01/10/2020

CS303 - Formal Languages and Automata Theory

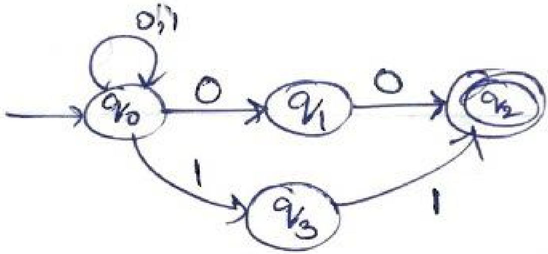
Mid-Semester Examination

Name of Student: M. Maheeth Reddy

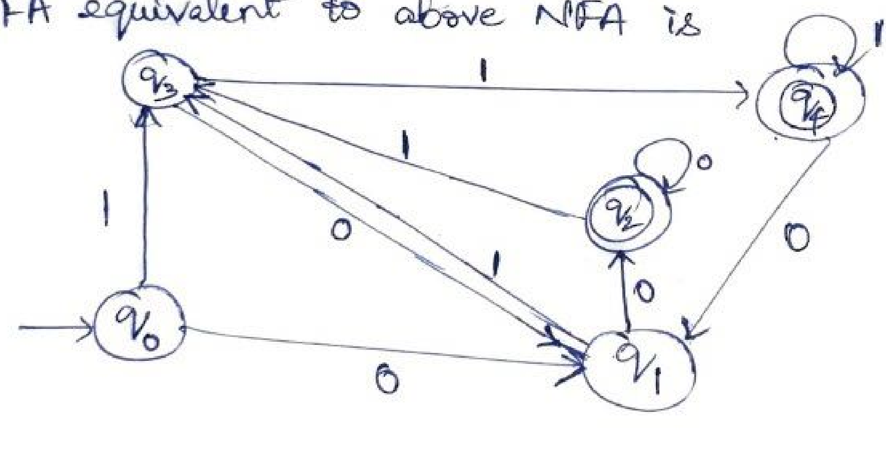
Roll No.: 1801CS31

Ans.1:

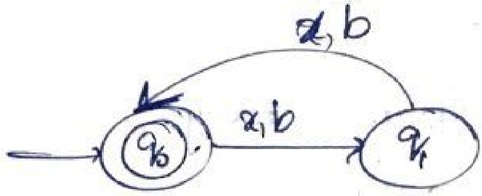
Required NFA is



DFA equivalent to above NFA is



Ans 2: Required DFA is

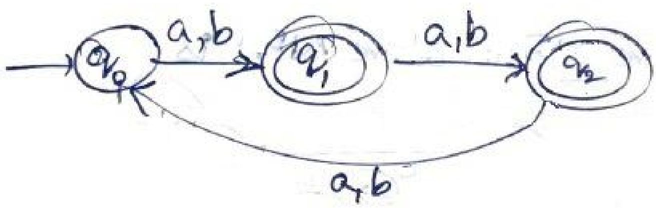


States of DFA:

① q_0 accepts strings which have even length (i.e., length is divisible by 2)

② Strings with odd length have q_1 as final ~~state~~. (non-accepting)

Ans 3: Required DFA is,



For strings $w \in \{a,b\}^*$,

if $|w| \bmod 3 = 0$, final state is q_0

$|w| \bmod 3 = 1$, final state is q_1

$|w| \bmod 3 = 2$, final state is q_2

Hence, q_1 and q_2 are accepting states

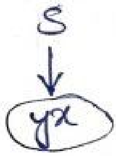
Ans 4 :

$G = \{S \to SS,\ S \to xy,\ S \to yx,\ S \to \lambda\}$ is a context-free grammar

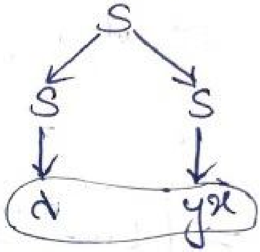
G is ambiguous

G produces 2 derivation trees for string yx.

Tree 1



Tree 2



G doesn't produce all strings with equal no. of x's & y's

G doesn't produce the string $yyxx$. The no. of x's & y's is equal to 2.

It only produces $xyxy$, $xyyx$, $yxxy$, $yxyx$.

