Abstract

Formal language is a well-known term in Computer Science, Linguistics and Mathematics. It is a set of sequences of symbols that may be guarded by rules that are specific for a particular formal language. In Computer Science, formal languages are used as the basis for defining programming languages. In the present study some discussions are done on the words over formal languages.

Alphabet of a formal language is the set of symbols, letters, or tokens from which the strings of the language may be formed; it may be finite or infinite but generally it is taken as finite. An ordered alphabet is a set of symbols 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 notation of ordered alphabet is $\Sigma = \{a_1 < a_2 < a_3 < \cdots < a_n\}$, for two symbols notation $\{a < b\}$ is used. Sometimes $\Sigma = \{a_1, a_2, a_3, \cdots, a_n\}$ notation is also used for an ordered alphabet. The strings formed from this alphabet are called words, and the words that belong to a particular formal language are sometimes called well-formed formulae.

Finite state automaton has its own importance in formal language. Application of state graph which is an inevitable part of finite state automata with reference to formal languages is studied. Finite state automaton is divided into two parts-deterministic and nondeterministic.

In this study description of how automata are used to describe regular languages is given. Some illustrations about finite state automaton and state-transition table and their contribution to represent regular languages are given here. An algorithm to construct deterministic finite state automata is presented. This algorithm gives the finite state automaton of a word over regular languages instantly.

The Parikh mapping or Parikh vector is an old and important tool in the

theory of formal languages. For the word $aabca$ the Parikh vector is $(3, 1, 1)$. The notion of Parikh matrix is an extension of Parikh mapping. 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 the word. As for example the tertiary word $abcddabdc$ has the Parikh matrix

$$\Psi_{M_4}(abcddabdc) = \begin{pmatrix} 1 & 2 & 3 & 4 & 3 \\ 0 & 1 & 2 & 3 & 3 \\ 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}_{5\times5}.$$

An algorithm is developed for showing Parikh matrix corresponding to a binary word. This algorithm is extended to ternary and tertiary ordered alphabet. Parikh matrix is not injective. Two words may have the same corresponding Parikh matrix. This property is known as M-ambiguity or amiability. To meet the problem of M-ambiguity an algorithm is constructed for binary ordered alphabet which gives all the amiable words corresponding to a given Parikh matrix. This algorithm is extended to ternary ordered alphabet.

M-ambiguous words are the problem of Parikh matrix. In this study a system is introduced to represent binary words in a two dimensional field. It is seen that there are some relations among the representations of M-ambiguous words in the two dimensional field. This notion is extended to ternary ordered alphabet. In this case the representation is of three dimensional type.

A set of equations is introduced which helps to calculate the M- ambiguous words over binary alphabet. This set of equations is extended to ternary and tertiary ordered alphabets.

There is a trend to compare the amiable words by various distances between them. In the present study a type of distance named as Stepping distance is introduced. Applying Stepping distance two M-ambiguous words are compared.

For reducing the problem of M-ambiguity a function named as M-ambiguity reduction factor is introduced. If both the Parikh matrix and the M-ambiguity reduction factor corresponding to a word is considered together then the problem of M- ambiguity is reduced to a large extent.

From the very beginning of the study of computational linguistics the investigations are started from both axiomatic methods and grammatical methods. A natural language is the language which is naturally used by human beings in their day-to-day life. ‘Natural Language Processing’ (NLP) is the computerized approach to process natural languages. If it is possible to digitize natural languages then it will be helpful in machine translation. So it is tried to make some bridge between natural languages and arithmetic. In this connection some works are done in the present study. Many investigations have been done on development of context-free grammar of natural languages. Depending on those research work Parikh matrix is applied on the context-free part of natural languages. Bengali language is taken for a case study. The Parikh matrix of every Bengali letter is a 51×51 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. With the advancement of context-free grammar for Bengali language this effort may result to be 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.

*****