Name : M. N. KHAN

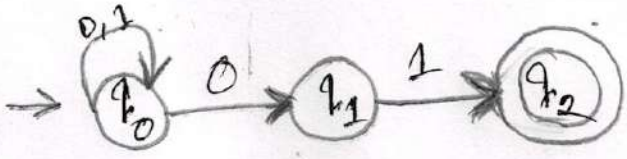
ID : 221C080041

Course code : CSI-317

8th batch (evening) CSE

Ans to the NO 01

(a) Ans: L = {w | w is binary string ends in 01}



Final state

(b) Ans:

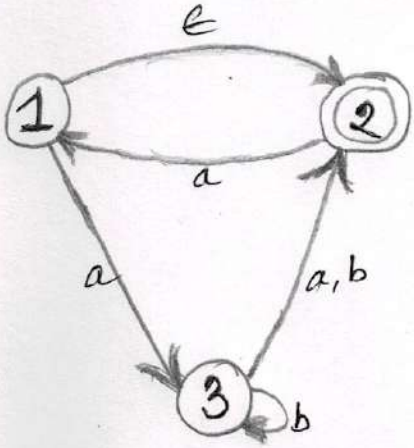
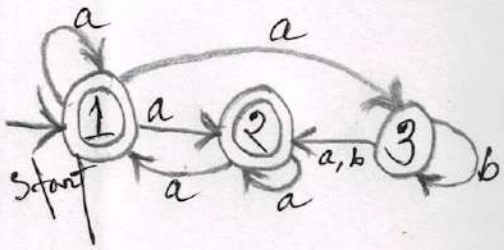


Fig : NFA

Transection table for ϵ-NFA :→

⇒ NFA transaction Diagram-



ϵ-NF	a	b	ϵ
→ 1	{3}	∅	{2}
2	{1}	∅	∅
3	{2}	{2,3}	∅

1b

Transection table for Equivalent NFA

NFA	a	b
→ 1	{1,2,3}	∅
2	{1,2}	∅
3	{2}	{2,3}

1

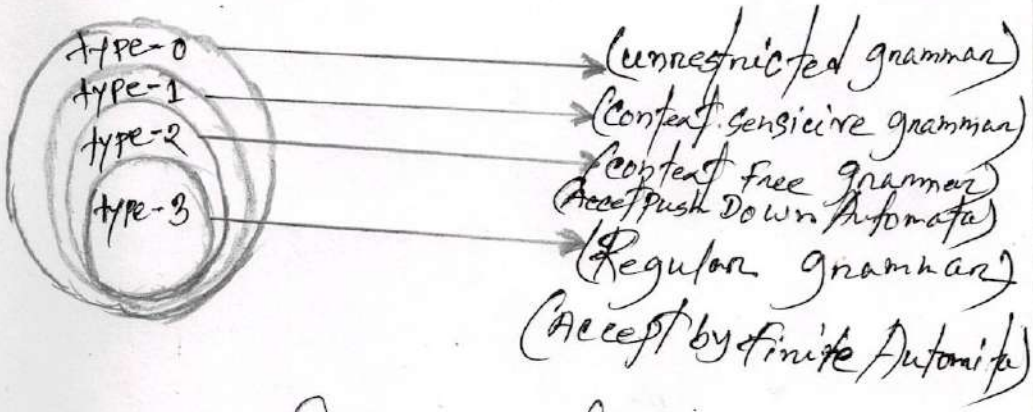
[C] Ans: Non-terminal are the Non-leaf Nodes in a parsetree. In the Expression grammar. E, TA, F are not terminals. some times non terminals are Enclosed between angle brackets to distingvish them from terminals,

<digit> and <Infer> are non terminals symbols and, S, A. are non terminal symbols.

Ans: to the Q: NO: 03

(a) Ans: According to chomsky hierarchy, Grammar is divided into 4 types as fallow:-

1. Type 0 is know as Unrestricted grammar.
2. Type 1 is known as context-sensitive grammar.
3. Type 2 is knowas context-free grammar.
4. Type 3 Regular grammar.



1. Type-0: Type 0 grammars include formal grammar. Type 0 Grammar Language are recognized by turing machine.

$\alpha \rightarrow \beta$ where,

α is $(V+T)^{*} V (V+T)^{*}$

V: variable

T: Terminals

β is $(V+T)^{*}$

Example:

$Sab \rightarrow bA$

$A \rightarrow S$

variable S, A and

Terminals a, b.

3
a

type 1: Context - Sensitive grammar:

→ First of all type 1 Grammar should type 0

→ Grammar Production in the from of

$$\alpha \rightarrow \beta$$

$$|\alpha| <= |\beta|$$

Example:

S --> AB

AB --> abC

B --> b.

type 2: context - free grammar generate context-free Language. The Language generated by the grammar is recognized by a Pushdown automata. in type-2.

Example:

S --> AB

A --> a

B --> b

type - 3: Type - 3 grammar generated regular languages. These Language are exactly all Language that can be accepted by finite - state Automation

Example:

S --> a.