G can be accepted by deterministic PDA

Language generated by G is $\{xy, yx, xyxy, xyyx, yxxy, yxyx, xyxyxy, xyxyyx, \ldots\}$ which can be represented by the regular expression $(xy + yx)^*$

∴ G is regular and regular languages can be described by deterministic PDA

Ans 5:

$L1 = \{x^m y^n \mid m,n \geq 0\}$

$L2 = \{x^n y^n \mid n \geq 0\}$

$L3 = \{x^n y^n z^n \mid n \geq 0\}$

L1 is a regular language. It is represented by the regular expression x^*y^*. Therefore L1 is context free.

L2 is a context-free language. It is generated by this context free grammar,

$$G = (V, T, S, P) \quad T = \{a, b\} \quad V = \{a, b, S\}$$

$$P = \{S \rightarrow aSb, S \rightarrow \lambda\}$$

Since, L1 and L2 are context free, Push Down Automata (PDA) can be used to recognize L1 and L2.

The grammar to generate L3 is given as

$$\left.\begin{aligned} S &\rightarrow aSBC \mid aBC \mid \epsilon \\ CB &\rightarrow BC \\ aB &\rightarrow ab \\ bB &\rightarrow bb \\ bC &\rightarrow bc \\ cC &\rightarrow cc \end{aligned}\right\}$$

The left hand sides of all productions are not single terminals. Therefore this is not context free.

Hence, all three languages are not context free.

6.

G1 is $\{ S \to XSY,\ X \to aXS | a | \lambda,\ Y \to SbS | X | bb \}$

Remove Null productions ($\underline{X \to \lambda}$)

$S \to XSY \mid SY$

$X \to aXS | aS | a$

$Y \to SbS | X | bb | \lambda$

Remove $\underline{Y \to \lambda}$

$S \to XSY | SY | XS | S$

$X \to aXS | aS | a$

$Y \to SbS | X | bb$

Remove unit productions ($S \to S,\ Y \to X$)

$S \to XSY | SY | XS | XSX | SX$

$X \to aXS | aS | a$

$Y \to SbS | bb$

Introduce new variables for terminals

$S \to XSY | SY | XS | XSX | SX$

$X \to T_a XS | T_a S | a$

$Y \to ST_bS | T_bT_b$ $\quad T_a \to a \quad T_b \to b$

Break productions with 3 non-terminals on RHS.

$S \to V_1Y \mid SY \mid XS \mid V_1X \mid SX$

$V_1 \to XS$

$X \to V_2S \mid T_aS \mid a$

$V_2 \to T_aX$

$Y \to V_3S \mid T_bT_b$

$V_3 \to ST_b$

$T_a \to a$

$T_b \to b$

This is CNF of G!

Ans 7

Difference between Ambiguous and Non-Ambiguous Grammars

Criteria	Ambiguous Grammar	Non-Ambiguous Grammar
Definition	A context free grammar for which, there exists more than one parse tree/ derivation tree for atleast one string generated by it.	A context free grammar for which there exists only one parse tree for all strings generated.
No. of leftmost/ rightmost derivations	There are more than one possible derivations	Only one derivation is possible.
Parse trees generated by leftmost/rightmost derivation	They are different	They are same.
Length of Parse Trees	They are comparitively lesser	Comparitively larger.
Speed of Derivation of Tree	Comparitively faster	Comparitively slower
No. of non-terminals	Lesser in comparision	Greater in comparision

Example

Ambiguous Grammar

$E \to E + E$

$E \to E * E$

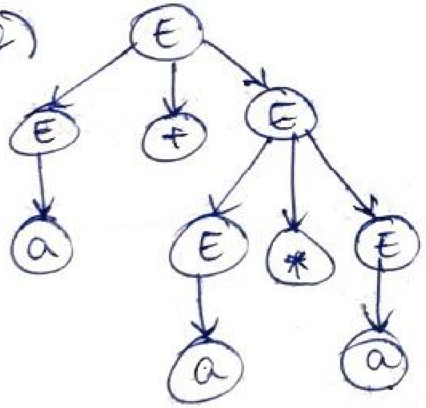
$E \to (E)$

$E \to a$

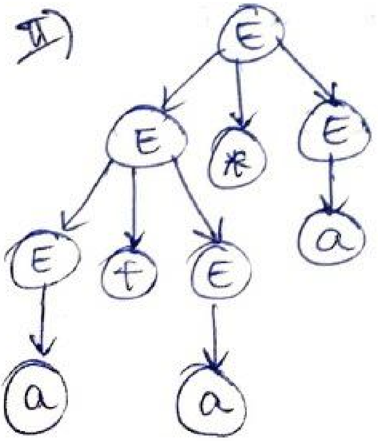
No-of non-terminals = 1

2 derivation trees for $a + a * a$

I)



