CS-303 Mid Semester

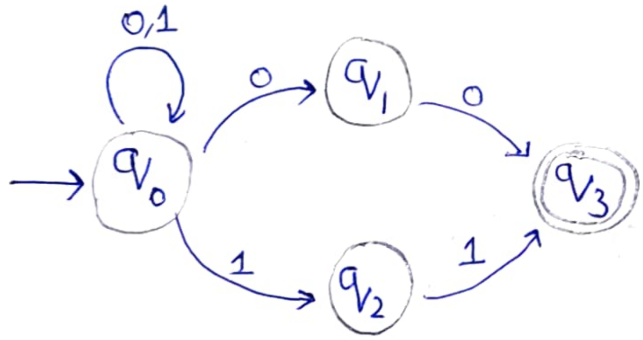
Name : P.V. Sriram

Roll No. : 1801CS37

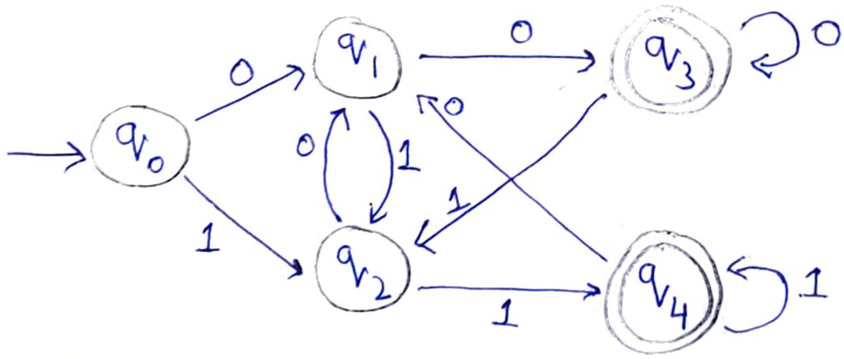
1) Required, NFA and DFA for strings ending with 00, 11

i.e) $L(R) = (0+1)^*00 + (0+1)^*11$

NFA:

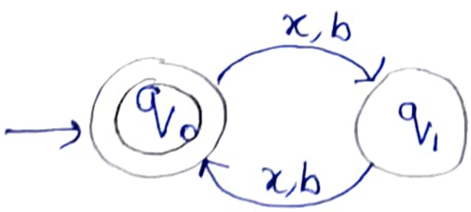


DFA:



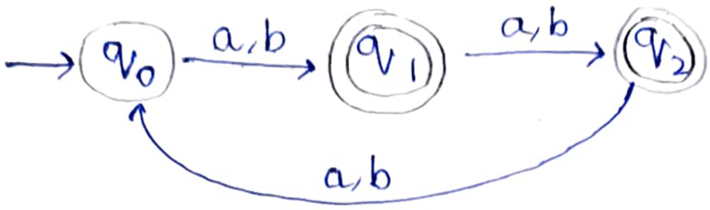
2) Required DFA which accepts strings of even length over {x, b}. i.e) $|w| \bmod 2 = 0$

DFA:



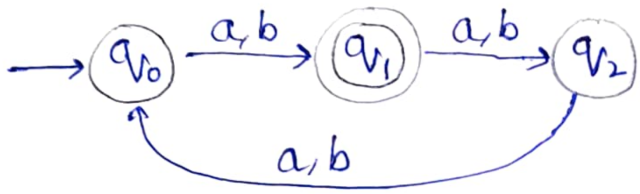
3) Required, DFA to accepts strings of length not divisible by 3

DFA:



For $|w| \bmod 3 = 1$

DFA:

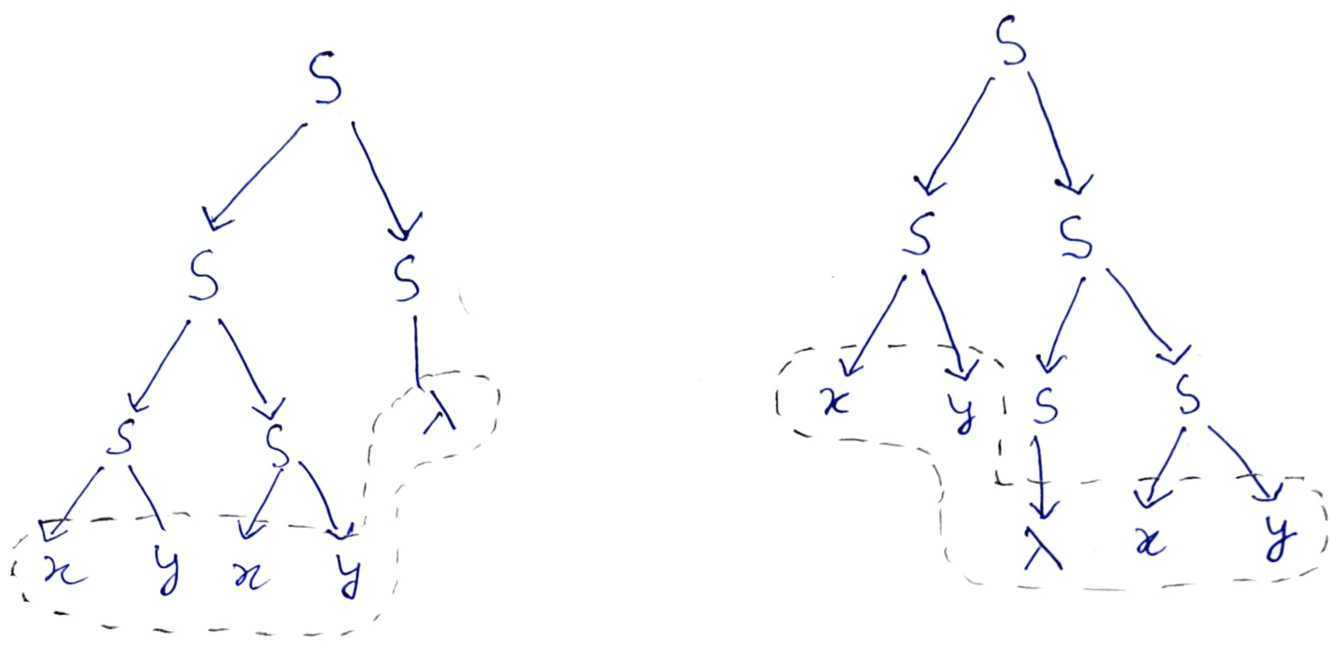


---

4) Given CFG; $G = \{S \to SS,\ S \to xy,\ S \to yx,\ S \to \lambda\}$

A) A Context free grammar G ~~has~~ is said to be ambiguous if it has more than one derivation tree for some string $w \in L(G)$. i.e) Existence of multiple left most or Right most derivation tree.

For example, consider string "xyxy"



We can see that two left most derivation tree produce $xyxy$. There fore, Ambiguous.

B) G cannot produce all the strings with equal number of x's and y's.

We can prove this using proofs by example.

Eg:- consider $w = xxxyyy$

Although this string has same number of xs, ys we cant derive it from grammar G

$\therefore$ Statement is wrong

c) $G = \{S \to SS, S \to xy, S \to yx, S \to \lambda\}$

This can be converted into

$G = \{S \to Sxy, S \to Syx, S \to \lambda\}$

This is a Left linear ~~Regular~~ Grammar and the language produced is Regular.

i.e) $L(G) = (xy + yx)^*$

All regular languages can be represented using Deterministic PDAs. Hence Proved

---

5) Given

A) $L1 = \{x^m y^n \mid m, n >= 0\}$

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

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

We know that all context-free grammars can be ~~used~~ to recognized ~~L1~~, ~~L2~~ by ~~a~~ PDAs

Therefore, if we can make a context free grammars out of L1, L2. We can prove the existance of PDA s

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

i.e) $x^* y^*$

$$G_1 = \{ S \rightarrow AB, \; A \rightarrow xA \mid \lambda, \; B \rightarrow yB \mid \lambda \}$$

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

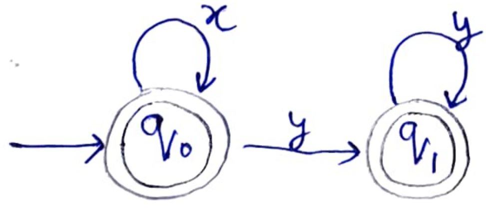
$$G_2 = \{ S \rightarrow xSy \mid \lambda \}$$

We can see both G_1, G_2 are CFGs. Therefore we can have PDAs to recognize both.

