Roll-No. 2019101053 Name - Archit Jain

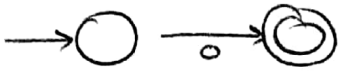
PHASE II

Q 19 Convert regular expression to NFA

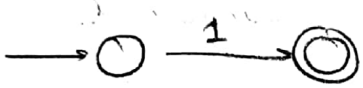
$(((00)^*(11)) \cup 01)^*$

Sol

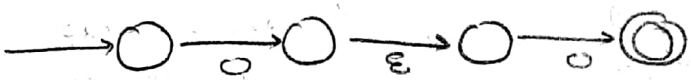
0 :



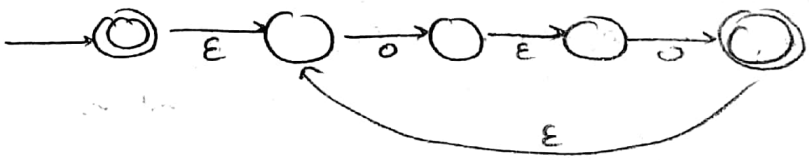
1 :



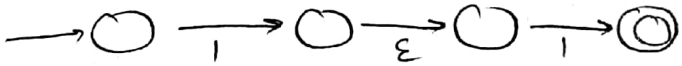
00 :



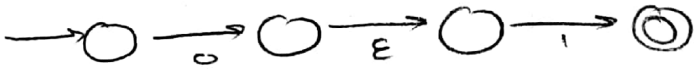
$(00)^*$:



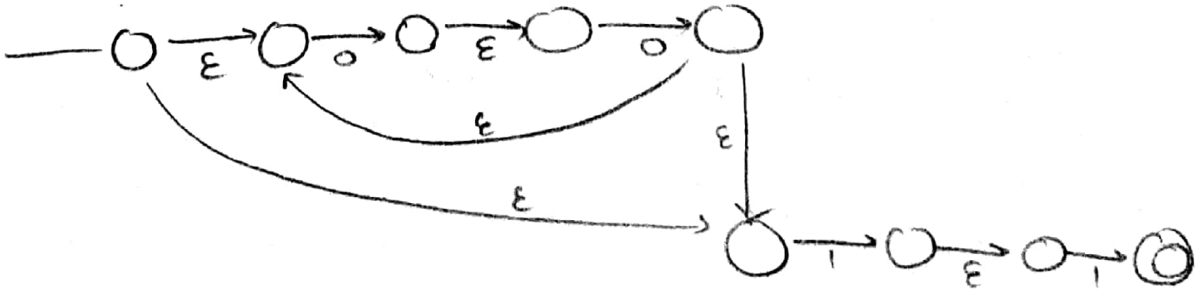
11 :



01 :

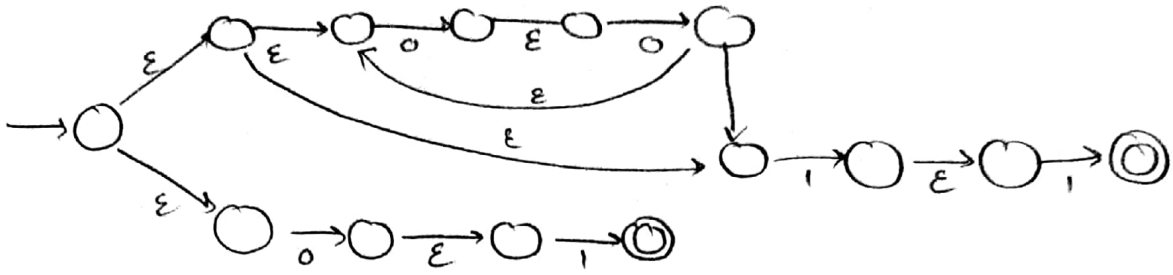


$(00)^*11$:

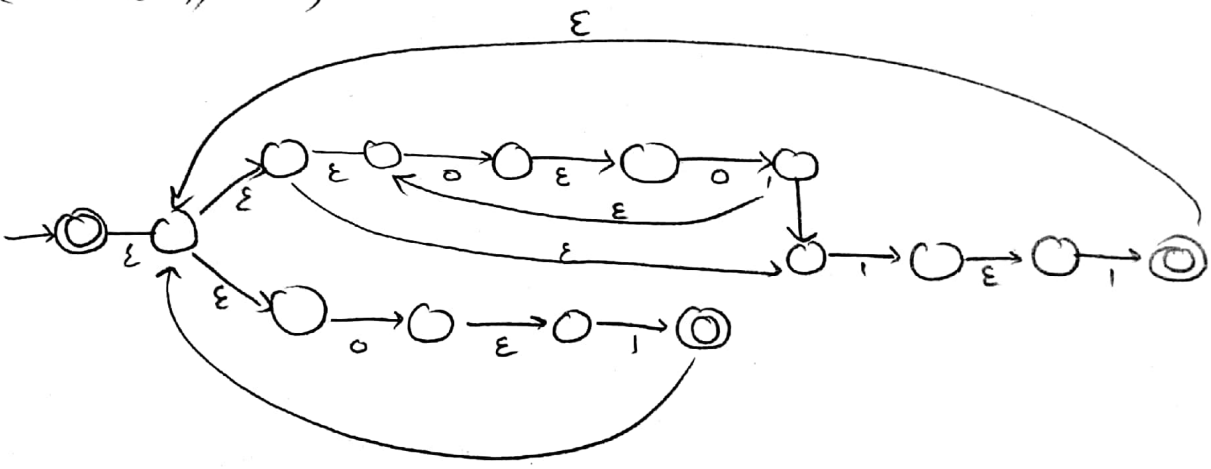


Scanned with CamScanner

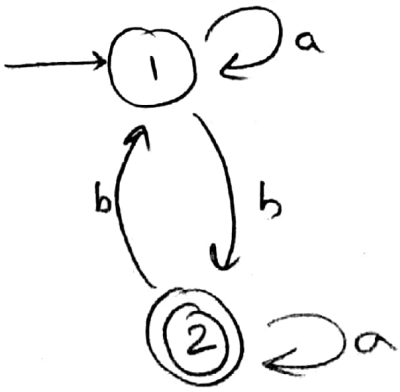
$(((00)^*(11)) \cup 01)$:



$(((00)^+(11)) \cup 01)^*$:



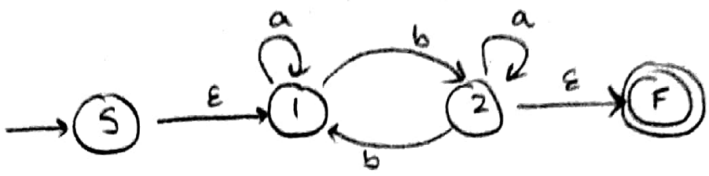
Q20 Convert following DFA to Regular expressions



