Bengali language is fully materialised. The same is true for other language also. Many tools of context-free grammar can be used to Bengali language. Various results of Parikh matrix can also be applied to natural languages for which context-free grammar is developed.

*****