B) If we can make a DFA to recognise a language, then we can say that the language is regular.

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

DFA :



$\therefore$ L1 is regular.

c) We already proved that L1, L2 are context free

For L3,

Pumping lemma for context free grammars:

Consider L3 is a context free grammar.

If A is a context free language, then A has a pumping length 'p' such that any string 's', where $|s| \geq p$ may be divided into 5 pieces $s = uvwxy$ such that

(1) uv^iwx^iy is in A for every $i \geq 0$

(2) $|vx| > 0$

(3) $|vwx| \leq p$

Consider pumping length m
Consider string $x^m y^m z^m$

$$w = x^m y^{\frac{m}{3}} y^{\frac{m}{3}} y^{\frac{m}{3}} z^m$$

if $\forall\ i \geq 0$ $x^m (y^{\frac{m}{3}})^i y^{\frac{m}{3}} (y^{\frac{m}{3}})^i z^m$ belong to L3 then it is context free.

Consider $i = 0 \Rightarrow$ $w' = x^m y^{\frac{m}{3}} z^m \notin L3$

∴ By contradiction using pumping lemma,

We prove L3 is NOT context free

6) Required to convert CFG to CNF.

$G_1 = \{ S \to XSY$
$X \to aXS \mid a \mid \lambda$
$Y \to SbS \mid X \mid bb \}$

Step 1 : Remove Null production

$S' \to S$
$S \to XSY$
$X \to aXS \mid a \mid \lambda$
$Y \to SbS \mid X \mid bb$

$\xrightarrow[X \to \lambda]{\text{Remove}}$

$S' \to S$
$S \to XSY \mid SY$
$X \to aXS \mid a \mid aS$
$Y \to SbS \mid X \mid bb \mid \lambda$

$\downarrow$ Remove $Y \to \lambda$

$S' \to S$
$S \to XSY \mid SY \mid XS \mid S$
$X \to aXS \mid a \mid aS$
$Y \to SbS \mid X \mid bb$

$\xleftarrow[S \to S \text{ (Trivial)}]{\text{Remove}}$

$S' \to S$
$S \to XSY \mid SY \mid XS$
$X \to aXS \mid a \mid aS$
$Y \to SbS \mid X \mid bb$

Step 2: Remove Unit Production

$S' \to S$
$S \to XSY \mid SY \mid XS$
$X \to aXS \mid a \mid aS$
$Y \to SbS \mid X \mid bb$

$\xrightarrow[S' \to S]{\text{Remove}}$

$S' \to XSY \mid SY \mid XS$
$S \to XSY \mid SY \mid XS$
$X \to aXS \mid a \mid aS$
$Y \to SbS \mid X \mid bb$