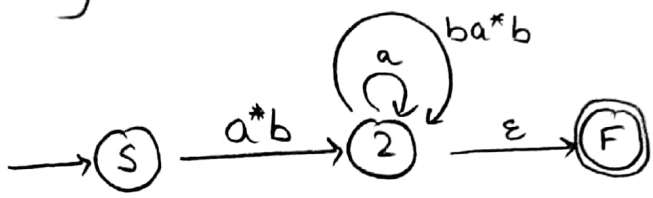
Scanned with CamScanner

Sol

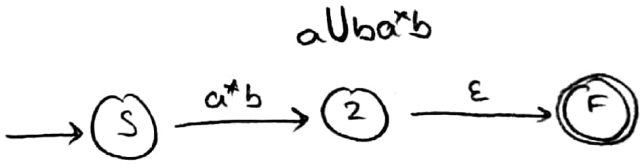
Lets add a start state (S) & accept state (F) to make original accept state as non-accept state



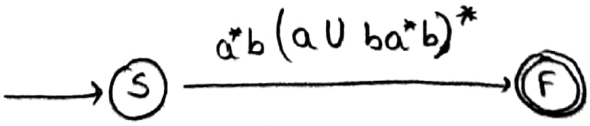
Eliminating state (1)



can convert a & ba*b using union



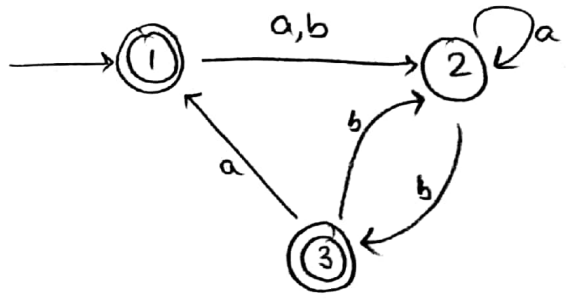
Eliminating State (2)



So Regular expression is $\boxed{a^*b(a \cup ba^*b)^*}$

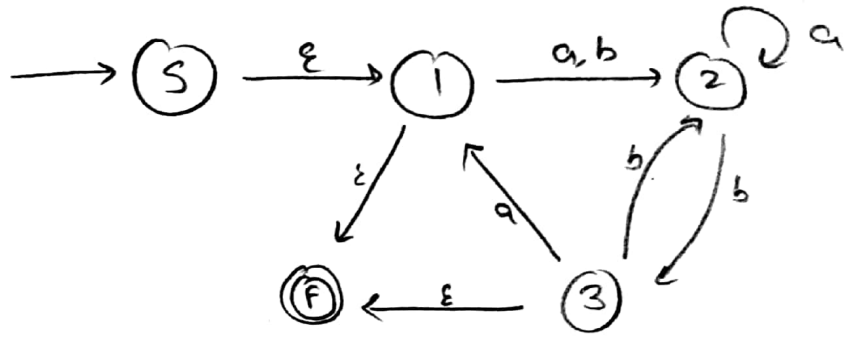
Scanned with CamScanner

Q 21 Convert DFA to Regular expression

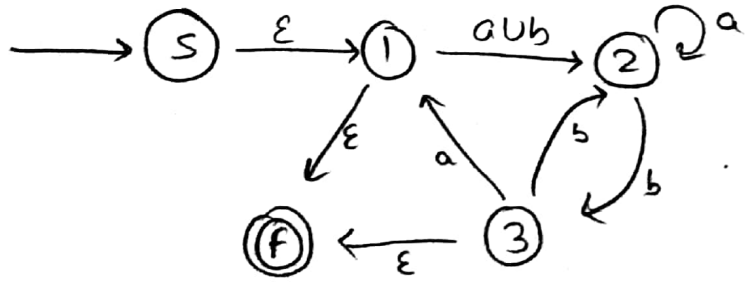


Sol

Adding new start state (S) & new accept state (F) and making accept state to non-accepting state

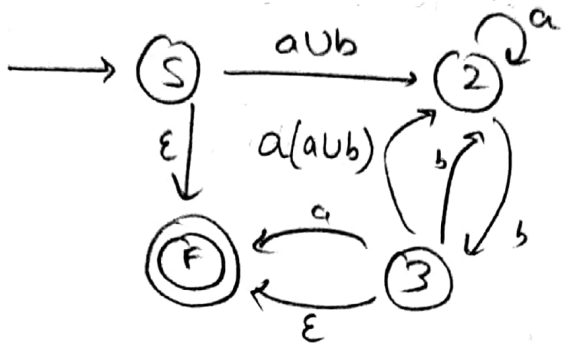


Union over state (1) and state (2)

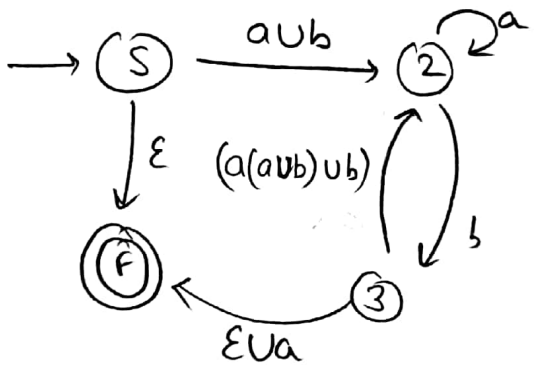


Scanned with CamScanner

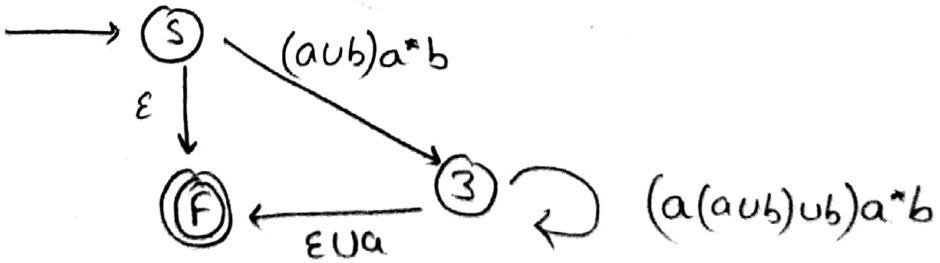
Eliminating state (1)



Forming Union b/w 2 & 3

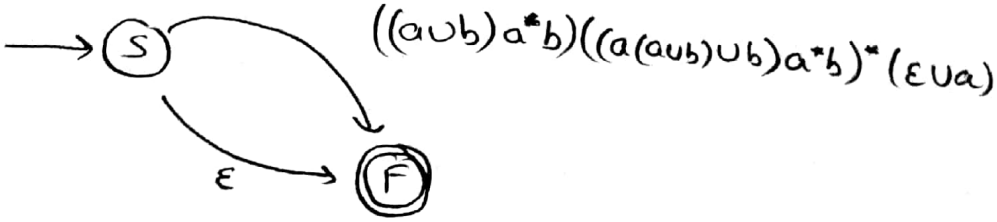


Eliminating State (2)

