M.C.A, Third Semester Examination, 2014-15
Subject: Theory of computation.
(M.C.A-304)

Time: Three Hours] **[Maximum Marks : 60**

Note: Question Number 1 is compulsory. Answer any four questions from the remaining.

Q1. **(Give answer in short)** **Marks : 10X2**

----

i. **Given L1 = {a, ab, a^2} and L2 = {b^2, aba} are the languages over A = {a, b}, determine {a} L1L2 (b) L2L2**

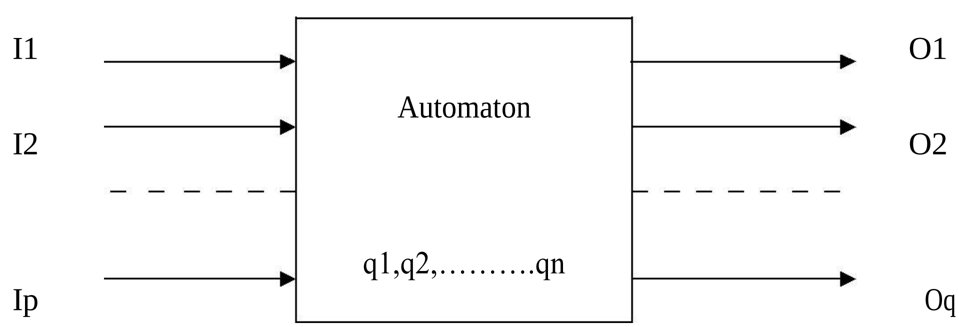
Ans: a) L1L2={ab^2aaba,abb^2,ababa,a^2b^2,a^2aba}

b) L2L2={b^2b^2,b^2aba, $abab^2$,abaaba}

----

ii. **Describe the model of discrete automaton.**

An automaton can be defined in an abstract way by the following figure.



Model of a discrete automaton

i) Input: - At each of the discrete instants of time t1,t2,…..input values I1,I2……… each of which can take a finite number of fixed values from the input alphabet ∑, are applied to the input side of the model.

ii) Output : - O1,O2….are the outputs of the model, each of which can take finite numbers of fixed values from an output O.

iii) States : - At any instant of time the automaton can be in one of the states q1,q2…..qn

iv) State relation : - The next state of an automaton at any instant of time is determined by the present state and the present input. ie, by the transition function.

Output relation : - Output is related to either state only or both the input and the state. It should be noted that at any instant of time the automaton is in some state. On 'reading' an input symbol, the automaton moves to a next state which is given by the state relation.

---

iii.. **What is the application of pumping lemma.**

Pumping lemma can be used to prove that certain sets are not regular.

Step-1 Assume L is regular. Let n be the number of states in the corresponding finite automation.
Step-II Choose a string w such that |w| >= n, Use pumping lemma to write w=xyz, with \xy\<= n and |y| > 0.
Step-III Find a suitable integer I such that xy^iz does not belongs to L. This contradicts the assumption . Hence L is regular.

Or

Give example to show a Language is not regular.

---

iv. **Write steps to remove a null move from an edge.**

Step1: Find all the edges d\starting from v_2.
Step2: Duplicate all these edges starting from v_1, without changing the edge lables.
Step3: If v_1 is an initial state, make v_2 also as initial state.
Step4: If v_2 is a final state, make v_1 as the final state.

---

v. **Given a Grammar G =({S}, {a, b}, S, P)with P defined as follows**
S→ aSb,
S→null

a) Obtain a sentence in language generated by G
b) Obtain the language L(G).

Ans: a) null or ab or aabb or aaabbb …..
b) $L=\{a^nb^n \mid n>=0\}$

---

vi. What **is the difference between leftmost and right most derivation?**

A derivation is called leftmost derivation if a production is applied only to the leftmost variable at every step. A derivation is called right most derivation if a production is applied only to the rightmost variable at every step.

---

vii**. Define the transition function δ in DFA and NFA.**

In DFA δ is the transition function mapping Q x ∑ into Q, Q is the set of States, ∑ is the set of input alphabets.