Remove $Y \to X$

$S' \to XSY \mid SY \mid XS$

$S \to XSY \mid SY \mid XS$

$X \to aXS \mid a \mid aS$

$Y \to SbS \mid bb \mid aXS \mid a \mid aS$

Step-3 : Add Extra variables for normalisisation

$S' \to XV_2 \mid SY \mid XS$

$S \to XV_2 \mid SY \mid XS$

$X \to T_a V_1 \mid T_a S \mid a$

$Y \to SV_3 \mid bb \mid T_a V_1 \mid a \mid T_a S$

$V_1 \to XS$

$V_2 \to SY$

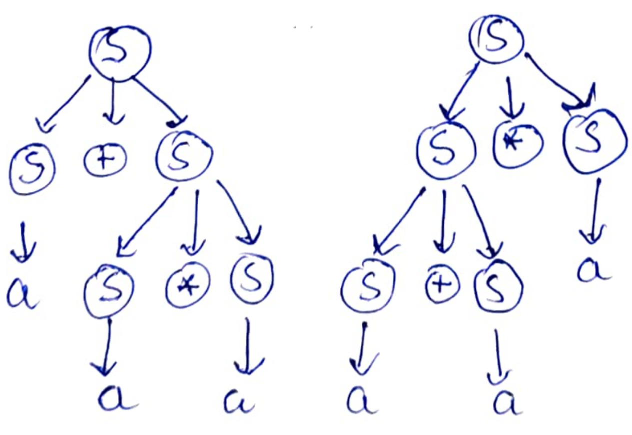
$V_3 \to T_b S$

$T_a \to a$

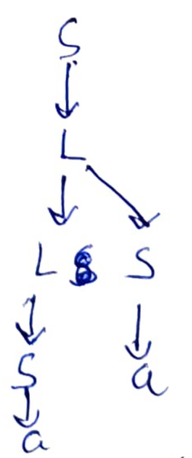
$T_b \to b$

7)

Ambiguous Grammar	Unambiguous Grammar
(i) An ambiguous grammar more than one derivation tree for a single string (atleast one string)	An unambiguous grammar has exactly one ~~string~~ derivation tree for every string
(ii) Single string has more than one possible Parse Tree Derivation tree Left most derivation tree Right most derivation tree Syntax tree	Single string has exactly one Parse Tree Derivation tree Leftmost derivation tree Right most derivation tree Syntax tree
(iii) Length of parse tree is less	Length of parse tree is large.
(iv) Contains less number of non-Terminals	Contains more number of non-Terminals.
(v) Left most and right most derivations are not same	Left most and right most derivations are same.
(vi) Eg:- $S \to S+S \mid S*S \mid a \mid b$	Eg:- $S \to L \mid a$, $L \to LS \mid S$



Ambiguous as $a+a*a$ has 2 Derivation tree



Unambiguous ~~bas~~ aa has 1 DT

8) Required, Converting CFG to GNF

$G_1 : \{ S \to XZ \mid WW,$

$W \to b \mid SW,$ (Already in CNF)

$X \to b,$

$Z \to a \}$

Step-1 : Use A_i variables ~~for~~ Existent variables

ie) $A_1 = S$; $A_2 = X$, $A_3 = Z$, $A_4 = W$

$\Rightarrow$

$A_1 \to A_2 A_3 \mid A_4 A_4$

$A_4 \to b \mid A_1 A_4$

$A_2 \to b$

$A_3 \to a$

Step-2 : for all $A_i \to A_j x$; make sure $i \le j$

$A_1 \to A_2 A_3 \mid A_4 A_4$

$A_4 \to b \mid A_1 A_4$

$A_2 \to b$

$A_3 \to a$

$\xrightarrow[A_4 \to A_1 A_4]{\text{Replace}}$

$A_1 \to A_2 A_3 \mid A_4 A_4$

$A_4 \to b \mid A_2 A_3 A_4 \mid A_4 A_4 A_4$

$A_2 \to b$

$A_3 \to a$

Replace $A_4 \to A_2 A_3 A_4$

$A_1 \to A_2 A_3 \mid A_4 A_4$

$A_4 \to b \mid b A_3 A_4 \mid A_4 A_4 A_4$

$A_2 \to b$

$A_3 \to a$

$A_1 \to A_2 A_3 \mid A_4 A_4$

$A_4 \to b \mid bA_3A_4 \mid A_4A_4A_4$ $\qquad (A_4 \to A_4 A_4 A_4)$ is a left recursion