3

[b] Ans: Writing down the differences between regular Language and context free language :-

Regular grammar :

* It is accepted by finite state automata.
* It is a subset of type-0, type-1 and type 2 grammars
* This Language it generated is called Regular Language.
* Regular Language are called / closed under operation like union-intersection, complement etc
* they are the most restricted form of grammar.

context grammar :

* Language generated by context free grammar is accepted by pushdown Automata.
* It is a subset of type 0 and type 1 grammar and superset of type 3 grammar.
* Also called phase structured grammar.
* Only one parse tree → unambiguos.
* More than One parse tree → Ambiguous.

3/

[C] Ans: DFA for regular Expression, $R = 1^*00^*1(0+1)^*$

⇒ DFA –

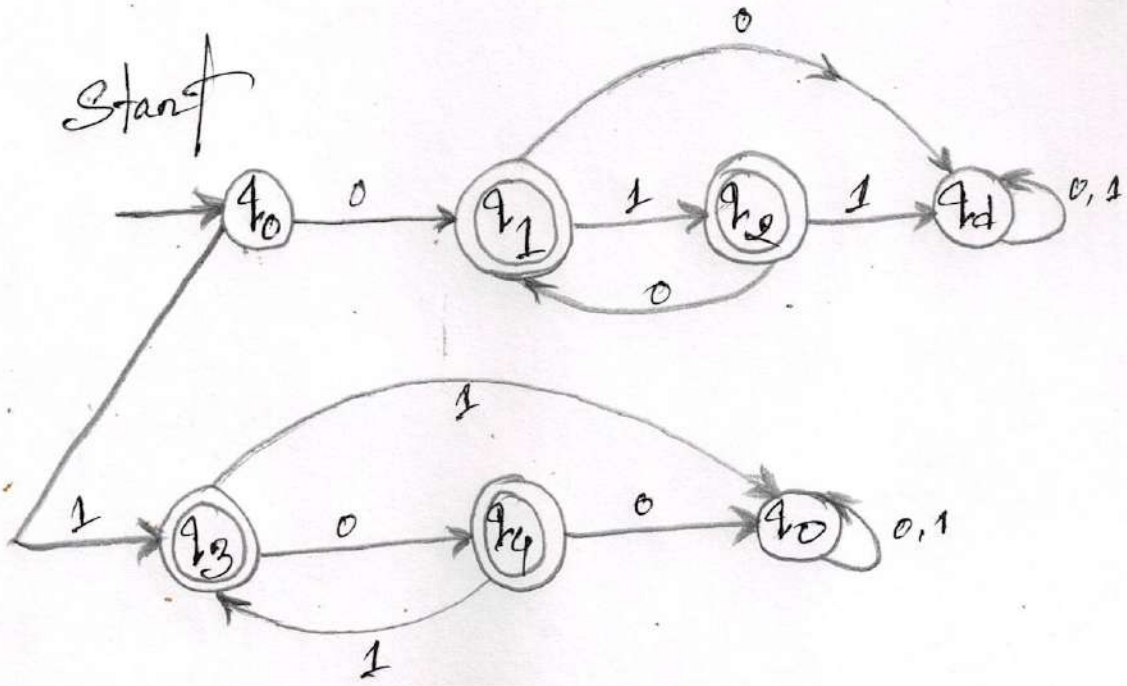


Fig :- $R = 1^*00^*1(0+1)^*$

Ans: to the Q: NO: 04

@ Ans: npda for the Language $L = \{a^m b^n c^n : m, n \geq 1\}$

→ Design a non deterministic PDA for accepting the Language $L = \{a^m b^n c^m \; n \geq 1\}$, ie:

$L = \{abc, abcc, abccc, aabbc, aaabbcc, aaabbbbccccc\}$

In each of the string the number of a's is equal to number of b's and the number of c's is independent of the number of a's and b's. This Problem is fact similar to the NPDA for accepting Language $L = \{a^n b^n \mid n >= 1\}$ the only difference is here we add.

→ Explanation; Here we need to maintain the order of a's, b's, c's. That is all the a's are coming first and then all the b's and then c's are coming. Thus, we need stack along with the stat Diagram. The count of a's and b's is maintained by stack, we will take 2 stack Alphabets.

$$\Gamma = \{a, Z\}$$

Where are Γ = set of all stack Alphabet

Z = Stack start symbol.