II)



Non-Ambiguous Equivalent Grammar

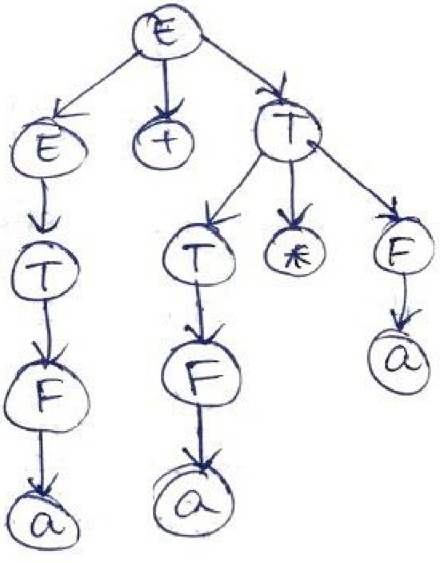
$E \to E + T \mid T$

$T \to T * F \mid F$

$F \to (E) \mid a$

No-of non-terminals = 3

Only one derivation tree for $a + a * a$



8. G_1 is $\{S \to XZ \mid WW,\ W \to b \mid SW,$
$X \to b,\ Z \to a\}$

Substitute $S \to XZ \mid WW$ in $W \to SW$

$S \to XZ \mid WW$

$W \to bXZW \mid WWW$

$X \to b$

$Z \to a$

Substitute $X \to b$ in $S \to XZ$ and $W \to XZW$

$S \to bZ \mid WW$

$W \to b \mid bZW \mid WWW$

$X \to b$

$Z \to a$

Remove $W \to WWW$

$S \to bZ \mid WW$

$W \to bT \mid bZWT$

$T \to WWT \mid \lambda$

$X \to b$

$Z \to a$

Remove $T \rightarrow \epsilon$,

$S \rightarrow bZ \mid WW$

$W \rightarrow bT \mid bZWT \mid b \mid bZW$

$T \rightarrow WWT \mid WW$

$X \rightarrow b$

$Z \rightarrow a$

Substitute $W \rightarrow bT \mid bZWT \mid b \mid bZW$ in $S \rightarrow WW$

$S \rightarrow bZ \mid bTW \mid bZWTW \mid bW \mid bZWW$

$W \rightarrow bT \mid bZWT \mid b \mid bZW$

$T \rightarrow WWT \mid WW$

$X \rightarrow b$

$Z \rightarrow a$

Substitute $W \rightarrow bT \mid bZWT \mid b \mid bZW$ in $T \rightarrow WW$

$S \rightarrow bZ \mid bTW \mid bZWTW \mid bW \mid bZWW$

$W \rightarrow bT \mid bZWT \mid b \mid bZW$

$T \rightarrow WWT$

$T \rightarrow bTW \mid bZWTW \mid bW \mid bZWW$

$X \rightarrow b$

$Z \rightarrow a$

Substitute $W \rightarrow bT \mid bZWT \mid b \mid bZW$ in $T \rightarrow WWT$

$S \rightarrow bZ \mid bTW \mid bZWTW \mid bW \mid bZWW$

$B \rightarrow bT \mid bZWT \mid b \mid bZW$

$T \rightarrow bTWT \mid bZWTWT \mid bWT \mid bZWWT$

$T \rightarrow bTW \mid bZWTW \mid bW \mid bZWW$

$X \rightarrow b$

$Z \rightarrow a$

This is GNF of G1

Ans 9:

Given NFA is,

Let the name of equivalent DFA is M

its transition function is δ

transition function of NFA is δ^*

Initial State of DFA is $\{q_0\}$

$$\boxed{\begin{array}{l} \delta(\{q_0\}, a) = \delta^*(q_0, a) = \{q_0, q_1\} \\ \delta(\{q_0\}, b) = \delta^*(q_0, b) = \{q_0\} \end{array}}$$

Add the transitions to DFA M.