$A_2 \to b$

$A_3 \to a$

To remove a left recursion, we need to add a new variable

$Z' \to A_4 A_4 \mid A_4 A_4 Z'$

$\Rightarrow A_1 \to A_2 A_3 \mid A_4 A_4$

$A_4 \to b \mid bA_3A_4 \mid bZ' \mid bA_3A_4Z'$

$Z \to A_4 A_4 \mid A_4 A_4 Z$

$A_2 \to b$

$A_3 \to a$

Replace $A_1 \to A_2 A_3$ and $A_1 \to A_4 A_4$

$A_1 \to bA_3 \mid bA_4 \mid bA_3A_4A_4 \mid bZ'A_4 \mid b$
$bA_3A_4Z'A_4$

$A_4 \to b \mid bA_3A_4 \mid bZ' \mid bA_3A_4Z'$

$Z' \to A_4A_4 \mid A_4A_4Z'$

$A_2 \to b$

$A_3 \to a$

Replace $Z \to A_4A_4,\ Z \to A_4A_4Z$

$A_1 \to bA_3 \mid bA_4 \mid bA_3A_4A_4 \mid bZA_4 \mid bA_3A_4Z'A_4$

$A_4 \to b \mid bA_3A_4 \mid bZ' \mid bA_3A_4Z'$

$Z' \to bA_4 \mid bA_3A_4A_4 \mid bZ'A_4 \mid bA_3A_4Z'A_4$

$Z' \to bA_4Z' \mid bA_3A_4A_4Z' \mid bZ'A_4Z' \mid bA_3A_4Z'A_4A_4$

$A_2 \to b$

$A_3 \to a$

Replacing with initial variables, we get

$S \to bZ \mid bW \mid bZWW \mid bZ'W \mid bZWZ'W$

$W \to b \mid bZW \mid bZ' \mid bZWZ'$

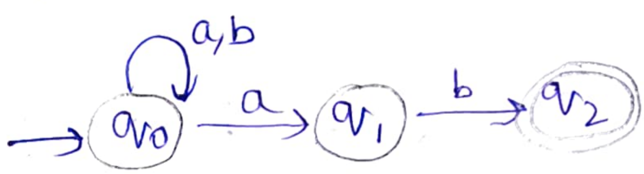
$Z' \to bW \mid bZWW \mid bZ'W \mid bZWZ'W$

$Z' \to bZ'WZ' \mid bZWZ'WZ' \mid bWZ' \mid bZWWZ'$

$X \to b$

$Z \to a$

9) Given, NFA :



Consider, First state q_0

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

$\delta'(\{q_0\}, b) = \{q_0\}$

consider state $\{q_0, q_1\}$

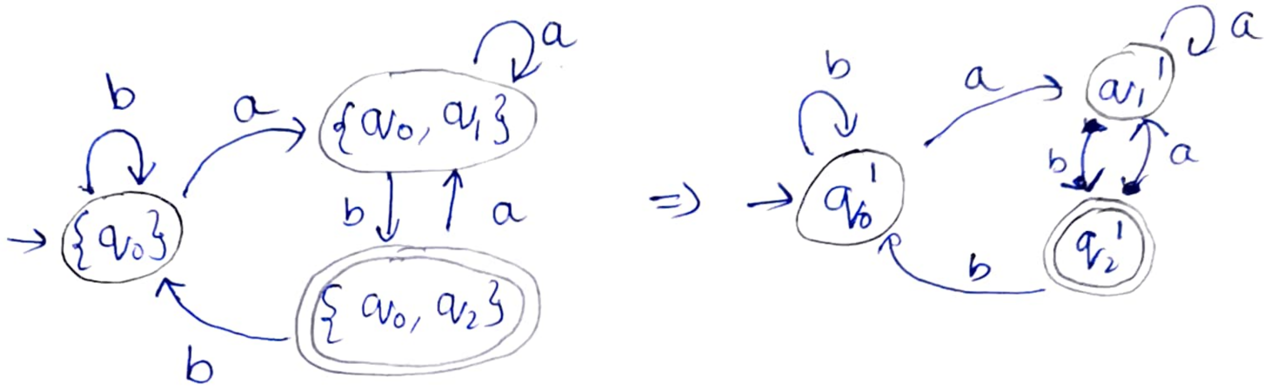
$\delta'(\{q_0, q_1\}, a) = \{q_0, q_1\}$

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

Consider state $\{q_0, q_2\}$

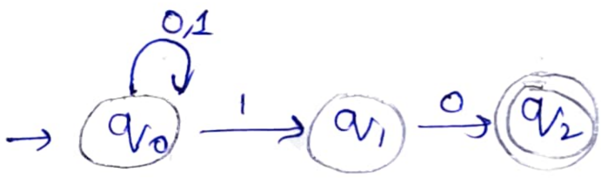
$\delta'(\{q_0, q_2\}, a) = \{q_0, q_1\}$