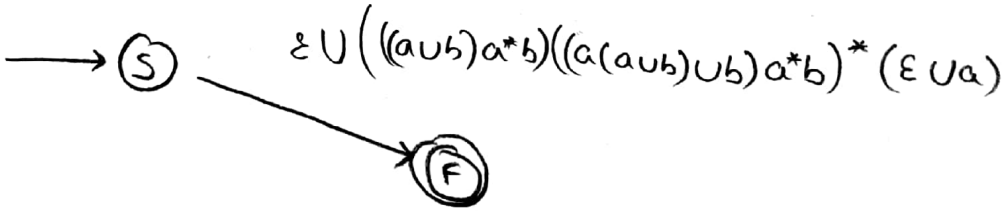


Scanned with CamScanner

Eliminating state (3)



Performing union



Regular expression for given DFA is

$$\boxed{\varepsilon \cup ((a \cup b)a^*b)((a(a \cup b) \cup b)a^*b)^*(\varepsilon \cup a)}$$

Scanned with CamScanner

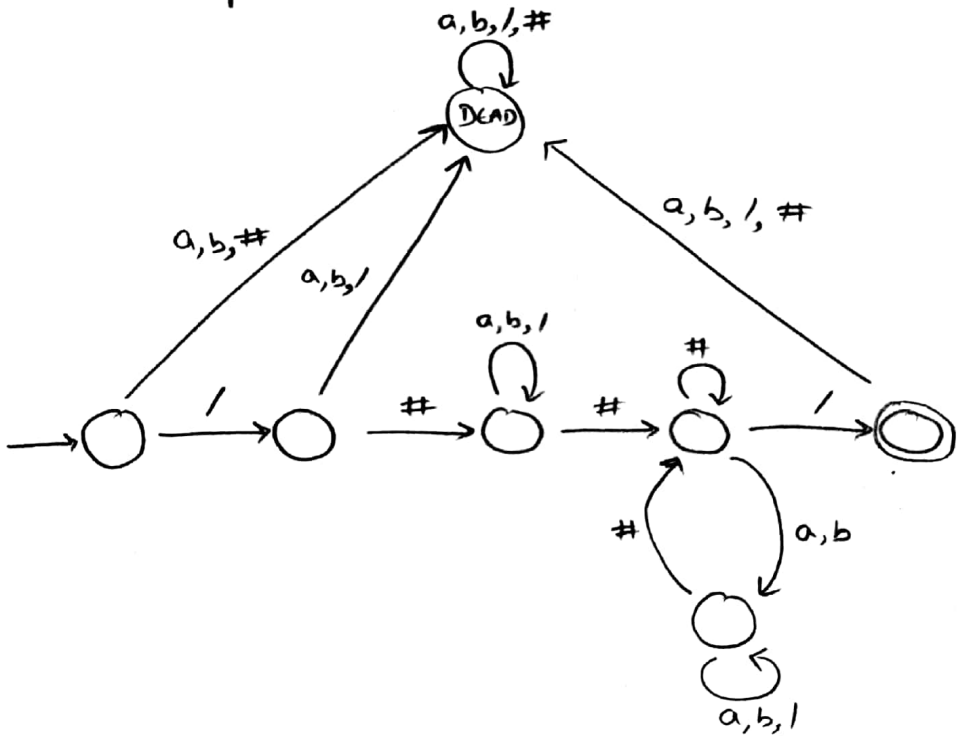
Q22

In certain programing language, comments appears b/w delimiters /# & #/ & other alphabets for C is $\Sigma = \{a, b, /, \#\}$. Give DFA

Sol

→ A member of C must begin with '/#' & end with '#/'

→ All alphabets $\Sigma = \{a, b, /, \#\}$



→ Regular expression

$$/\#(a+b+/)^*\#(\#+(a+b)(a+b+/)^*\#)^*/$$

Scanned with CamScanner

Q23

Let B be any language over alphabet Σ.
Prove $B = B^+$ iff $BB \subseteq B$

Sol

Assume $B = B^+$ — (1)
Showing $BB \subseteq B$

→ Since, for every language $BB \subseteq B^+$ — (2)

→ By substituting (1) in (2), it can be obtained that $BB \subseteq B$

→ Hence, it has been proved that $BB \subseteq B$ iff $B = B^+$

Assume $BB \subseteq B$
Showing $B^+ = B$

→ It is known that $BB \subseteq B^+$ — (3)

→ Let w be a string w can be split into elements $x_1 x_2 \cdots x_k$ such that $w = x_1 x_2 \cdots x_k$ for $x_i \in B$ & $k \geq 1$
Since $x_1 x_2 \in B$ and it is assumed that $BB \in B$

→ Similarly holds for x_3 such that $x_3 \in B$ & $BB \subseteq B$
Thus $x_1, x_2, x_3 \in B$

→ Continuing this $x_1, x_2 \cdots x_k \in B$ that is $w \in B$

→ It means if elements of string $(x_1 \cdots x_k)$ belongs to B then string $w \in B$

→ Thus $B^+ \subseteq B$ — (4)

→ From (3) & (4) $B = B^+$

→ Hence, it has been proved for both directions that, for any language B, $B = B^+$ iff $BB \subseteq B$

Hence Proved

Scanned with CamScanner

Q24

Using pumping lemma to show the following language are not regular

$$A_1 = \{ www \mid w \in \{a,b\}^* \}$$

Sol

→ Assume A_1 is a regular language

→ Let p be the pumping length given by the pumping lemma

→ Consider a string $s = a^p b a^p b a^p b \in A_1$

→ By pumping lemma string can divided into 3 pieces xyz such that $|xy| \leq p$, $|y| > 0$ and $xy^iz \in A_1 \ \forall i \geq 0$

→ So $s = a^p b a^p b a^p b = xyz$

→ Let aabaabaab be a string $\in A_1$. The pumping lemma of string is 2, $x = a$, $y = a$, $z =$ baabaab

$$s = \underset{x}{a} \ \underset{y}{a} \ \underset{z}{baabaab}$$

→ Pump the middle part such that xy^iz $(i \geq 0)$
For $i = 2$ y becomes aa

$$s = \underset{x}{a} \ \underset{y}{aa} \ \underset{z}{baabaab} \qquad (i=2)$$

→ The string aaabaabaab $\notin A_1$ hence contradiction

→ Therefore A_1 is not a regular language.

