Relation between Language of Classes

Bhupender yadav
Department of Computer Science Engineering,
Om Institute of Technology & Management, Hisar, INDIA

Dr.Rohit Bajaj
Department of Computer Science Engineering,
Om Institute of Technology & Management, Hisar, INDIA

Abstract - In this paper we discuss about relation between language of classes we will explore how a string is processed in finite automata as regular language and the same string processed in context free grammar and recursive enumerable language.

Keywords:. finite Automata, non terminal, turing machine and terminal symbols.

I. I. INTRODUCTION

A language is a collection of sentences of finite length all constructed from a finite alphabet of symbols. A string in regular language is processed by finite Automata. Automata is a system in which energy material and information are transformed to perform some useful functions. At discrete time of interval we apply discrete inputs correspondingly outputs are come. Generally we apply input at initial states, this input processed on intermediate states and finally we reach to final state. The finite automata can be represented by as follows:

1. A finite set of states, often denoted Q
2. A finite set of input symbols, often denoted Σ
3. A transition function that takes as arguments a state and an input symbol and returns a state. The transition function is commonly denoted δ If q is a state and a is a symbol, then δ(q, a) is a state p (and in the graph that represents the automaton there is an arc from q to p labeled a)
4. A start state, one of the states in Q
5. A set of final or accepting states F (F ⊆ Q)

Notation: A DFA A is a tuple

$$A = (Q, \Sigma, \delta, q_0, F)$$

A string in context free grammer can be processed by parse tree a context free grammer can be represented by as follows.
G=(Vt, Vn, P, S) where
Vt= A finite set of terminals represented by small letter i.e. a, b………………….
Vn= A finite set of non-terminals represented by Capital letter i.e. A, B………………….
P=Production rules
S=Starting symbol and non terminal
A string in recursive enumerable language processed by turing machine a turing machine can be repsented by (Q, Σ, ∂, τ, q0 , b, F)
Q is a finite set of states
Σ, set of input symbol
∂ is a transition function
τ is a tape symbol
q0 is the initial stage
b is a blank symbol
f is a sub set of Q is the set of final states

II. REVIEW ON LITERATURE

The problem of learning relation between language of classes has an extensive history. To understand this history, we broadly divide results into those addressing the passive learning of finite automata, context free grammer, turing machine in first two approaches the learner has no control over the flow of data it receives,and also not able to decide where the data have to move The approaches to deal with the string are different.

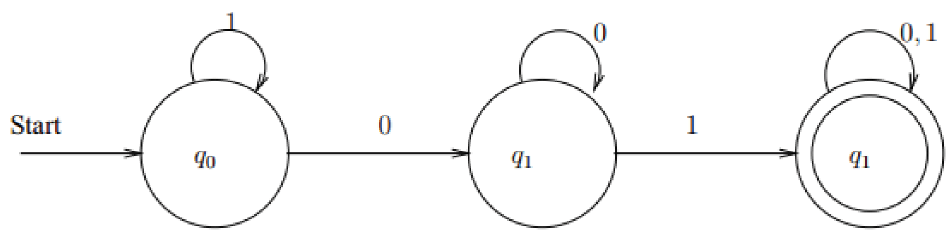
III. METHODOLOGY

The methodologies to processing the string in regular language, context free grammer and recursive enumerable language are different.The string in regular language is processed by finite automata as follows

$$A = (\{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\})$$

where the transition function δ is given by the table

		0	1
$\rightarrow$	q_0	q_2	q_0
$\star$	q_1	q_1	q_1
	q_2	q_2	q_1



Here the machine is represented by A, States are (q0,q1,q2), the input symbol are (0,1) and the transition function is given by the transition table and the string processing is done by the transition diagram shown in figure.

The string in context free language is processed by parse tree as follows:

As we know the cfg can be represented G = (Vn, Vt, P, S) with variables Vn, terminal symbols Vt, set of productions P and the start symbol from V called S.

A derivation tree is constructed with

1) each tree vertex is a variable or terminal or epsilon

2) the root vertex is S

3) interior vertices are from V, leaf vertices are from T or epsilon

4) an interior vertex A has children, in order, left to right, X1, X2, ... , Xk when there is a production in P of the form A -> X1 X2 ... Xk

5) a leaf can be epsilon only when there is a production A -> epsilon and the leafs parent can have only this child.

-A "derivation tree" sometimes called a "parse tree" uses the rules above: start with the starting symbol, expand the tree by creating branches using any right side of a starting symbol rule, etc.

