Notice that in state table, for 0 input, output states are A or D. And for input 1, output states are B, C or E.

∴ we divide the states in an appropriate manner.

A, D ⇒ even number

B, C, E ⇒ odd number

A ⇒ 000

B ⇒ 001

C ⇒ 011

D ⇒ 100

E ⇒ 101

- Rewrite the state ~~it~~ table using the new names.

a) Convert into GNF

1) S → A0

A → 0B

B → 0A | 1

b) Remove left recursion

1) E → E+T | T

T → T*F | F

F → id | (E)

2) S → Aa | b

A → Ac | Sd | ε

c) Give CFG for $(a+b)^*cc(a+b)^*$

d) Give CFG that generates all possible positive even integer upto 998.

e) Design or draw a state table for a sequence detector that detects the string 1001.

Solution:

c) $S \to ABA$

$A \to a \mid b \mid Aa \mid Ab \mid \varepsilon$

$B \to cc$

~~$S \to aA \mid bS$~~

$S \to ABA$

$A \to a \mid b \mid Aa \mid Ab \mid \varepsilon$

$S \to AccA$

$A \to aA \mid bA \mid \varepsilon$

↓ Removing Null States (Not Required for this question)

$S \to AccA \mid ccA \mid Acc \mid cc$

$A \to aA \mid bA \mid a \mid b$

d) $S \to AAC$

$A \to 0 \mid 1 \mid \ldots\ldots \mid 9$

$C \to 0 \mid 2 \mid 4 \mid 6 \mid 8$

b) 2) $S \to Aa \mid b$

$A \to Ac \mid Sa \mid \varepsilon$

↓ Remove ε

$S \to Aa \mid b \mid Aa$

$A \to Ac \mid Sa \mid c$

$S \to Aa \mid b \mid Aa$

$A \to SaB \mid cB$

$B \to cB \mid \varepsilon$

Indirect Left Recursion

$S \to Aa$

$S \to SaB$ — left recursion

$A \to A\beta_1 \mid \beta_2 \mid \beta_3$

$A \to \beta_2 B \mid \beta_3 B$

$B \to \beta_1 B \mid \varepsilon$

$A \to A\beta_0 \mid A\beta_1 \mid \beta_2 \mid \beta_3$

$A \to \beta_2 B \mid \beta_3 B$

$B \to \beta_1 B \mid \beta_0 B \mid \varepsilon$

↓ Removal of ε Production

$A \to \beta_2 B \mid \beta_3 B \mid \beta_2 \mid \beta_3$

$B \to \beta_1 B \mid \beta_0 B \mid \beta_1 \mid \beta_0$

$S \rightarrow SaBa \mid b \mid a$

$A \rightarrow SaB \mid cB$

$B \rightarrow cB \mid \varepsilon$

Removing left recursion from S,

$S \rightarrow bC \mid aC$

$C \rightarrow aBaC \mid aBa$ [∵ Removing ε production]

$A \rightarrow SaB \mid cB$

$B \rightarrow cB \mid \varepsilon$ (Ans)

a) $S \rightarrow AO$

$A \rightarrow OB$

$B \rightarrow OA \mid 1$

$A_1 \rightarrow A_2 0$

$A_2 \rightarrow 0A_3$

$A_3 \rightarrow 0A_2 \mid 1$

$S \rightarrow AC$

$C \rightarrow 0$

$A \rightarrow CB$

$B \rightarrow CA$

$B \rightarrow 1$

} CNF

i) $A_1 \rightarrow A_2A_3$

ii) $A_3 \rightarrow 0$

iii) $A_2 \rightarrow A_3A_4$

iv) $A_4 \rightarrow A_3A_2$

v) $A_4 \rightarrow 1$

} Rename

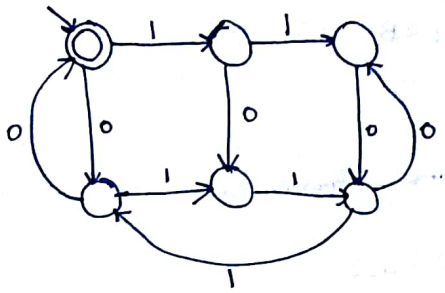
iii) $A_2 \rightarrow 0A_4$ [∵ $A_3 \rightarrow 0$]

iv) $A_4 \rightarrow 0A_2$ [∵ $A_3 \rightarrow 0$]

i) $A_1 \rightarrow 0A_4A_3$ [∵ $A_2 \rightarrow 0A_4$]

} GNF (Ans

1.