Hence Proved

Scanned with CamScanner

Q25

Use pumping lemma to show the language is not regular

$$A_3 = \{a^{2n} \mid n \geq 0\} \quad (a^{2^n} \text{ means string of } 2^n \text{ a's})$$

Sol → Lets assume A_3 is regular language

→ Let p be pumping length given by pumping lemma
Consider $s = a^{2p} \in A_3$ and $|s| > p$

→ By pumping lemma, this string divided into 3 pieces xyz such that $|xy| \leq p$, $|y| > 0$ & $xy^iz \in A_3 \; \forall \; i \geq 0$

→ Let say pumping length string is 2, $x = a$, $y = a$, $z = aa$

$$s = \underset{x}{a} \; \underset{y}{a} \; \underset{z}{aa}$$

→ Pumping $xy^iz \; (i \geq 0)$ for $i = 2$

$$s = \underset{x}{a} \; \underset{y}{aa} \; \underset{z}{aa} \quad (i=2)$$

→ The string $aaaaa \notin A_3$, Hence contradiction

→ Therefore A_3 is not a regular language

Hence Proved

Scanned with CamScanner

Q26

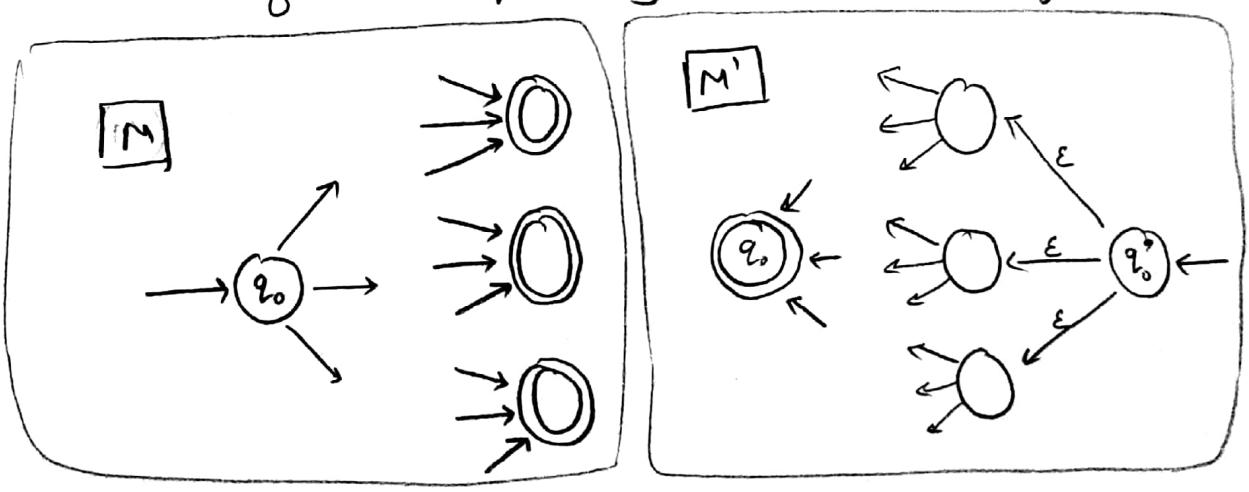
For any string $w = w_1 w_2 \cdots w_n$, the reverse of w, written w^R, is the string w in reverse order $w_n \cdots w_2 w_1$. For any language A, let $A^R = \{w^R \mid w \in A\}$. Show that if A is regular so is A^R.

Sol

Let $M = (Q, \Sigma, \delta, q_0, F)$ be DFA that recognizes A

Now we build a NFA M' for A^R as follow

- Reverse all arrows of M
- Convert the start state for M as only accept state q'_{accept} for M'
- Add a new start state q'_0 for M' and from q'_0 add ϵ transitions to each state of M' corresponding to accept state of M



Here $q'_0 = q'_{accept}$

- For any $w \in \Sigma^*$ there is a path following w from the start state to an accept state in M iff there is a path following w^R from q'_0 to q'_{accept} in M'
- That means that $w \in A$ iff $w^R \in A^R$.

Scanned with CamScanner

Q27

Let $\Sigma_2 = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$

Consider each row to be binary number & let

$D = \{ w \in \Sigma_2^* \mid$ the top of row w is larger number than is the bottom row$\}$

Show D is regular

Sol

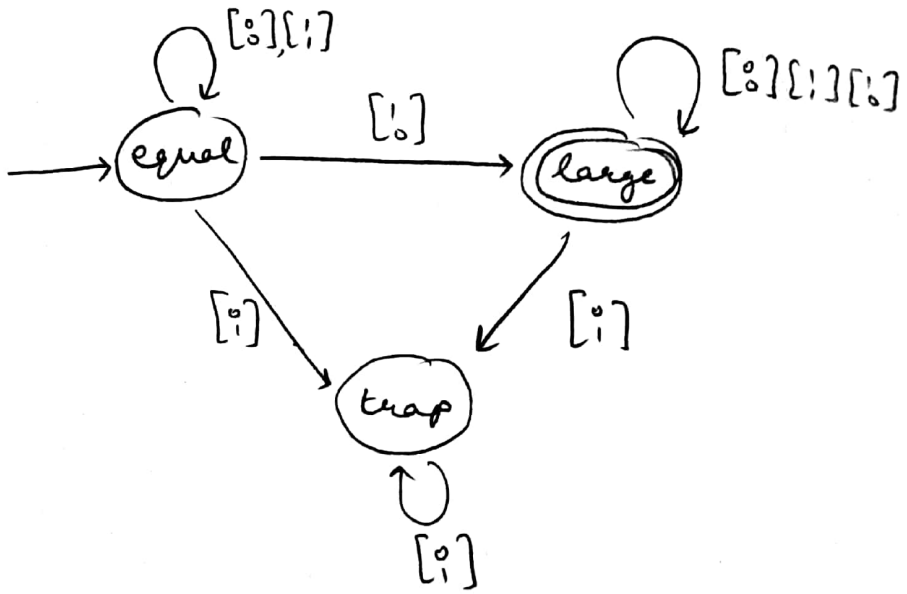
Language for given expression

$$L = \left\{ \varepsilon, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \ldots \right\}$$

Here each row is binary number

Let M be DFA, over input alphabet Σ_2

State transition diagram of M is as follows



Scanned with CamScanner

⇒ for D to be a regular language.

A language is said to be regular if it recognize by DFA

Let take string from language

$$D \Rightarrow w = \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$$

Initial state of above DFA is <u>equal</u>

Parse string

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

equal ⟶ large ⟶ large

Here 'large' is final state, the string accepted by DFA

Thus, language of given D is accepted by given DFA we defined a DFA to recognize language D

<u>Therefore, D is regular language</u>

Scanned with CamScanner

Q28

Let B and C be languages over $\Sigma = \{0, 1\}$

Define $B \leftarrow C = \{w \in B \mid$ for some $y \in C$, strings w & y contains equal numbers of 1s$\}$

Show that the class of regular language is closed under $\leftarrow$ operation.

