Name – Aishwarya Kaiwart
– 19207701
– Theory of computation
– 07/10/2021
– BSc 5th sem
CT1.

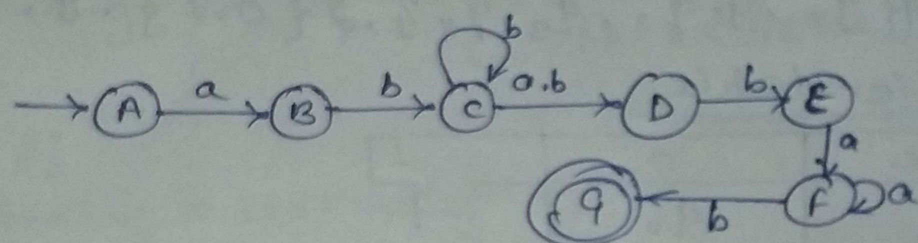
PART-A

Ans ② NFA refers to Nondeterministic finite Automata. A finite Automata is said to be non deterministic, if there is more ~~to be no~~ than one possible transition from one state on the same input symbol.

Difference between NFA & DFA

DFA	NFA
↳ Deterministic finite Automata.	↳ NFA stands for Nondeterministic finite automata.
↳ DFA cannot use empty string transition.	↳ NFA can use empty string transition.

Ans ①



Ans ⑤ Context free language:– CFL is a language which is generated by a context free grammer or type 2 grammer and gets accepted by a pushdown Automata.

Ans ⑥ Closure of a regular set is regular.

IF $L = \{a, aaa, aaaaa, \ldots\}$

i.e. $RE(L) = a(aa)^*$

$L^* = \{a, aa, aaa, aaaa, aaaaa, \ldots\}$

~~excluding~~.

$RE(L^*) = a(a)^*$

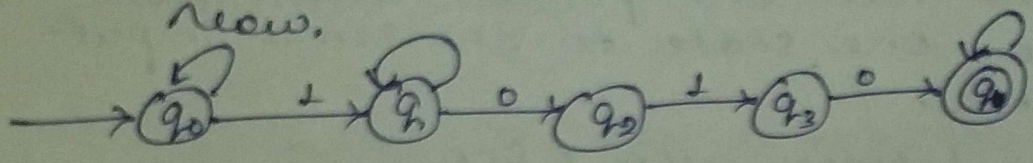
PART B

Ans 5) Design a DFA that accepts strings containing 1010 as substring.

we know from DFA,

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

Now,



Here,

Q : finite set of states

Σ : Input symbols

δ : Transition function

q_0 : Initial state

f : final state

$\therefore A = (\{q_0, q_1, q_2, q_3, q_4\}, \{0,1\}, \delta, q_0, \{q_4\})$

Transition table

State	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_2	q_1
q_2	ϕ	q_3
q_3	q_4	ϕ
$*q_4$	q_4	q_4

Now, transition function for 1010,

$\delta(q_0, \varepsilon) = q_0$

$\delta(q_0, 1) = \delta(\delta(q_0, \varepsilon), 1) = \delta(q_0, 1) = q_1$

$\delta(q_0, 10) = \delta(\delta(q_0, 1), 0) = \delta(q_1, 0) = q_2$

$\delta(q_0, 101) = \delta(\delta(q_0, 10), 1) = \delta(q_2, 1) = q_3$

$\delta(q_0, 1010) = \delta(\delta(q_0, 101), 0) = \delta(q_3, 0) = q_4$

Ans① If L1 and L2 are two context free languages, their intersection L1 ∩ L2 need to be context free. for example.

$L1 = \{a^n b^n c^m \mid n >= 0 \text{ and } m >= 0\}$ and

$L2 = \{a^m b^n c^n \mid n >= 0 \text{ and } m >= 0\}$

$L3 = L1 \cap L2 = \{a^n b^n c^n \mid n >= 0\}$ need not be context free.

L1 Says number of a's should be equal to number of b's and L2 says number of b's should be equal to the number of c's. Their intersection says both condition need to be true, but pushdown automata can compare only two so it cannot be accepted by ~~it can~~ pushdown automata.

Hence not context free.

~~The language L = {a}~~

[PART A]

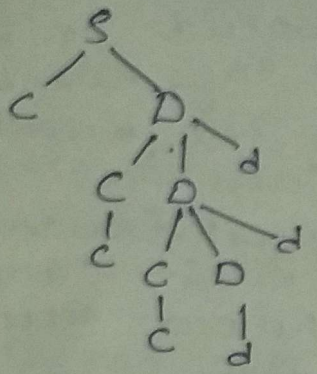
Any④ strings in L^* → (I) babbabbbabb

(II) bbbbabbbbb.

(2) $S \to cD \mid cd$, $C \to cCD \mid c$, $D \to CDd \mid d$ and the string is "ccccddd"

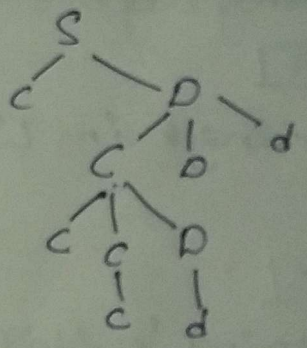
L.M.D

$S \to cD$
$\to cCDd \quad (D \to CDd)$
$\to ccDd \quad (C \to c)$
$\to cCCDdd \quad (D \to CDd)$
$\to ccCDdd \quad (C \to c)$
$\to cccDdd \quad (C \to c)$
$\to ccccddd \quad (D \to d)$



R.M.D

$S \to cD$
$\to cCDd \quad (D \to CDd)$
$\to cCdd \quad (D \to d)$
$\to ccCDdd \quad (C \to cCD)$
$\to ccCddd \quad (D \to d)$
$\to cccddd \quad (C \to c)$



left most Derivation & Right most derivation both are different so grammer is ambiguous.