CS-73

Q1. Define the following concepts fromaley:.

A) Kleene closure of a alphabet:-
Closure of an alphabet is definded by set of all possible strings over that alphabet including null string. (^. String) .in other words we can say that when a alphabet having kleene closure then the minimim string will be ^. It is represented by E*.
For eg:- E={a}
Then kleene closure of {a} will be E*= {^,a,aa,aaa,-----}

b) Finite Automata :-

A finite automaton can be represented by a 5 tuple (Q,E,δ,q0,F) Where

(i) Q is a finite non-empty set of states,
(ii) E is a finite non-empty set of input called input alphabet.
(iii) Δ is a function which maps Q*E→Q and is usually called direct transition function.
(iv) Q0 € Q is the initial state.
(v) F c Q is the set of final states .
The various component of finite automata are as follow:-
(i) Input tape
(ii) Reading Head
(iii) Finite control

© Godel Number:-

It is a mechanism of unique encoding of string .

Let

$W=d_{i1},d_{i2},d_{i3}$----------------------------------$d_{ik;}$ $K\geq 0$

I=position of the symbol

K=length of the string encoding of w using following formala will give an integer number which will be unique . That integer no is known as Godel no and represented by $g_n(w)$.

$G_n(w)=\beta_i^{k-1}$-----$+\beta_{ik-3}^{3}+\beta_{ik-2}^{2}+\beta^{1}{}_{ik-1}+ik$.

(d) Regular Expression:-

Regular expressions are useful and systematic way for representing certain sets of strings in an algebric fashion. Regular expression consists of some alphabet, and operators like (+,* etc).

Every regular expression can be recognised by a transition system for every string in the set of regular expression there exists a path from the initial state to a final state with path value of that string.

(e) Primitive Recursive Function:-

A function f over n, where n represent natural no, is called a primitive recursive function if,

(a) It is any one of the three initial functions(Z,S,U)
(b) It can be obtained by applying composition and recursion finite no of time of times to the set of initial functions.

(F) Unsolvable Problem:-

A problem P is called unsolvable problem if the language representing the problem is not decidable, or we can say there in not a definite algorithem so that we design a machine which recognizes the language L corresponding to the problem p.

The Halting problem of TM is a Type of unsolvable Problem.

(G) Turing decidable problem:-

A language L C E* representing a problem other E is said to be Turing –Decidable , if there is a Turing Machine M which always halt when any input W € E*, where w € L or

w €/L. in the other words we can say that a language L over an alphabet is said to be Turing Decidable if L and its complement (L⁻or E* - L) are Turing Acceptable.

(h) Moore Automata:-

A Moore Automata is also a part of finite automata which consists 6 tuples. This machine having property to produce output strings. The six tuples are as follow:-

Q→set of states

E→ set of alphabet

Δ(δ)→ Transition function

Q0→ inetial state , q0 ϵ Q

X →output function

Δ → output string

The representation of moore machine is as follow.

Q0/x -------------------------- > ---------------------------------- q1/y

Mean the output string depends upon current state only.

Q2 State and prove pumping lemma for finite automata.

And:- <u>Statement of pumping lemma:-</u>

The pumping lemma is used to prove certain sets are not regular by using several steps.

(i) Assume L is regular , let n be the number of states in the corresponding finite automata.

(ii) Choose a string w such that $|w| \geq n$. use pumping lemma to write w=xyz such that

$|y|>0$

$|xy| \leq n$

(iii0 find a suitable integer I such that $xy^i z \epsilon / L$. hence L is not regular.

e.g:- $L_{(m)}=\{a^n b^n | n\geq 1|$ is not regular

Let L be a regular language.

Take a string w=

Aaaaaaa bbbbbbb

k----------------→|k--------------→|

n n

$x=a^k$

$y=a^{n-k}$

$z=b^n$

so, for all $I \geq 1$, $xy^i z$ is alsi in L.

Let i=2

$Xy^2z = xyyz$

$=a^k.a^{n-k}.b^n$

$=a^{k+n-k+n-k}.b^n$

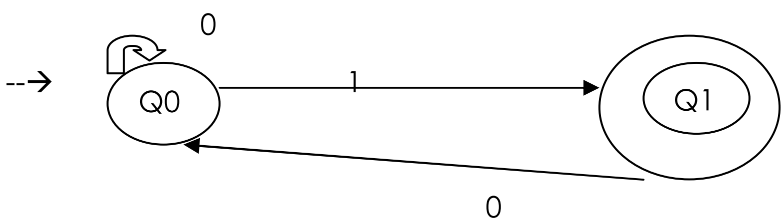
$=a^{2n-k}b^n$

Which not follow original pattern. So our assumpt was wrong. Hence the language is not regular.