Sol

If B and C are regular languages then $B \leftarrow C$ is also a regular language

So given B & C are regular languages

So let M_B be DFA recognizing language B

$$M_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$$

& M_C be DFA recognizing langug C

$$M_C = (Q_C, \Sigma, \delta_C, q_C, F_C)$$

Now constructing NFA to recognize $B \leftarrow C$

Let $N = (Q, \Sigma, \delta, q_0, F)$ be NFA

and N has to decide whether a string $w \in B \leftarrow C$ or not

For that first machine checks whether $w \in B$ or not, if $w \in B$ then non-determini-stically find out string y that contains same number of 1s as in w & checks that $y \in C$

This tells that each string B, there are C parallel machines will exist

Scanned with CamScanner

Thus $Q = Q_B \times Q_C$

Σ = same as B & C

δ : given by, for $(q, r) \in Q$ & $a \in \Sigma$

$$\delta(q, r), a = \begin{cases} \{(\delta_B(q,0), r)\} & \text{if } a=0 \\ \{((\delta_B(q,1)), \delta_C(r,1))\} & a=1 \\ \{(q, \delta_C(r,0))\} & a=\epsilon \end{cases}$$

$q_0 = (q_B, q_C)$

$F = F_B \times F_C$

Thus we defined an NFA N to recognize $B \leftarrow C$

Hence $B \leftarrow C$ is regular

Therefore class of regular languages closed under $B \leftarrow C$ operation.

Scanned with CamScanner

Q29

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

ADD = $\{x = y + z \mid x, y, z$ are binary & x is sum of y & $z\}$

Show ADD is not regular

Sol

The language is said to be regular if it satisfies pumping lemma

⇒ Pumping lemma says for a p (pumping length) for a string $S \in A$ of length atleast P, then S may be divided into 3 pieces, $S = uvw$

For each $i \geq 0$ $uv^iw \in A$; $|v| > 0$; $|uv| \leq p$

Let go ahead with assuming ADD is regular language

Let p be pumping length & string be $x = y + z$

Considering string to $1^{p+1} = 10^p + 1^p \in ADD$

for $p = 2$; $111 = 100 + 11 \in ADD$

Dividing $S = 111 = 100 + 11$ into 3 parts

$$= \underset{u}{1} \; \underset{v}{1} \; \underset{w}{1 = 100 + 11}$$

Pump middle part by 2 so that $v = 11$

$S = (1)(1)^i(1 = 100 + 11)$ for i

$S = (1)(11)(1 = 100 + 11)$ $(i = 2)$

$$= \underset{u}{1} \; \underset{v}{1} \; \underset{w}{1 = 100 + 11} \quad (i = 2)$$

Scanned with CamScanner

Now the string we got after pumping is $1111 = 1100 + 11$

This string $\notin$ ADD because addition of 2 binary numbers 100, 11 is 111 not 1111

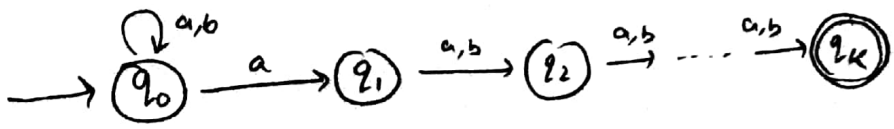
Hence contradiction due to violation of pumping lemma

Therefore, ADD is not a regular language.

Q30 Let $\Sigma = \{a, b\}$. For each $k \geq 1$ let C_k be language consisting of all strings that contains an a exactly k places from right-hand end. Thus $C_k = \Sigma^* a \Sigma^{k-1}$. Describe an NFA with k+1 states that recognizes C_k, both in terms of state & formal description.

Sol) So $C_k = \Sigma^* a \Sigma^{k-1}$ for each $k \geq 1$; $\Sigma = \{a, b\}$

Let N be NFA with k+1 states that recognizes C_k



Scanned with CamScanner

Formal Description of NFA N

$$N = (Q, \Sigma, \delta, q_0, F)$$

Q = set of states $= \{q_0, q_1 \cdots q_k\}$

Σ = set of alphabet $= \{a, b\}$

q_0 = start state $= \{q_0\}$

F = set of final state $= \{q_k\}$

δ = The transition function

$$\delta(q_i, a) = \begin{cases} \{q_0, q_1\} & i = 0 \\ \{q_{i+1}\} & 0 < i < k \\ \phi & i = k \end{cases}$$

$$\delta(q_i, b) = \begin{cases} \{q_0\} & i = 0 \\ \{q_{i+1}\} & 0 < i < k \\ \phi & i = k \end{cases}$$

$$\delta(q_i, \epsilon) = \phi \ \forall i$$

Scanned with CamScanner

Q31

Prove that for each $n > 0$, language B_n exists where

(a) B_n is recognizable by NFA that has n states

(b) $B_n = A_1 \cup A_2 \cup \cdots \cup A_k$ for regular languages A_i, than atleast one of A_i requires DFA with exponentially many states.

Sol

(a)

Suppose B_n be a language where $n > 0$.
Now has to prove that language is recognisable by an NFA with n states

Basis Let $n=1$ hence $B_n = \{\varepsilon, 0, 1\}$. Therefore we can design an NFA $N = (\{q_0\}, \Sigma, \delta, q_0 \{q_0\})$ with a single state given language as
$\delta(q_0, \varepsilon|0|1) = q_0$

Proving By Induction

Divide B_n in two regular expression say E & F of length $n_1, n_2 < n$ & $n_1 + n_2 = n$
By inductive hypothesis it can easily concluded that NFA's accepting E & F consisting n_1 & n_2 states but it is already known to us that set of regular expression is clousure under union

Therefore language B_n is recognizable by NFA with n states

Scanned with CamScanner

(b) Now for $B_n = A_1 \cup A_2 \cup \cdots \cup A_k$ where A_i's are regular

If a DFA is constructed ~~with~~ which is equivalent to DFA of given NFA

There could be atleast n & atmost 2^n states in resultant equivalent DFA Every regular language is ~~cons~~ recognized by DFA so there is a corresponding DFA for all A_is. Now by peigion hole principle, one can state that there is atleast on DFA which requires 2^i states to recognize a language among all the A_i

Scanned with CamScanner

Q 32

If A is any language, let $A_{\frac{1}{3}-\frac{1}{3}}$ be set of all strings in A with their middle thirds removed

$$A_{\frac{1}{3}-\frac{1}{3}} = \{xz \mid \text{for } y, |x| = |y| = |z| \text{ and } xyz \in A\}$$

Show that, if A is regular then $A_{\frac{1}{3}-\frac{1}{3}}$ is not regular.

Sol

Let $A = \{a^* \# b^*\}$ is reguler Language

we know that $\{a^* b^*\}$ is reguler language

Now $A_{\frac{1}{3}-\frac{1}{3}} \cap \{a^* b^*\} = \{a^n b^n \mid n \geq 0\}$

Clearly $\{a^n b^n \mid n \geq 0\}$ is not regular because if P is pumping length and

$S = xyz = aabb$ is $p = 2$

$x = a \quad y = a \quad z = bb$

For $S = xy^i z \quad \forall i$

For $i = 2 \quad S = xy^2 z = aaabb$

$$= \underset{x}{a}\ \underset{y}{aa}\ \underset{z}{bb}$$

$xy^2 z = a^3 b^2 \notin \{a^* b^*\}$

Hence $\{0^* 1^*\}$ is not regular

As regular languages are closed under intersection and $\{0^* 1^*\}$ is not regular.

