M.Sc. (Third Semester) Examination,2013
Theory of Computation
Paper: Fourth

---

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

a) L1L2 = { abb,aaba,abbb,ababa,aabb,aaaba}
b) L2L2 = { bbbb, bbaba, ababb,abaaba}

---

ii) Type 1 Grammar:

A grammar is called type 1 or context dependent if all its production are type 1 productions.

Type 1 production : A production of the form φAψ->φαψ is called a type 1 production if α not equal to null Where φ is the left context ψ is the right context A € V_n and α $(V_n \cup \sum)^*$

The production S-> null is also allowed in a type 1 grammar but in this case S does not appear on tht right hand side of any production.

Example:

2A->1B
B->0

---

iii) In bottom up parsing parsing takes place from the terminal nodes to the root node. In bottom-up parsing the derivation tree is traversed from the given input string to the start of the grammar symbol. Ex: S-> AB

A-> a

B-> b

For string ab

=aB (Using transition rule B->b)

=AB (Using transition rule A->a)

=S (Using transition rule S->AB)

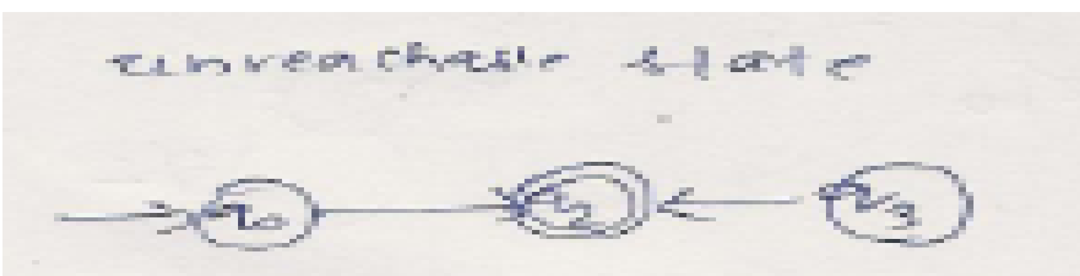
----

iv) a) ba+ab

b) $aa(a^*)$

----

v) Unreachable states are the states which are not **reachable from the initial** states upon the application of **any input sequence.** In the following diagram Q3 is unreachable state.



----

vi) 1100 + 1010 + 111

----

vii)

Left move :

Suppose δ (q,xi) =(p,y,L)

Id before processing

X1,x2…………………….xi-1 q xi ………………..xn

After processing

X1,……………….xi-2 p xi-1 y xi+1……………xn

Right move:

Suppose δ (q,xi) =(p,y,R)

Id before processing

X1,x2…………………….xi-1 q xi ………………..xn

After processing

X1,……………….xi-2 xi-1 y p xi+1……………xn

----

viii)

In leftmost derivation, at each step, leftmost non terminal is replaced by its RHS given in the production rule. The symbol ==> is used to denote leftmost derivation

In rightmost derivation, at each step, rightmost non terminal is replaced by its RHS given in the production rule. The symbol ==> *is used to denote rightmost* derivation

Example: S-> AB

A->a

B->b

Leftmost derivation:

S->AB

->a B

-> Ab

Right tmost derivation:

S->AB

->a B

-> Ab

----

ix) A type 2 production is a production of the form A->α where A € V_n and α € $(V_n \cup \sum)^*$

Example S-> Aa, A->a

----

x)

A language is called type 1 or context dependent if its grammar contains all its production as type 1 productions. The production S-> null is also allowed in a type 1 grammar but in this case S does not appear on tht right hand side of any production.

Type 1 production : A production of the form φAψ->φαψ is called a type 1 production if α not eqal to null

2A->1B

B->0

A Language is called a context free Language if its grammar contains all type 2 production language

A type 2 production is a production of the form A->α where A € V_n and α (V_n U ∑)*