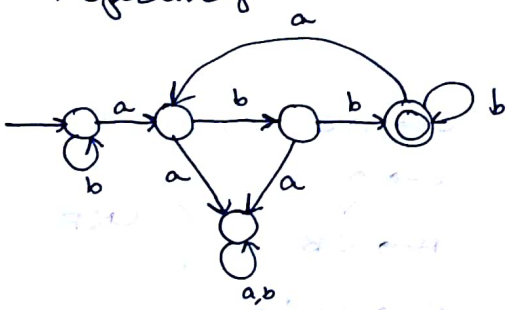
a) Number of 1's and 0's are divisible by 3 and 2 respectively.

b) Even and odd no. of 0's and 1's

c) Odd and even no. of 1's and 0's

d) Number of 1's and 0's are divisible by 2 or 3 respectively.

2.



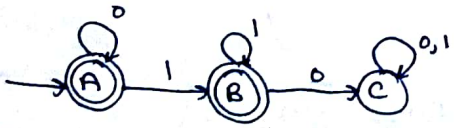
a) every a is followed by exactly two b's

b) " " " " " atleast " b's

c) does not contain substring 'abb'

d) " " " 'aaa' as substring

3. Write Regular expression for —



$A = 0^*$

$B = A1 + B1 = 0^*1 + B1 = 0^*11^*$

~~$C = B0 + C0 + C1$~~

$\therefore \quad A + B$

$= 0^* + 0^*11^*$ (Ans)

4. Construct right linear grammer for the following —

	0	1
→ (A)	B	A
(B)	C	A
C	C	D

$A \to 0B \mid 1A \mid \lambda$

$B \to 0C \mid 1A$

$C \to 0C \mid 1D$ (Ans)

5. Which of the following is equivalent regular expression

$(0+1)^*10(0+1)^* + (0+1)^*11(1+0)^*$

a) $(0+1)^*(10+11)(0+1)^*$

b) $(0+1)^*(10+11)0$

c) $(0+1)^*(10+11)$

d) none of these

$$(0+1)^*10(0+1)^* + (0+1)^*11(1+0)^*$$

$$= (0+1)^*\left[10(0+1)^* + 11(1+0)^*\right]$$

$$= (0+1)^*(10+11)(1+0)^*$$

6. $S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$

which of the following cannot be generated?

a) aaaa c) abba

b) baba d) babaaabab

7. $L_1 = \{a^n b^n c^m \mid n, m \geq 0\}$

$L_2 = \{a^n b^m c^m \mid n, m \geq 0\}$

which is false?

a) $L_1 \cap L_2$ is a CFL.

b) L_1 and L_2 both are CFL.

c) $L_1 \cup L_2$ is a CFL.

d) $L_1 \cap L_2$ is a context sensitive language.

10
0,

$$L_1 \cap L_2 = a^n b^n c^n, n > 0$$

↓

Cannot be implemented using 1 stack.

∴ it cannot be a CFL.

8. i) $L_1 = \{ 0^{n+m} 1^n 0^m \mid n, m > 0 \}$ → CFL

ii) $L_2 = \{ 0^{n+m} 1^{n+m} 0^n \mid n, m > 0 \}$ → Not a CFL

iii) $L_3 = \{ 0^{n+m} 1^{m+n} 0^{m+n} \mid n, m > 0 \}$ → Not a CFL

Which are CFL and which are not?

ii) One counter for 0^{n+m} and 1^{n+m}, and another counter for 1^{n+m} and 0^n.