In NFA δ is the transition function mapping Q x ∑ into 2^Q which is the power set of Q, set of all subsets of Q.

---

viii. **What do you mean by bottom up parsing?**

In bottom up parsing the derivation tree is built from the given input string to the top (the root with label S) Example: String id+Id*id

Grammar: E→E+E
E→E*E
E→id

id+id*id
id+id*E (E→id)
id+E*E (E→id)
id+E (E→E*E)
E+E (E→id)
E (E→E+E)

---

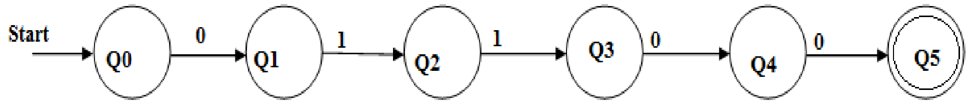
ix. **Define a parse tree.**

A derivation tree for a CFG G=(V_N, ∑, P, S) is a tree satisfying the following condition.

i) every vertex has a label which is a variable (non terminal) or terminal.
ii) The root has label which is non terminal S
iii) The label of an internal vertex is a variable.
iv) If the vertices n_1,n_2,……,n_k written with labels X_1,X_2, …..X_k are the sons of vertex n with label A, then A→ X_1,X_2, …..X_k is a production in P.
v) A vertex n is a leaf if its label is a ϵ ∑ or λ ; n is the only son of its father if its label is λ

---

x. **Design a NFA to accept the string 01100 only.**



Marks:4X10

Q2 **Given a grammar G with production rules** **6**

S →aB
S →bA
A →aS
A →bAA
A →a
B →bS
B →aBB
B →b

Obtain the (i) leftmost derivation, and (ii) rightmost derivation for the string "aaabbabbba".

(i)

S→aB (S→aB)
S→aaBB(B→aBB)
S→aaaBBB(B→aBB)
S→aaabBB(B→b)
S→aaabbB(B→b)
S→aaabbaBB(B→b)
S→aaabbabB(B→b)
S→aaabbabbS(B→b)
S→aaabbabbbA(S→bA)
S→aaabbabbba(A→a)

(ii) S→aB(S→aB)
S→aaBB(B→aBB)
S→aaBbS(B→bS)
S→aaBbbA(S→bA)
S→aaaBBbba(A→a)
S→aaaBbbba(B→b)
S→aaabSbbba(B→bS)
S→aaabbAbbba(S→bA)
S→aaabbabbba(A→a)

(b**) Explain the ID and move of aTuring machine. Indicate the major difference between Turing machine and Push Down automaton.** 4

An ID of a Turing machine M is a string αβγ, where β is the present atate of M, the entire input string is split αγ, the first symbol of γ is the current symbol a under the R/W head and γ has all subsequent symbols of the input string, and the string α is the substring of the input string fomed by all the symbols to the left of a.

Suppose x1x2……….xi-1 q xi…..xn |--- is the string before processing

Right move is defined as
(q,xi)=(p,y,R) then
x1x2……….xi-1 q xi…..xn |---
x1x2……….xi-1 y p xi+1…..xn

Left move is defined as

Let (q,xi)=(p,y,L) then
x1x2……….xi-1 q xi…..xn |---
x1x2……….p xi-1 y xi+1…..xn

Turing Machine: 1) R/W head on tape moves towards left and right direction.
2) No use of stack.
3) δ is defined as a mapping (q,x) onto (q',y,D) Where D is the direction of movement of R/W head
4) It accepts type 0 languages.(Because used to solve any kind of Computing function)
5) It performs following operation in one scan i) writing a new symbol in the cell being currently scanned ii) moving to the cell left of the present cell iii) moving to the cell right of the present cell iv) A change or no change in state

Pushdown Automation:1) R/W head on tape moves towards right direction only.
2) Uses stack.
3) δ is defined as a transition function from Q X (∑ U {λ}) X Γ to the set of finite subsets of $QX\Gamma^*$
4) It accepts context-free languages.
5) It performs two operation in one scan i) Change or no change on the top of the stack. ii) Change or no change of the state.