Example S-> Aa,   A->a

--- -------------------------------

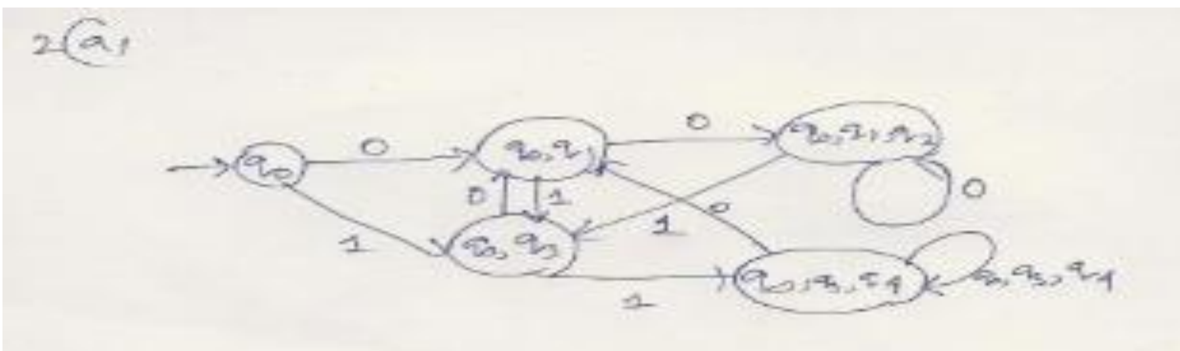
SECTION-B

2.a

Transition table

	0	1
Q0	Q0,Q1	Q0,Q3
Q0,Q1	Q0,Q1,Q2	Q0,Q3
*Q0,Q1,Q2	Q0,Q1,Q2	Q0,Q3
Q0,Q3	Q0,Q1	Q0,Q3,Q4
*Q0,Q3,Q4	Q0,Q1	Q0,Q3,Q4

Transition Diagram



Deterministic and Non deterministic Finite Automata

-------------------------- ---

2.b

Deterministic finite automata (DFA)

Each state of an automaton of this kind has a transition for every symbol in the alphabet.

Deterministic Finite Automata can be defined as M=(Q,∑,δ,q0,F) where Q is the set of states

∑ is the input symbols

δ is the transition function Q x ∑
Q

q0 is the start state

F is the final state

Non deterministic finite automata (DFA)

Non Deterministic Finite Automata can be defined as M=(Q,∑,δ,q0,F) where Q is the set of states

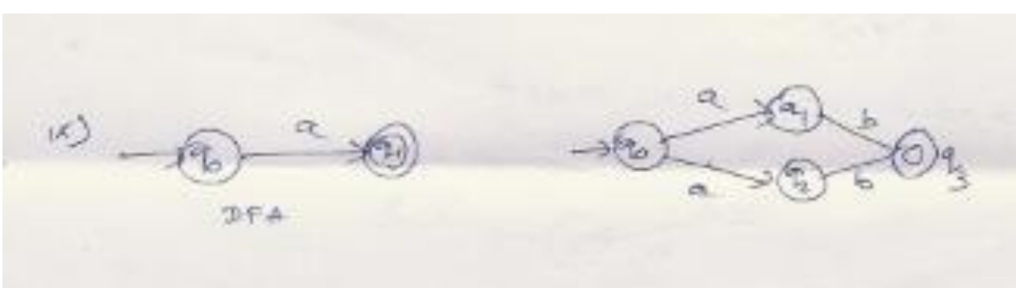
∑ is the input symbols

δ is the transition function Q x ∑
2^Q

q0 is the start state

F is the final state

Diagrams for DFA and NFA



-- --------------------

---

3. i) L= { a^n: n>0}

L= {a,aa,aaa,aaaa,………………………..}

Let w=aaa

And x=a y=a z=a

If i= 1 w=aaa belongs to L

If i=2 w=aaaa belongs to L

If i=3 w=aaaaa belongs to L

If i=4 w=aaaaa belongs to L

If i=5 w=aaaa belongs to L thus regular

----

ii) L= { a^p: p>0}

L= {a,aa,aaa,aaaaa,aaaaaaa,aaaaaaaaa…………………..}

Let w=aaa

And x=a y=a z=a

If i= 1 w=aaa belongs to L

If i=2 w=aaaa does not belongs to L thus not regular

3.b

Type 0 production: A production with no restriction .Any type of production. Ex A->a

Type 1 production : A production of the form φAψ->φαψ is called a type 1 production if α not eqal to null

2A->1B

B->0

A type 2 production is a production of the form A->α where A € V_n and α (V_n U ∑)*

Example S-> Aa, A->a

A type 3 production is a production of the form A->a or A-> aB where A , B € V_n and a ϵ∑

Example B-> aC, A->a

4.a

i) a or b followed by b or c over the alphabet a,b,c

ii) all binary number precced by 1

iii) String accepting only zero

iv) Set of string over 0 and 1 where 0 or 1 is followed by zero or any numbers of 11

4.b Context free Language: Languages formed from contex free grammar is contex fre e language

Definition(Contex free grammar)

A CFG can be defined as G=(V,T,P,S) where V is the set of non terminals, T is the set of terminals, S is the start symbol and P is the set of productions of the form A->α where A belongs to V_N, α϶(VUT)*.

Derivation tree (Parse tree)

The derivation in a CFG can be represented by using trees called 'derivation tree' or 'parse tree'. A derivation tree for a CFG is a tree satisfying the following :

i) every vertex has a label which is a variable (non terminal) or terminal.

ii) The root has label which is non terminal

iii) The label of an internal vertex is a variable.

ie, a derivation tree is a labeled tree in which each internal node is labeled by a non terminal and leaves are labeled by terminals. Strings formed by labels of the leaves traversed from left to right is called the 'yield of the parse tree'. Ie, the yield of a derivation tree is the concatenation of the labels of the leaves without repetition in the left-to-right ordering.

Eg: Let G=({S,A},{a,b},P,S) where P is
defined as S-> aAS/a, A->b

S->aAS->aaASS->aabaa

A language is called type 1 or context dependent if its grammar contains all its production as type 1 productions. The production S-> null is also allowed in a type 1 grammar but in this case S does not appear on tht right hand side of any production.