$A_{\frac{1}{3}-\frac{1}{3}}$ is not reguler

Therefore, If A is reguler then $A_{\frac{1}{3}-\frac{1}{3}}$ is not necessarily regular is proved

Scanned with CamScanner

Q 33

Consider grammar $G_4 = (V, \Sigma, R, \langle EXPR \rangle)$

V is $\{\langle EXPR \rangle, \langle TERM \rangle, \langle FACTOR \rangle\}$

Σ is $\{a, +, \times, (,)\}$

Rules:-

$\langle EXPR \rangle \rightarrow \langle EXPR \rangle + \langle TERM \rangle \mid \langle TERM \rangle$

$\langle TERM \rangle \rightarrow \langle TERM \rangle \times \langle FACTOR \rangle \mid \langle FACTOR \rangle$

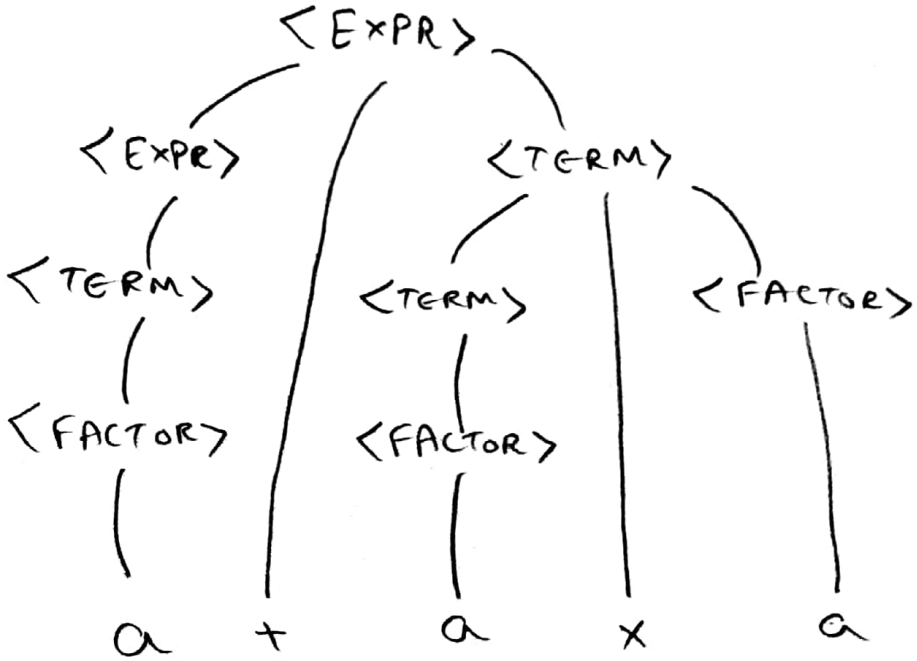
$\langle FACTOR \rangle \rightarrow (\langle EXPR \rangle) \mid a$

Give parse tree for

(i) $a + a \times a$ (ii) $(a+a) \times a$

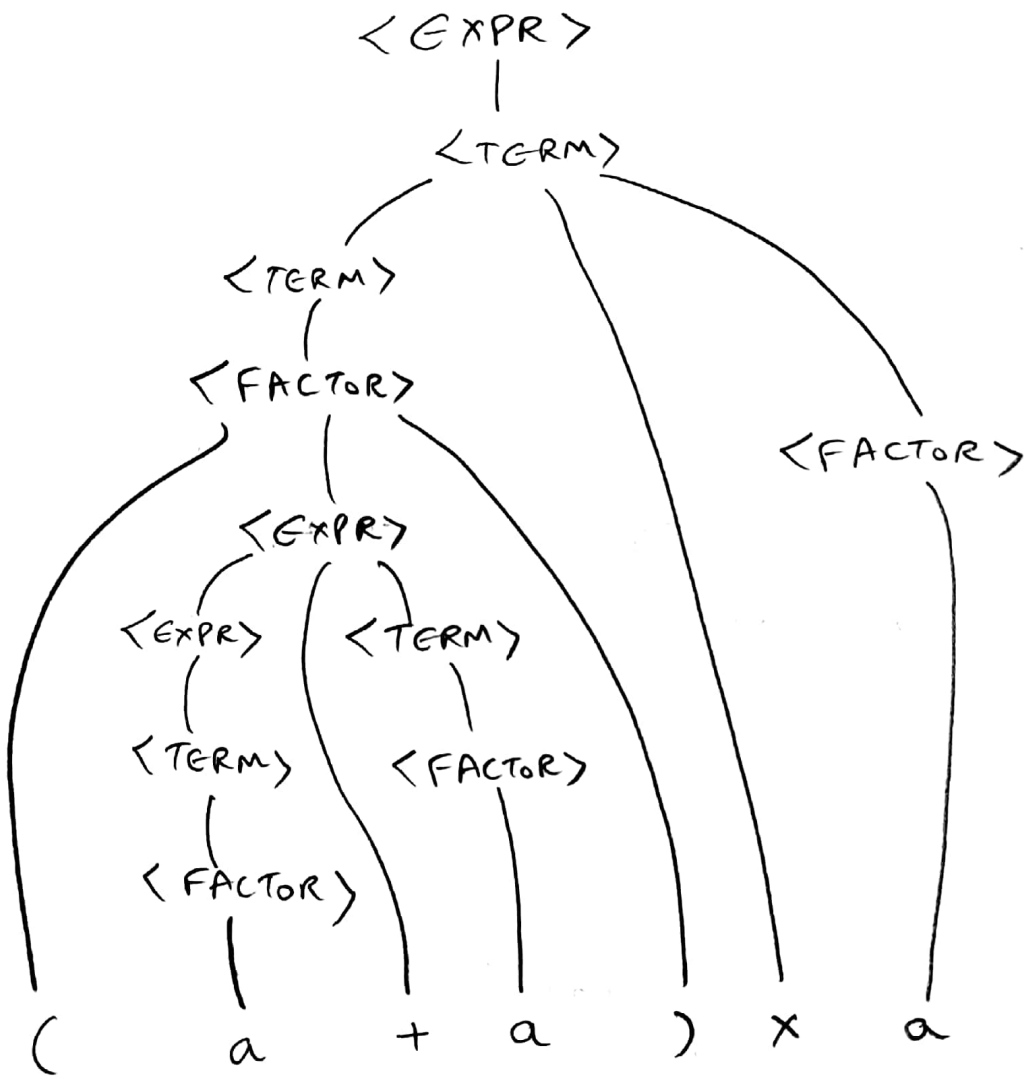
Sol

For $(a + a \times a)$



Scanned with CamScanner

(ii) For $(a+a) \times a$



Scanned with CamScanner

Q 34

Give context-free grammar for following languages for $\Sigma = \{0,1\}$

Sol

(i) $\{w \mid w$ contains atleast 3 1s$\}$

$S \rightarrow P1P1P1P$

$P \rightarrow 0P \mid 1P \mid \varepsilon$

(ii) $\{w \mid w$ starts & ends with same symbol$\}$

$S \rightarrow 0P0 \mid 1P1 \mid 0 \mid 1$

$P \rightarrow 0P \mid 1P \mid \varepsilon$

(iii) $\{w \mid$ the length of w is odd$\}$

$S \rightarrow 0 \mid 1 \mid 0S0 \mid 01S \mid 10S \mid 11S$

(iv) $\{w \mid w = w^R$, w is palindrome$\}$

$S \rightarrow 0 \mid 1 \mid 0S0 \mid 1S1 \mid \varepsilon$

(v) empty set

$S \rightarrow S$

Scanned with CamScanner

Q 35

Let C be a context-free language R be a regular language. Prove that language $C \cap R$ is context-free.