Q-3 a) Given a grammar G defined by the production rules **6**

S→AB
A→Aa
B →Bb
A→a
B →b

Show that the word w =a ^{2}b^4 ϵ L(G),Where L is a language determined by G.

S→AB(S→AB)
S→AaB(A→Aa)
S→aaB(A→a)
S→aaBb(B→Bb)
S→aaBbb(B→Bb)
S→aaBbbb(B→Bb)
S→aabbbb(B→b)

---

b**) Write a brief note on Linear Bounded Automaton.** 4

Linear bounded automation is described by the format
M=(Q,∑,Γ,δ,q0,b,C,$,F)
The input alphabet ∑ contains two special symbol C and $. C is called the left end marker and $ is called right end marker .The R/W never moves beyond the end mark.
Where Q= finite set of states

∑ = Input alphabet

Γ = is an alphabet called the stack

Q0= is the initial state q0 € Q

B is the blank symbol

F is set of final states F subset / equal to Q

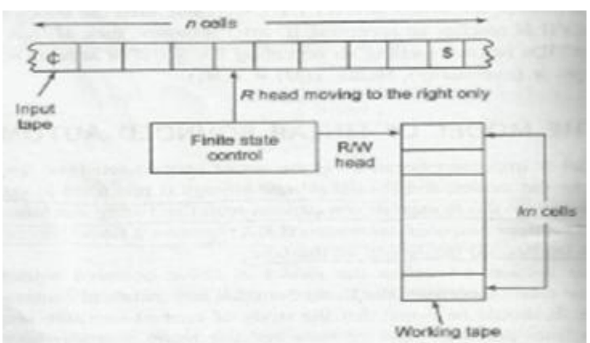


Fig. 9.11 Model of linear bounded automaton.

Components of LBA: Tapes, Finite control, R/W head

There are two tapes : one is called the input tape, and the other, working tape. On the input tape the head never prints and never moves to the left. On the working tape the head can modify the contents without restriction.
For left end of the tape
δ (q,C)→(p,C,R)
For Right end of the tape
δ (q,$)→(p,$,L)

Q-4 a) Give examples for Moore Model of finite automata with outputs. 4

Present State	Next State δ a=0	Next State δ a=1	Output λ
→q0	q3	q1	0
.q1	q1	q2	1
.q2	q2	q3	0
.q3	q3	q0	0

b) Convert the following NFA to DFA 6

State Q	Inputs 0	1
->q0	q0, q3	q0,q1
q1	---	q2
*q2	q2	q2
q3	q4	------
*q4	q4	q4

Ans:

	0	1
Q0	Q0,Q3	Q0,Q1
Q0,Q1	Q0,Q3	Q0,Q1,Q2
*Q0,Q1,Q2	Q0,Q3	Q0,Q1,Q2

Q0,Q3	Q0,Q3,Q4	Q0,Q1
*Q0,Q3,Q4	Q0,Q3,Q4	Q0,Q1

Q-5 (a) Show that the grammar G with production **5**

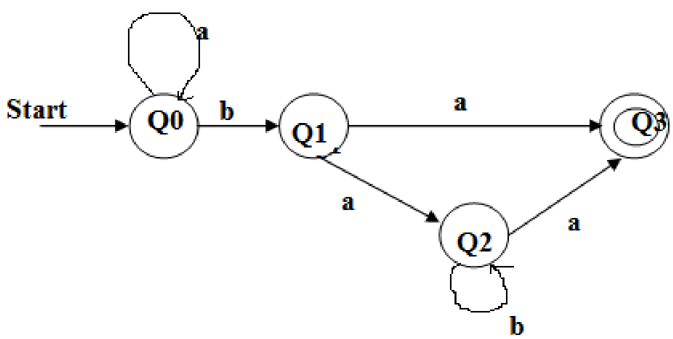
S →a| aAb| abSb
A→aAAb| bS
is ambiguous.