Q(3) Construct a deterministic finite automata accepting the following set:-

$\{w \in \{0,1\}^*$:w is of form 010010001 0^i 1 ; i=1,2,3}

Ans:- The DFA of above strings will be



Where

Q={q0, q1}

E={0,1}

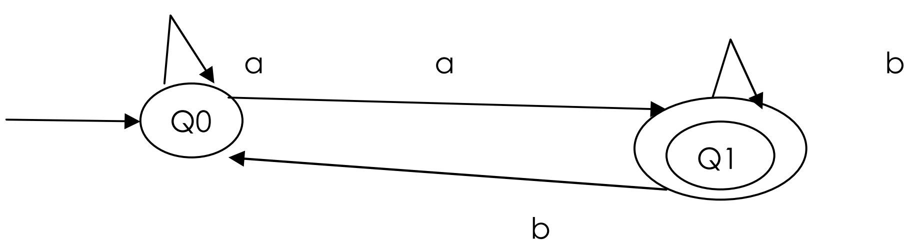
i.s={q0}

f.s={q1}

transition table

state	0	1
Q0	Q0	Q1
Q1	Q0	-

Q(4) describe infomally language accepted by the deterministic finite automata given below:-



ans: the language of the above DFA can be determined by the help of regular expression.

Q0=^ +q0a+q10-----------------------------------(1)

Q1=q0b+q1b--------------------------------------(2)

By eq(2)

Q1=q0b+q1b

R Q R P

Q1=q0b b*------------------------(3)

Put the value of q1 in e.q.(1)

Q0=^+q0a+q0b b*a

Q0=^+q0[a+b b*a]

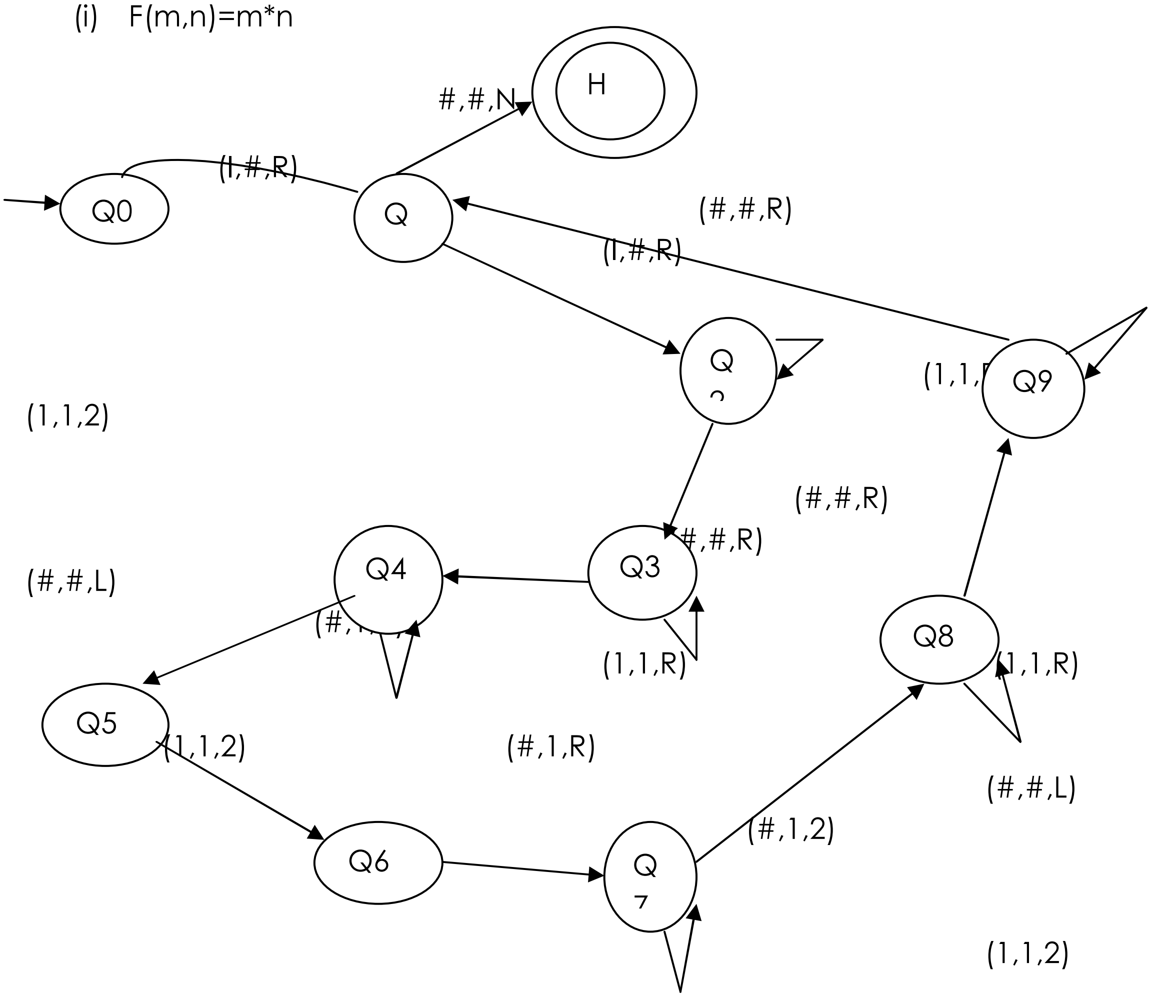
Q0=[a+b b*a]*--------------------(4)