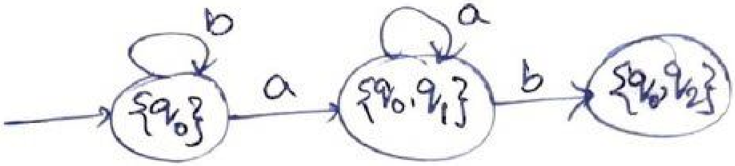
$$\delta(\{q_0, q_1\}, a) = \delta^*(q_0, a) \cup \delta^*(q_1, a) = \{q_0, q_1\} \cup \phi$$

$$\Rightarrow \boxed{\delta(\{q_0, q_1\}, a) = \{q_0, q_1\}}$$

$$\delta(\{q_0, q_1\}, b) = \delta^*(q_0, b) \cup \delta^*(q_1, b) = \{q_0\} \cup \{q_2\}$$

$$\Rightarrow \boxed{\delta(\{q_0, q_1\}, b) = \{q_0, q_2\}}$$

Add the transitions to DFA M.



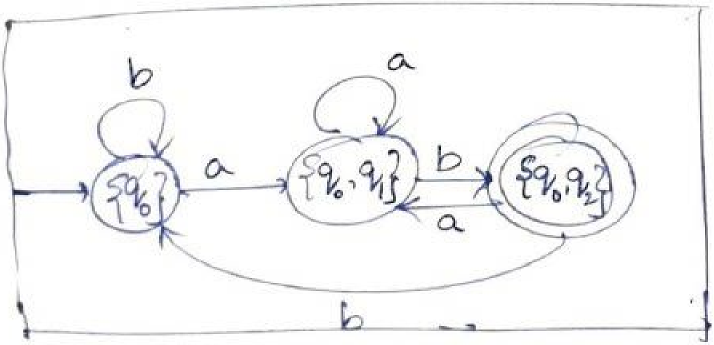
$$\delta(\{q_0, q_2\}, a) = \delta^*(q_0, a) \cup \delta^*(q_2, a)$$
$$= \{q_0, q_1\} \cup \phi$$

$$\boxed{\delta(\{q_0, q_2\}, a) = \{q_0, q_1\}}$$

$$\delta(\{q_0, q_2\}, b) = \delta^*(q_0, b) \cup \delta^*(q_2, b)$$
$$= \{q_0\} \cup \phi$$

$$\boxed{\delta(\{q_0, q_2\}, b) = \{q_0\}}$$

Add these transitions to DFA M



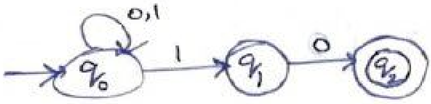
Since, no new states are added.

This is final DFA

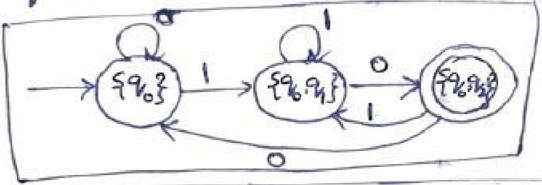
Ans 10: Answer is 3

Given regular expression is $(0+1)^* 10$

Corresponding NFA is



Equivalent DFA (obtained by converting NFA to DFA)



It has 3 states

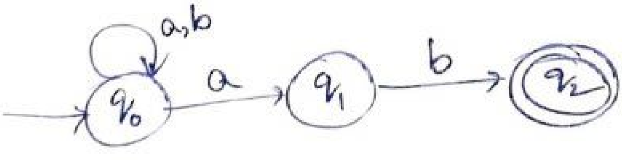
$\therefore$ $\boxed{\text{Number of states in minimal DFA corresponding to } (0+1)^* 10 \text{ is } 3}$

Ans 11:

In the given set of strings on {a,b}, each string has some possible sequence of a,b followed by substring ab.

Therefore, NFA must have one state to take any possible sequence of a,b as input.

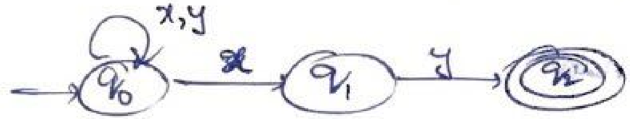
There must be two more states such that if strings ends in ab, last state is accepting.