Sol

Let C be Context free language & R be regular

Now we have to prove $C \cap R$ is CFL

Let P be PDA recognize C, and
D be DFA recognize R

PDA that recognize $C \cap R$ with set of states $Q \times Q'$

where Q be set of states P
Q' be set of state of D

Here P' will do what P does & keep track of states D

The PDA that recognize $C \cap R$ accepts the string w if & only if it stops a state $q \in F_P \times F_D$ where

F_P is states of accepts of P
F_D is states of accepts of D

So $C \cap R$ recognized by P'

Therefore, $C \cap R$ is context-free language.

Scanned with CamScanner

Q 36

$C = \{x\#y \mid x, y \in \{0,1\}^* \text{ and } x \neq y\}$. Show that C is a context-free language

Sol

- Given that a string $x\#y$ is language C if and only if $x \neq y$ or strings x & y are vary at some specific position
- Its CFG is

$$S \longrightarrow A\#B \mid B\#A$$
$$A \longrightarrow TAT \mid 0$$
$$B \longrightarrow TBT \mid 1$$
$$T \longrightarrow 0 \mid 1$$

As grammar for C is defined in terms of CFG The language produces a string that contains $x\#y$, and x & y are different character for same index position

Hence, it is proved that C is CFL.

Proved

Scanned with CamScanner

Q 37

Let $\Sigma = \{1,2,3,4\}$ and $C = \{w \in \Sigma^* \mid$ in w, the number of 1's equals the numbers of 2's and number of 3's equals numbers of 4's$\}$. Show that C is not context free.

Sol

Consider C is context free. So, C has pumping length p

Take $s = 1^p 3^p 2^p 4^p \in C$ with $|s| > p$

Therefore, there exist $uvxyz$ such that

(a) $uv^ixy^iz \in C$ for $i \geq 0$

(b) $vy > 0$

(c) $vxy \leq p$

Case I

If vxy contains a (1 or 2) then $uv^2xy^2z \notin C$ since it cannot be same number of 1s & 2s. Hence contradiction

Case II

If vxy contains (3 or 4), then $uv^2xy^2z \notin C$ since it cannot be same number of 3s & 4s. Hence contradiction

Hence it always contradict

Therefore C is not context free

Scanned with CamScanner

Q 38

Show that class of context free languages is closed under regular operations, union, concatenation and star

Sol. **Union:**

In order to show context free language closed under union operation consider S_1 & S_2 & 2 diff. language L_1 & L_2

Grammar

$$S \rightarrow S_1 | S_2$$

If both language belongs to context free language then union of both should also belong to context free language

Hence $L_1 \cup L_2 \in CFL$

This implies context free language is closed under union operation.

Concatenation:

Grammar

$$S \longrightarrow S_1 S_2$$

If both language belongs to context free language then concatenation of both language should belong to context free language $\forall$ $L_1, L_2 \in CFL$

Hence, $\{w_1 w_2 : w_1 \in L_1 \wedge w_2 \in L_2\} \in CFL$

This implies context free language is closed under concatenation operation.

Scanned with CamScanner

Star :

Consider one start variable S_1 for language L_1

Grammar

$$S \rightarrow S_1 S \mid \epsilon$$

If languge belongs to context free language then star of language should belong to context free language $\forall$ $L_1 \in CFL$

By above, we can say if we generate zero or many string which is the defination of star

This implies that context free language is closed under star operation.

Scanned with CamScanner

Q 39

Give context-free grammar generating for language

The complement of language $\{a^n b^n \mid n \geq 0\}$

Sol

Let L be language that is a compliment of given language. L can be obtained as

$$L = \{a^n b^m : n \neq m\} \cup \{(a \cup b)^* ba (a \cup b)^*\}$$

Let us consider

$$L_1 = \{a^n b^m : n \neq m\}$$

$$L_2 = \{(a \cup b)^* ba (a \cup b)^*\}$$

Grammar for L_1

$$S_1 \rightarrow aS_1b \mid T \mid U$$

$$T \rightarrow aT \mid a$$

$$U \rightarrow Ub \mid a$$

Grammar for L_2

$$S_2 \rightarrow RbaR$$

$$R \rightarrow RR \mid a \mid b \mid \varepsilon$$

Therefore $L = L_1 \cup L_2$

$$S \rightarrow S_1 \mid S_2$$

$$S_1 \rightarrow aS_1b \mid T \mid U$$

$$S_2 \rightarrow RbaR$$

$$T \rightarrow aT \mid a$$

$$U \rightarrow Ub \mid a$$

$$R \rightarrow RR \mid a \mid b \mid \varepsilon$$

Scanned with CamScanner

Q 40

Give unambigous CFG's for following language

(i) {w | the number of a's & b's in w are equal}

Sol

Grammer:

$S \rightarrow aSb \mid bSa \mid \varepsilon$

In above grammar, production rules $S \rightarrow aSb$ and $S \rightarrow bSa$ generates equal number of terminals a's and b's

$S \rightarrow aSb$

$\rightarrow a(aSb)b$

$\rightarrow aa(aSb)bb$

$\rightarrow aaabbb$

(ii) {w | is every prefix of w the number of a's is atleast the number of b's}

Sol

Grammer:

$S \rightarrow aS \mid a \mid b \mid \varepsilon$

The production for regular expression aaab

$S \rightarrow aS$

$\rightarrow aaS$

$\rightarrow aaaS$

$\rightarrow aaab$

This grammar is an unambiguous grammer because there is no possible way to get same regular expression.