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



10) Given, expression $(0+1)^*(10)$

NFA:



Conversion to DFA:

Consider, First state q_0

$$\delta'(\{q_0\}, 1) = \{q_0, q_1\}$$

$$\delta'(\{q_0\}, 0) = \{q_0\}$$

Consider state $\{q_0, q_1\}$

$$\delta'(\{q_0, q_1\}, 0) = \{q_0, q_2\}$$

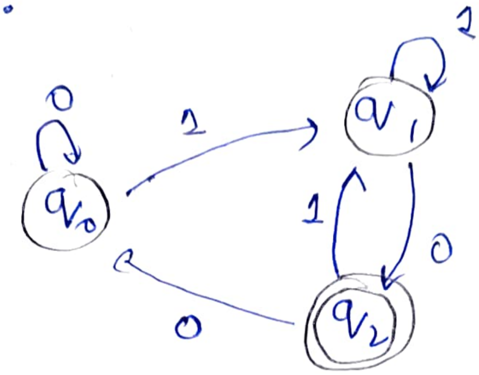
$$\delta'(\{q_0, q_1\}, 1) = \{q_0, q_1\}$$

Consider state $\{q_0, q_2\}$

$$\delta'(\{q_0, q_2\}, 0) = \{q_0\}$$

$$\delta'(\{q_0, q_2\}, 1) = \{q_0, q_1\}$$

DFA :



Minimization

q_1		
q_2	ε	ε
	q_0	q_1

	0	1
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_0	q_1

DISTINCT $(q_0, q_1) = (q_0, q_2)$

$\therefore$ DISTINCT $(q_0, q_1) = 0$

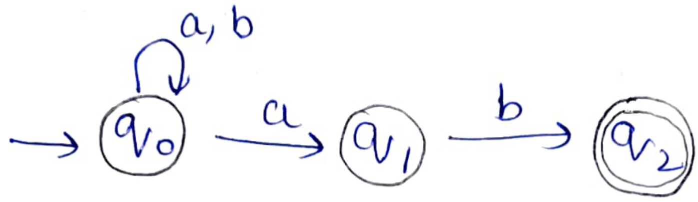
q_1	0	
q_2	ε	ε
	q_0	q_1

$\therefore$ No two states are equivalent

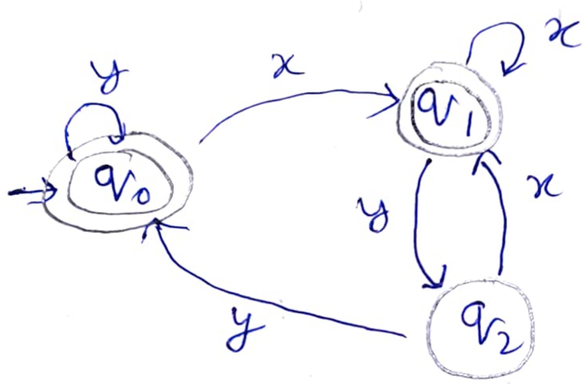
$\therefore$ Minimal DFA

$\boxed{\text{States} = 3}$

11) Required, NFA to accept strings ending with 'ab'.



12) Required, NFA to accept strings not ending in 'xy'.

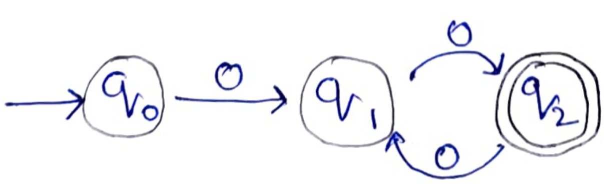


This is done using swap of final, non-final states from DFA of previous question.