S→abSb (S→abSb)
S→abab (S→a)

S→aAb
A→abSb(A→bS)
A→abab(S→a)

The string abab is derived from two different production rule and generates two different derivation tree. So the grammar is ambiguous.

(b) Write Regular Grammar for the expression a*b(a+ab*a) **5**



A0→aA0
A0→bA1
A1→aA2
A2→bA2
A2→aA3
A2→a
A1→aA3
A1→a

Q-6 (a)Given the CFG with P given by **5**

S → AB
A →aAA| null
B→ bBB | null

Eliminate the null productions to obtain P for an equivalent CFG.

Because A→null and B→null
W1= { A,B}
A→aAA | aA|a
B→bBB|bB|b

W2={A,B,S} as S→AB, A→aAA, B→bBB
S→AB|A|B|null

S→AB|A|B|null
A→aAA | aA|a
B→bBB|bB|b

6.b) Write abrief note on PDA.

A PDA is defined as 7 tuple notation

M=(Q,∑, ┌, δ,q0,z0,F)

Where Q= finite set of states

∑ = Input alphabet

┌ = is an alphabet called the stack

Q0= is the initial state q0 € Q

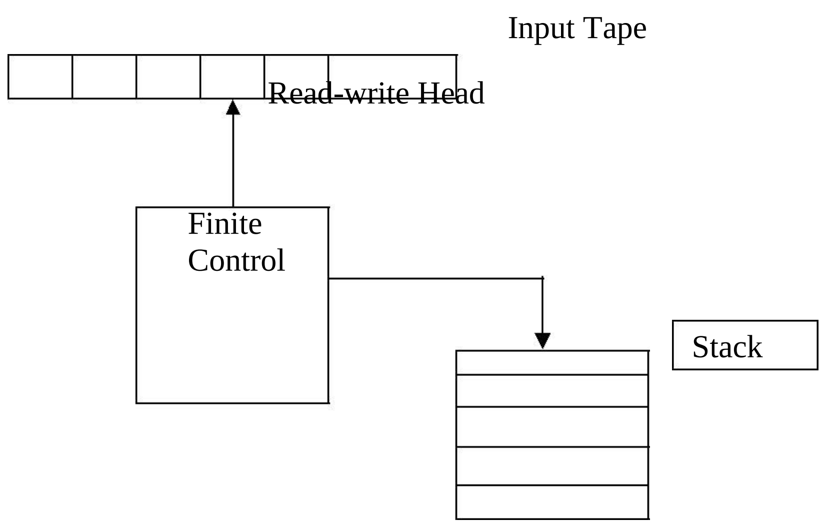
F is set of final states F subset / equal to Q

δ is a transition mapping δ = QX (∑ U { €}) X ┌ -> Q X $┌^*$

Basic model of PDA consists of 3 components:

i) an infinite tape
ii) a finite control
iii) a stack

Now let us consider the 'concept of PDA' and the way it 'operates'.



PDA has a read only input tape, an input alphabet, a finite state control , a set if initial states, and an initial state . in addition it has a stack called the pushdown stack. It is a read-write pushdown store as we add elements to PDS or remove element from PDS. A finite automation is on some state and on reading, an input symbol moves to a new state. The push down automaton is also in some state and on reading an input symbol , the topmost symbol ,it moves to a new state and writes a string of symbols in PDS.

Example 1: Construct a PDA that accepts the language $\{a^n b^n \mid n \geq 0\}$.

$M = (Q, \Sigma, \Gamma, \delta, q_1, Z, F)$

$Q = \{q_1, q_2, q_3, q_4\}$

$\Sigma = \{a,\ b\}$

$\Gamma = \{a,\ b,\ z\}$

$F = \{q_1,\ q_4\}$, and δ consists of the following transitions

1. $\delta(q_1, a, z) = \{(q_2, az)\}$
2. $\delta(q_2, a, a) = \{(q_2, aa)\}$
3. $\delta(q_2, b, a) = \{(q_3, \in)\}$

4. $\delta(q_3, b, a) = \{(q_3, \in)\}$
5. $\delta(q_3, \in, z) = \{(q_4, z)\}$

$(q_1, aabb, z) \vdash (q_2, abb, az)$ (using transition 1)

$\vdash (q_2, bb, aaz)$ (using transition 2)

$\vdash (q_3, b, az)$ (using transition 3)

$\vdash (q_3, \in, z)$ (using transition 4)

$\vdash (q_4, \in, z)$ (using transition 5)

q_4 is final state. Hence ,accept. So the string *aabb* is rightly accepted by *M*.