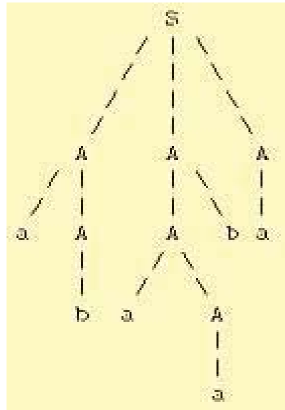
Vt: a, b

Vn: S, A

productions:S -> AAA | AA

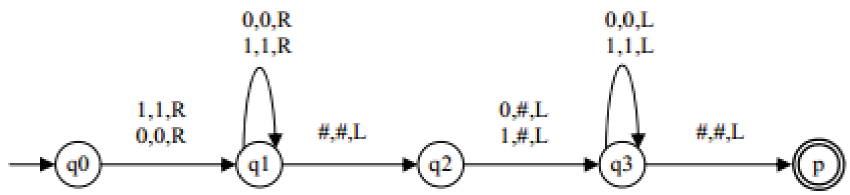
A-> AA | aA | Ab | a | b

String abaaba has derivation tree:



A string in recursive enumurable language is processed by turing machine as follows

Simple Eraser. This Turing machine reads strings in the language given by the expression (0,1)* and replaces the right-most symbol by a blank (#).



TM = ({q0,q1,q2,q3},{0,1},{0,1,#},δ,q0,{p}) where δ is given by:

Q \ Σ	0	1	#
q_0	(q_1 , 0 , R)	(q_1 , 1 , R)	∅
q_1	(q_1 , 0 , R)	(q_1 , 1 , R)	(q_2 , # , L)
q_2	(q_3 , # , L)	(q_3 , # , L)	∅
q_3	(q_3 , 0 , L)	(q_3 , 1 , L)	(p , # , L)

Then, the above Turing machine processes the input string "1110" as follows:

$\#q_0 1110\# \rightarrow \#1q_1 110\# \rightarrow \#11q_1 10\# \rightarrow \#111q_1 0\# \rightarrow \#1110q_1\# \rightarrow \#111q_2 0\# \rightarrow$

$\#11q_3 1\#\# \rightarrow \#1q_3 11\#\# \rightarrow \#q_3 111\#\# \rightarrow q_3\#111\#\# \rightarrow p\#\#111\#\#$

So these three methodologies are used to define the relationship between languages of classes.

IV. FINDINGS

In this paper we found how a string can be proceed using different languages used in Automata theory. every language have its own limitations and benefits. In a regular language when we get any string as an input,it encountered at initial state and passes through intermediate states and finally reached at final state. if the string is reached at final state then it is accepted by automata.

In the regular language the string is always moves left to right direction.

In the context free languages the string is processed by productions rules. with the help of productions rules we can processed the string to final state.in the context free grammar we also use stackfor the string processing to final state.

In the recursive enumerable language the string is processed by input symbol as well as tape symbol.the direction of string processing can be left or right direction which is depend upon control unit.

V. CONCLUSION

In this paper we observed that there are different languages to process the string in automata. A simple string can be processed by regular language, by the regular languagewe can only define that the string can be accepted or not. when we want to compare the two string we use thecontext free grammar with the help of stack and we process the assignment operator. Palindrome operations can also be performed using context free grammer.when we talk about the comparisonbetween three string we used recursive enumerable language.every algorithm and function canbe performed by universal turing machine which can be processed by recursive enumerable. Wecan also perform arithmetic,unary,logic operations through recursive enumerable language.

REFERENCES

[1] S.Even and R.E.Tarjan, "A combinatorial problem which is complete for polynomial space,"J.ACM 23:4(1976),pp. 710-719

[2] J.Hartmanis,P.M. lewis II, and R.E. Stearns, "Hierarchies of memory limited computations,"Proc. Sixth Annual IEEE Symposium on Swutching Circuit Theory and logical design(1965),pp.179-190.

[3] J.E. Hopcroft and J.D.Ullman,Introduction to Automata Theory,Languages and Computations Addison-Wesley,Reading MA.1979.

[4] D.E Knuth,The Art of computer Programming,Vol.II Seminumerical Algorithms,Addison- Wesley,Reading MA,1997(third edition).

[5] R.Motwani and P.Raghavan,Rondomized Algorithms,Cambridge Univ.Press,1995