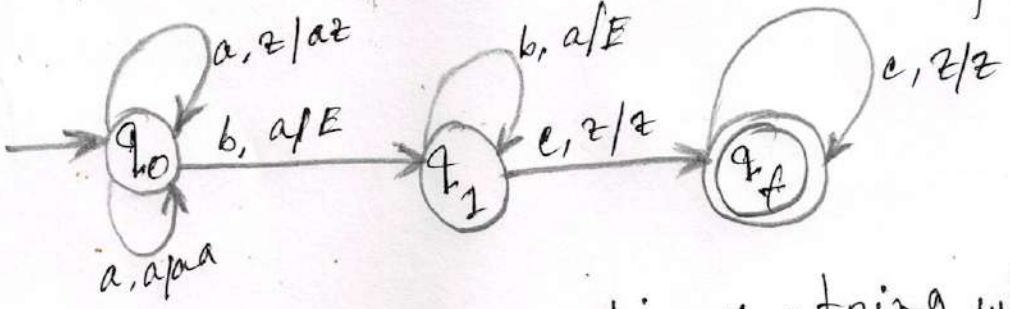
4/a/ PDA stack transition function →

$$\delta(q_0, a, z) \vdash (q_0, az)\ \delta(q_0, a, a) \vdash (q_0, aa)\ \delta(q_0, b, a) \vdash (q_1, a)\ \delta(q_1, b, a) \vdash (q_1, a)\ \delta(q_1, c, a) \vdash (q_2, c, a) \vdash (q_2, \varepsilon) \vdash (q_f, \varepsilon, z) \vdash (q_f, z)$$

where,

q_0 = Initial state q_f = final state

ε = Indicated Pop. Operation

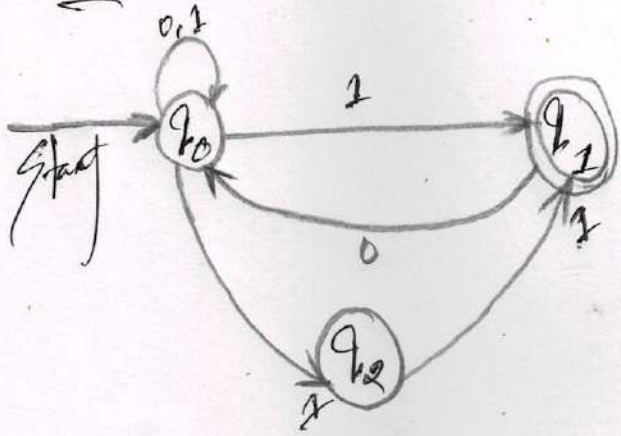


4/b// Ans: L = {w | w is a binary string which contains the substring 11}

→ q_0 : Start state (initially off) also means the most recent input was not a 1.

→ q_1 : Has never seen 11 but the most recent input was a 1

→ q_2 : Has seen 11 at least once.



Ans: the Q: No: 05

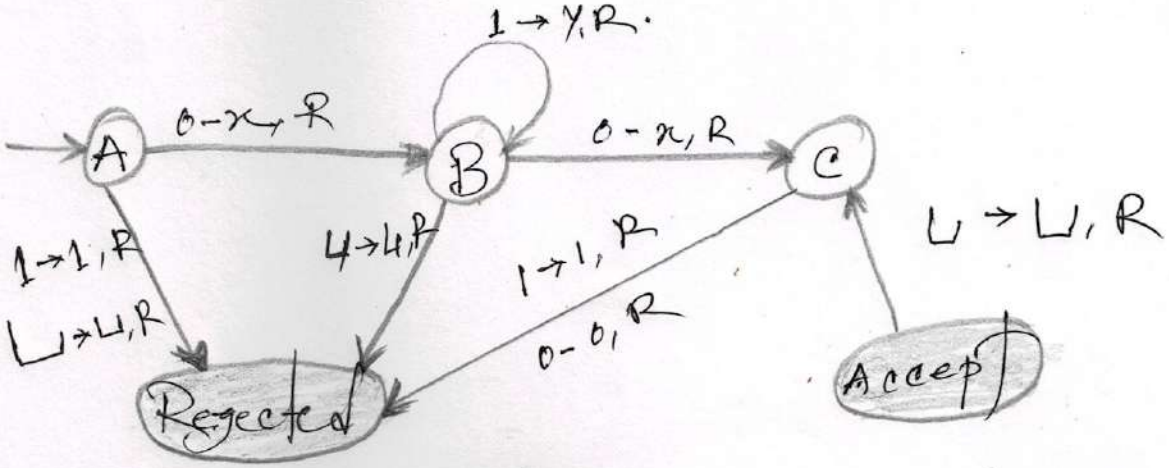
a) **Ans: Turing machine:** A Turing Machine (TM) is a mathmatical model wich consists of an infinie bngth tape derided into calls on whic input isgiven. It consists of a head witch reads the input tape.

A state registen stores the state of the turning machine. After reading on input symbol, it is reaplaced with another symbol. Its Intenal state, is changed, and it maves from one call to the right on left. If the TM reaches the final state, the input string is accepted, othewise rejected.

b) **Ans:** Turning machine the recognizes,

$\{L_0(0+1)^*1\}$ ⟶

$\Sigma = 01$ $\quad L = 1^*1$ $\quad b = \sqcup$



5. Ans: Empty stack is (). Is use to check if a stack is empty or not-

This method requires no parameters. It Return true. If the stack is empty and False if the stack is not empty.

std:: stack: POP: Removes the element on top of the stack effectively reducing its size by one. The element removed is the latest elements. Interested in to the stack.

whose value can be retrieved by calling member stack :: top.