Q7. (a) Obtain a grammar in Chomsky Normal Form (CNF) equivalent to the grammar G
with productions P given by
5

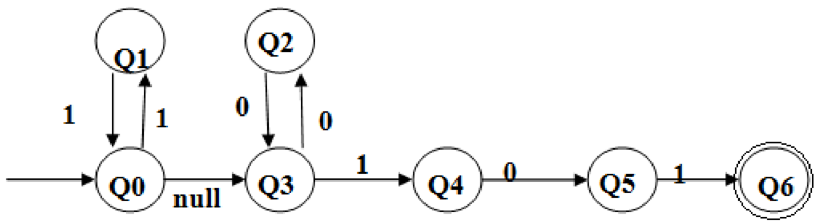
S→ ABa
A→aab
B→ AC

S→ABa
S→X1X2
X1→AB
X2→a
A→aab
A→X3X4
X3→aa
X4→b
X3→X2X2
B→AC

S→X1X2
X1→AB
X2→a

A→X3X4
X4→b
X3→X2X2
B→AC

(b) Explain the procedure to convert NFA with null move to ordinary NFA with examples.



Step1: Find all the edges d\starting from v_2.

Q3→Q2
Q3→Q4

Step2: Duplicate all these edges starting from v_1, without changing the edge lables.

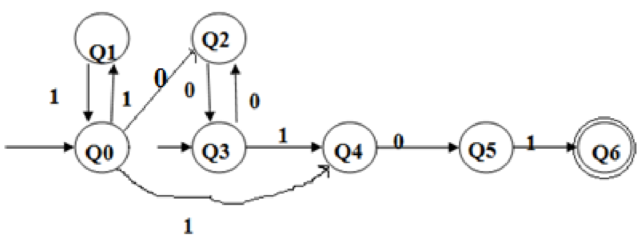
Q0→Q2
Q0→Q4

Step3: If v_1 is an initial state, make v_2 also as initial state.

Not Applicable

Step4: If v_2 is a final state, make v_1 as the final state.

Not Applicable



Q-8. a) Obtain a Grammar which generates the language. **4**

L ={a^nb^n : n >=0}

S→aSb

S→null

---

b) Design a PDA that accepts L={ a ^{n}b^{n} c ^{m} where n,m >=1} .Show acceptability of a string by the PDA by taking suitable example . 6

$\delta(Q0,a,Z_0)\rightarrow(Q0,a\ Z_0)$ R1

$\delta(Q0,a,a)\rightarrow(Q0,a\ a)$ R2

$\delta(Q0,a,b)\rightarrow(Q1,null)$ R3

$\delta(Q1,a,b)\rightarrow(Q1,null)$ R4

$\delta(Q1,c,Z_0)\rightarrow(Q1,Z_0)$ R5

$\delta(Q1,null,Z_0)\rightarrow Qf$ R6

Example aabbccc

$\delta(Q0,a,Z_0)\rightarrow(Q0,a\ Z_0)$ R1

$\delta(Q0,a,a)\rightarrow(Q0,a\ a)$ R2

$\delta(Q0,b,a)\rightarrow(Q1,b,null\ )$ R3

$\delta(Q1,b,a)\rightarrow(Q1,b,null\ )$ R4

$\delta(Q1,c,Z_0)\rightarrow(Q1,Z_0)$ R5

$\delta(Q1,c,Z_0)\rightarrow(Q1,Z_0)$ R5

$\delta(Q1,c,Z_0)\rightarrow(Q1,Z_0)$ R5

$\delta(Q1,null,Z_0)\rightarrow Q_f$ R6