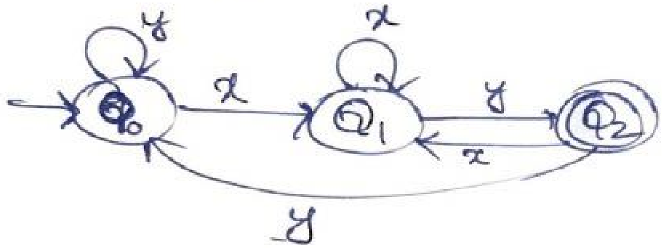
This is the required NFA

Ans 12 :

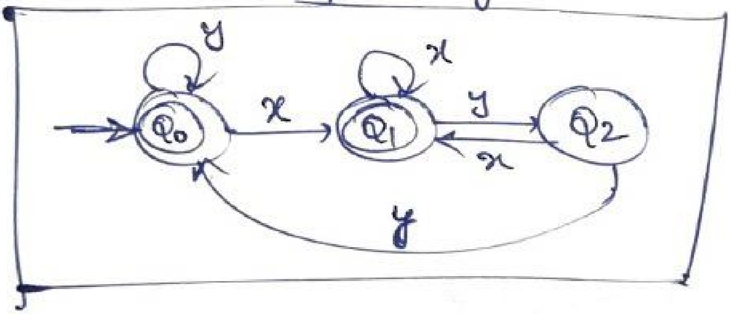
minimal NFA to accept strings ending in xy is



equivalent DFA for this is



∴ minimal NFA to accept strings not ending in xy is



This is obtained by swapping accepting states of previous DFA with its non-accepting states, Also, all DFA's are NFA's.
And this is the minimal DFA that can be obtained.

Ans B:

$$\{0^{2x} \mid x \geq 1\} = \{(00)^x \mid x \geq 1\} = \underbrace{(00)^+}_{\text{regular expression}}$$

$$\{0^y 0^x 0^{y+x} \mid y \geq 1 \text{ and } x \geq 2\} = \{0^{2(y+x)} \mid y \geq 1 \text{ and } x \geq 2\}$$

$$= \{(00)^{y+x} \mid (y+x) \geq 3\}$$

say $(y+x) = k$

$$= \{(00)^k \mid k \geq 3\}$$

$$= \underbrace{0000(00)^+}_{\text{regular expression}}$$

$\because$ Languages given in S1 and S2 have regular expressions.

$\therefore$ $\boxed{\text{Both S1 and S2 are correct.}}$

Ans 14:

Start state is q_1. Since it has a self loop over itself, the regular expression begins with a^*

For q_1 to q_2, input is b. Since there is self loop on q_2 b^* is also present in regular expression. The expression obtained so far is a^*bb^*

From q_2 to q_3, the possible input is $a(bb^*a)^*$

From q_3 to q_1, the input should be a

Again at q_1 there are self loops so we include a^* at the end

Regular exp. obtained so far is $a^*bb^*a(bb^*a)^*aa^*$

This is equivalent to $a^*(bb^*a)^+a^+$. As the automata has same initial and final states, $(a^*(bb^*a)^+a^+)^*$ is also accepted.

Observe a^* is also accepted.

∴ Regular expression obtained is

$$\boxed{\begin{aligned} &a^* + (a^*(bb^*a)^+a^+)^* \\ &= a^* + (a^*(b^+a)^+a^+)^* \end{aligned}}$$

Ans 15: Language in (C) is represented by regular exp.
In the regular expression $(0+1)^* 0 (0+1)^* 0 (0+1)^*$, the number of 0's represented by $(0+1)^*$ can be greater than or equal to zero.
Since, the expression specifies two zeroes we can conclude that in the corresponding language, each string has atleast two zeroes

Ans 16 :

$L1 = \{\phi\}$ $\quad\quad L2 = \{a\}$

L1 is empty language. Hence the concateration of L1 with any language is empty language.

$$\therefore \quad L1\,L2^* = \phi$$

Kleene closure of empty language is $\{\epsilon\}$,

$$\therefore \quad L1^* = \{\epsilon\}$$

$$\Rightarrow \boxed{L1L2^* \cup L1^* = \{\epsilon\}}$$