Type 1 production : A production of the form φAψ->φαψ is called a type 1 production if α not eqal to null

2A->1B

B->0

A Language is called a context free Language if its grammar contains all type 2 production language

A type 2 production is a production of the form A->α where A € V_n and α $(V_n \cup \sum)^*$

Example S-> Aa, A->a

----

5. a) A0->0A0

A0->1A1

A0->0A2

A1->0A3

A3->1A4

A4->1A4

A4->0A5

A5->0A6

A5->0

----

5. B Unit removal

A->D

D->E

E-> F   are I unit production

As E-> F and F-> aS

So E-> aS

Now A->D

D-> E

As D-> E and E->aS

So D->aS

Now A-> D
And D->aS

So A-> aS

After unit removal S->AaB | aaB, A-> aS , B->bbA | null

Null removal

Phase-I

W0 = { B } as B-> null

W1= { B, S }

As S-> AaB and S-> aaB

Phase-II

S->AaB | aaB | Aa | aa

A->aS

B-> bbA

---

6. a)   Phase 1: W0= { A, S}

As A->b and S->a

Phase II: S->a

---

b) d(q0,B)= (q0,B,R)

d(q0,a)= (q1,B,R)

d(q1,a)= (q1,a,R)

d(q1,b)= (q1,b,R)

d(q1,B)= (q1,B,L)

d(q1,b)= (q2,B,L)

d(q2,b)= (q3,B,L)

d(q3,b)= (q4,B,L)

d(q4,b)= (q4,b,L)

d(q4,a)= (q4,b,L)

d(q4,B)= (q0,B,R)

d(q0,null)= qf

----

7. a) S-> AB where

A->aA and B-> bB

A-> aA (Not in CNF)

A-> AA

A->a

B->bB (Not in CNF)

B->BB

B->b

Or

S-> X1X2

X1->aA

X1->AA

X2->bB

X2->BB

B->b

i) CNF= S->AB

A->AA

B->BB

A->a

B->b

Or CNF= S->X1X2

X1->AA

X2->BB

A->a

B->b

7.b.

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 ┌*

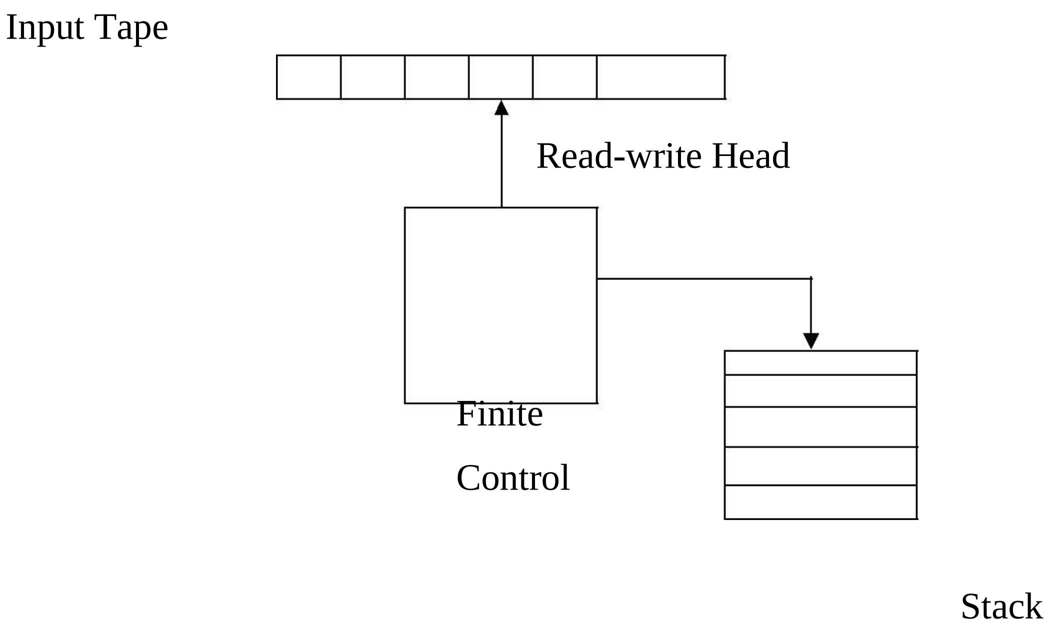
Basic model of PDA consists of 3 components:

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*.

---

8. a) A context free grammar G such that some word has two parse trees is said to

be ambiguous. A grammar which generates two or more parse tree for the same grammar.

The given grammar is ambiguous because for the same string abaa it produces two derivation tree by using the derivations

S->SbS->abS->abSa->abaa S->Sa->SbSa->Sbaa->abaa

The parse tree of the above derivations are different . Thus the language is ambiguous.

c) DFA accepting 111

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

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

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

Q0,Q3	Q0,Q3	Q0,Q1,Q3
Q0,Q1,Q3	Q0,Q3	Q0.Q1,Q2,Q3