13) Given, $S1 = \{0^{2x} \mid x >= 1\}$

we can express this language in the following DFA

DFA:

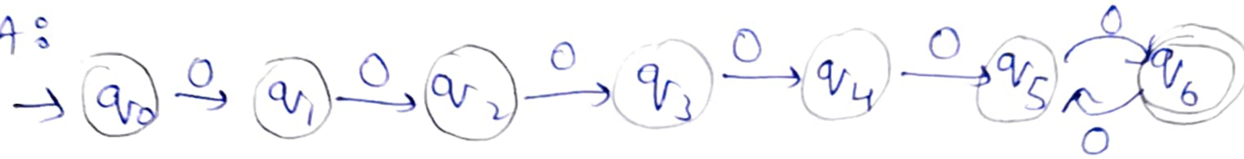


$\therefore$ S1 is regular

$S2 : \{ 0^{y} 0^{x} 0^{(y+x)} \mid y >= 1 \text{ and } x >= 2 \}$

$= \{ 0^{2(x+y)} \mid y \geq 1 \text{ and } x \geq 2 \} = \{ 0^{2k} \mid k \geq 3 \}$

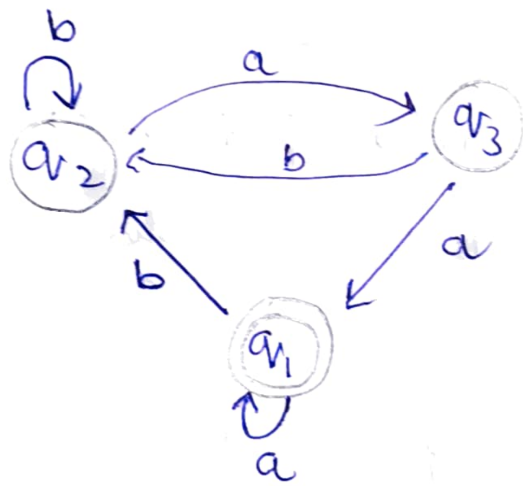
NFA:



$\therefore$ S2 is regular aswell

---

14) Given automata:



$q_1 = \cancel{\lambda} + q_1 a + q_3 a$ — ①

$q_2 = q_1 b + q_2 b + q_3 b$ — ②

$q_3 = q_2 a$ — ③

Substituting $q_3 = q_2 a$ in ①, ②, we get

$q_1 = \cancel{\lambda} + q_2 aa + q_1 a$; $q_2 = q_1 b + q_2 b + q_2 ab$

we use the property, $A = B + AC$

$\Rightarrow A = BC^*$

(Left recursion)

$$q_1 = \lambda + q_2 aa + q_1 a$$

$$\Rightarrow q_1 = (q_2 aa + \lambda) a^* \quad - (4)$$

$$\text{||}^{ly} \quad q_2 = q_1 b (b + ab)^* \quad - (5)$$

use (5) in (4)

$$q_1 = q_1 b (b+ab)^* aaa^* + a^*$$

using $A = B + AC \Rightarrow A = BC^*$ again

we get $\boxed{a^* (b(b+ab)^* aaa^*)^*}$

---

15) Given, Regular expression $(0+1)^* 0 (0+1)^* 0 (0+1)^*$

$$\underbrace{(0+1)^*}_{A} \; 0 \; \underbrace{(0+1)^*}_{B} \; 0 \; \underbrace{(0+1)^*}_{C}$$

Here, we can see that A, B, C is all possible strings of 0,1 varying from null to ∞ length.

So in the extreme case we can see that the string definitely consists of atleast 2 zeros. They may or may not be consequential and we cannot say which number is there at the end of string.

∴ $\boxed{\text{Option C is correct}}$

16) Given $L1 = \{\phi\}$ and $L2 = \{a\}$.

~~L1 E~~

Required $L1\,L2^* \cup L1^*$

$$L_1^* = \phi^* = \varepsilon$$

Therefore,

$$L_1 L_2^* \cup L_1^* = \phi \cdot (L_2)^* \cup \phi^*$$

$$= \phi \cup \{\varepsilon\}$$

{ (Because ϕ concatenated with other lang is ϕ and $\phi^* = \{\varepsilon\}$)

$$\Rightarrow \quad \phi \cup \{\varepsilon\} = \boxed{\{\varepsilon\}}$$