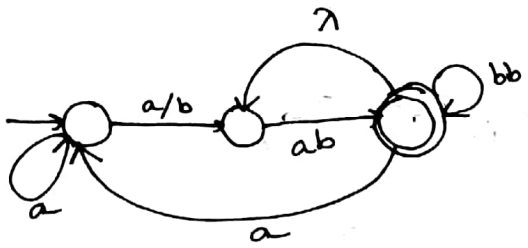
But for CFL, only one counter can be used.

∴ not a CFL (Ans)

- $0^n 1^n 2^m 3^n 4^n$ → Not a CFL

$0^n 1^n 2^m 3^p 4^p$ → CFL

9.



Language is —

a) $a^*(a+b)(\lambda + ab)(bb)^*$

b) $a^*(a+b)ab(bb + ab + aa^*(a+b)ab)^*$

c) $a^*(a+b)ab(bb)^*$

d) $a^*(a+b)ab((bb)^* + a(a+b)ab(bb)^*)$

10. $S \to aSAb \mid \varepsilon$

$A \to bA \mid \varepsilon$

The language is

a) $((a+b)^* b)^*$

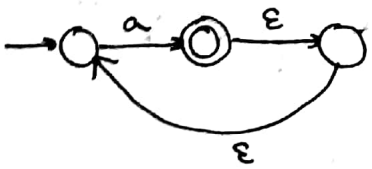
b) $\{a^m b^n \mid m \leq n\}$

c) $\{a^m b^n \mid m = n\}$

d) $a^* b^*$

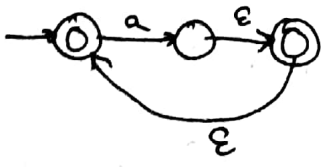
aSAb
a aSb
aSAb

11. Write the complement of the language accepted by the automata



a) ϕ b) $\{\varepsilon\}$ c) a^* d) $\{a, \varepsilon\}$

Complement —



∴ c) a^* (Ans)

12. $L = \{ab, aa, baa\}$. Which of the following is not in L.

a) abaabaaabaa b) aaaabaaaa

c) baaaadbaaaab d) baaaaaabaa

13. $(0+1)^* 0 (0+1)^* 0 (0+1)^*$. What is the language?

The string which produces at least two zeros (Ans)

14. $0^*(10^*)^*$ denotes the same as

a) $(1^*0)^*1^*$ ✓

b) $0+(0+10)^*$ ✗

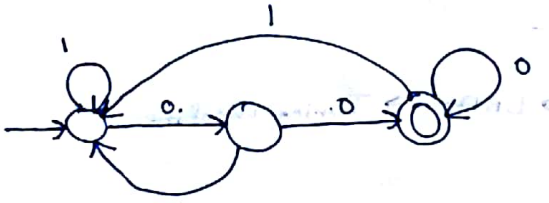
c) $(0+1)^*$ ✗

d) none

$0^*(10^*)^*$

$= a^*b^*$

15.



Find the Language ~~(Choose the bo~~

a) begins with either 0 or 1

b) ~~ends with either~~ ends with 0

c) ends with 00 ✗

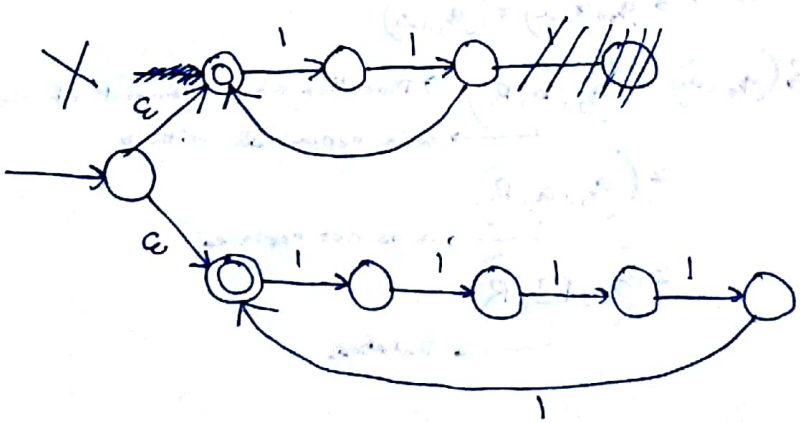
d) contain 00

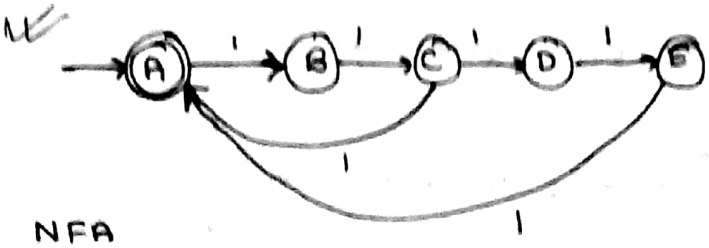
All are correct here. But option (c) is most restrictive, such that all others are valid too.