Put the value of q0 in e.q.(3)

Q1=[a+b b*a]*b b* ans.

Q(5) Construct one turing Machine for computing each of the following functions.

(i) F(m,n)=m*n

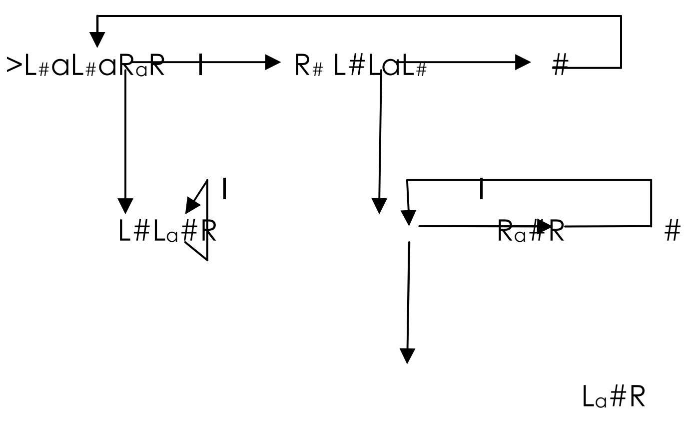


The transition table will be as follow:-

State alphabet	1	#
→ qo	#,R,q1	------
Q1	#, R, q2	#, R, H
Q2	1, R, q2	#, R, q3
Q3	1, R, q3	#, R, q4
Q4	1, R, q4	1, R ,q5
Q5	---	1, R ,q6
Q6	---	1, R ,q7
Q7	1, L,q7	#, R ,q8
Q8	1,L, q8	#, R ,q9
Q9	1, L, q9	#, R ,q1

(ii) $F(m,n) = \begin{cases} m+n & \text{if } m \geq n \\ 0 & \text{if } m < n \end{cases}$

Ans:- the given function is also called "Modulus Function" and represented by (m÷n), the TM for this function will be as follow:-



Where

$R_{\#}$ which finds the first blank square to the right of the currently scanned square.

$L_{\#}$ Which finds the first blank square to the left of the currently scanned square.

R_a moves one cell to the right and writes a in the place of new symbol scanned.

L_a moves one cell to the left and writes a in the place of the new symbol scanned.

Q(6) Construct one grammer for each of the following languages

(a) $a^m b^n$: m<n

ans:- the CFG for given language will be as follows:-

(a) < V,E,R,S >

Where

V={s,B}

E={a,b}

R={s→aSb

S→B

B→Bb

B→b

}, S: Starting variable

(b) { W∈ {0,1}* ; W=W^R}

G=< v, E, R, S >

Where

V= {S}

E= {0,1}

R={ S→0S0

S→1S1

S→0

s→1

}

S: Starting Variable

Q(7) show that each of the following functions is primitive recursive function.

(i) F(m,n)=4mn

Ans: step (a) put n=0

F(m,0)=4m*0

=z(m)

=0

Step (b) put n=n+1

F(m,n+1)=4m(n+1)

$=4mn+4m$

$=f(m,n)+4m$

$=F\{ f(m,n),4m\}$

[we know addition function is a primitive Recursive function for two parameters]

Where F represences additional function.

(ii) $F(m,n)=(5m)^{2n}$

Ans :- step (a) put n=0

$F(m,0)=(5m)^{2*0}$

$=(5m)^0$

$=1$

$=S(Z(m))$

Step (b) put n=n+1

$F(m,n+1) = (5m)^{2(n+1)}$

$=(5m)^{2n} * (5m)^{2n}$

$=f(m,n) * (5m)^{2n}$

In the last question we already prove that the multiplication is a primitive recursive function for two parameters , so that

$F(m,n)=(5m)^{2n}$ is also a primitive recursive function.