16. Draw the DFA for $(111+11111)^*$ what is the minimum number of states?

3, 5, 6, 8, 9, 10, 11, 12, ...

↑ acceptable no. of 1's





NFA

5 States

Convert to DFA.

Turing Machine

FA → PDA → LBA → Turing Machine

FA: Regular; PDA: Context-Free

- $a^n b^n$ → Not Regular, but PDA
- Whatever be computable, that can be implemented with Turing Machine.

Turing Machine is a 7 ~~tuple~~ tuple —

$$M = (Q, \Sigma, \Gamma, q_0, \delta, \square, F)$$

- Q → Set of States
- Σ → Alphabet Set
- Γ → Tape Symbol
- q_0 → Start State
- δ → transition function
- $\square$ → blank space
- F → Set of Final States

FA → $\delta(q_0, a) = q_1$

PDA → $\delta(q_0, z, a) = (q_1, z)$

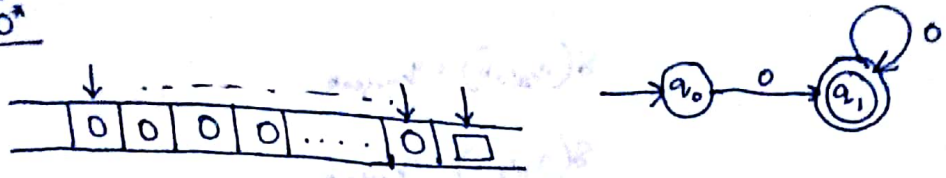
Turing Machine → $\delta(q_0, a) = (q_1, b, R)$ → Direction of Movement of the R/W head; a is replaced with b

$= (q_1, a, R)$ → a is not replaced

$= (q_1, \square, R)$ → Deleted

Example:

00^*



$Q =$

$\Sigma = \{0, 1\}$

$\Gamma = \{0, 1, \square\}$

q_0

$\delta(q_0, 0) = (q_1, \square, R)$ → Changing state to q_1 is a must. Otherwise A will also be acceptable.

$\delta(q_1, 0) = (q_1, \square, R)$ $\delta(q_1, \square) = q_{accept}$

$\delta(q_0, 1) = q_{reject}$

$\delta(q_1, 1) = q_{reject}$

- This could also have been done using a FA.

Polynomial detection

$\delta(q_0, a) = (q_a, \square, R)$

$\delta(q_0, b) = (q_b, \square, R)$

$\delta(q_0, \square) = q_{accept}$

a b b a □

□ b b a □

□ b b □

□ □ b □

□ □ □ □

$\delta(q_a, a) = (q_a, a, R)$

$\delta(q_a, b) = (q_a, b, R)$

$\delta(q_a, \square) = (q_{a'}, \square, L)$

$\delta(q_{a'}, a) = (q_2, \square, L)$

$\delta(q_b, b) = (q_b, b, R)$

$\delta(q_b, a) = (q_b, a, R)$

$\delta(q_b, \square) = (q_{b'}, \square, L)$

$\delta(q_{b'}, b) = (q_2, \square, L)$

$\delta(q_{a'}, b) = q_{reject}$

$\delta(q_{b'}, a) = q_{reject}$

$\delta(q_2, a) = (q_2, a, L)$

$\delta(q_2, b) = (q_2, b, L)$

$\delta(q_2, \square) = (q_0, \square, R)$

$\delta(q_{a'}, \square) = q_{accept}$

$\delta(q_{b'}, \square) = q_{accept}$

} Handling strings with odd length