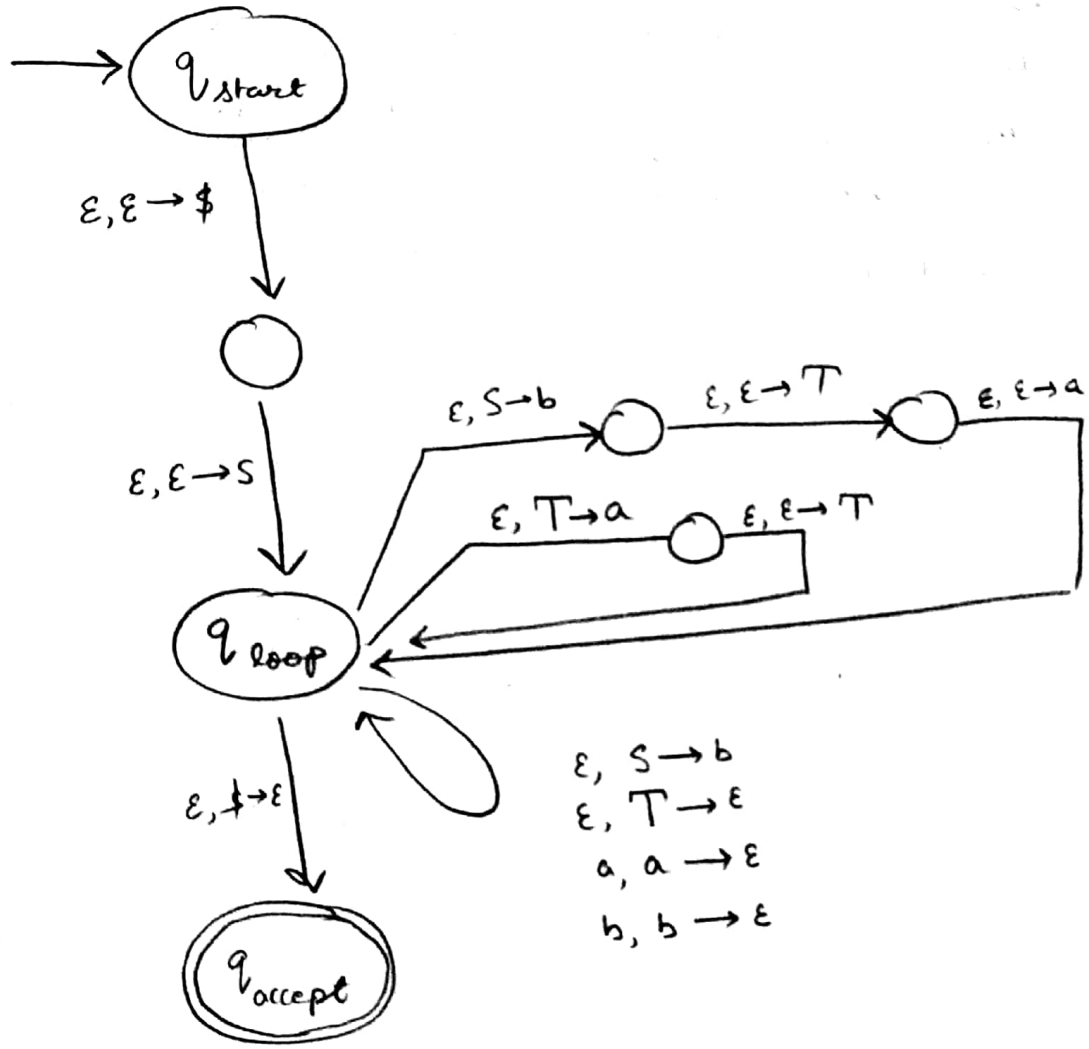
Scanned with CamScanner

Q 41 Construct a PDA from following CFG

$S \rightarrow aTb \mid b$

$T \rightarrow Ta \mid \varepsilon$

Sol

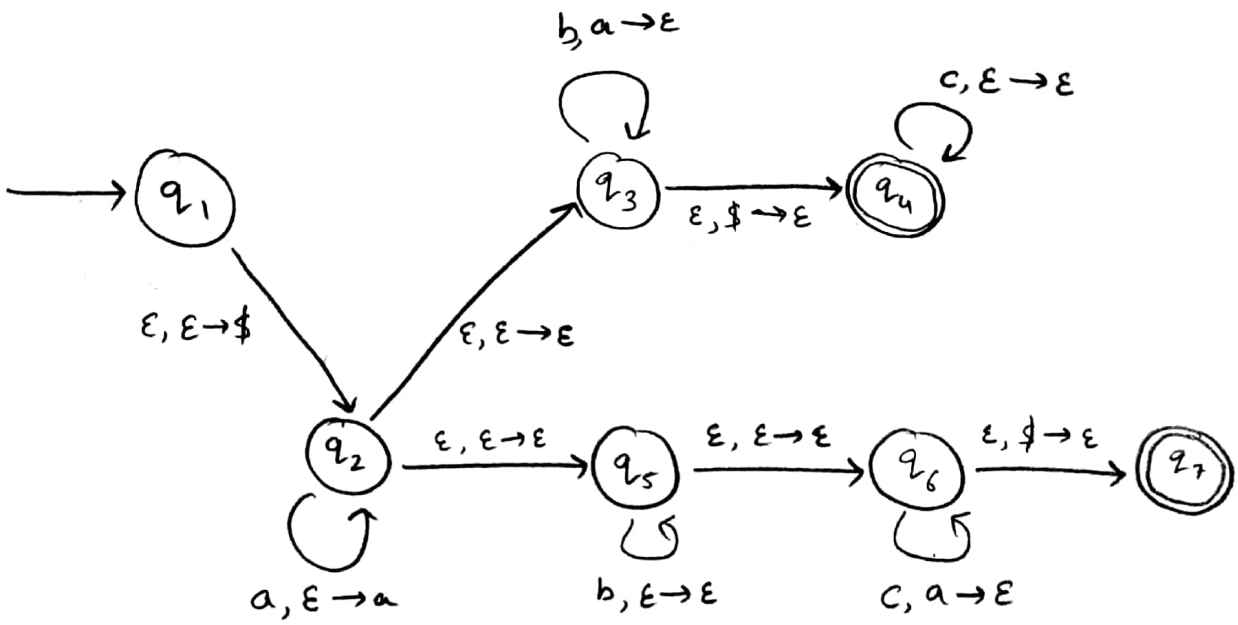


Scanned with CamScanner

Q42 Give PDA for language

$\{ a^i b^j c^k \mid i, j, k \geq 0 \ \& \ i = j \text{ or } i = k \}$

Sol



Scanned with CamScanner

Q 43

Give context-free grammar generates language

$$A = \{a^i b^j c^k \mid i=j \text{ or } j=k \text{ where } i,j,k \geq 0\}$$

Is your grammar ambigous? why or why not?

Sol

The language A can be split into 2 languges

$$A_1 = \{a^i b^j c^k \mid i,j,k \geq 0,\ i=j\}$$

$$A_2 = \{a^i b^j c^k \mid i,j,k \geq 0,\ j=k\}$$

Using language A_1 & A_2 the grammar of following will be

$$S = S_1 \mid S_2$$

for A1

$$S_1 \rightarrow S_1 c \mid E \mid \varepsilon$$

$$E \rightarrow aEb \mid \varepsilon$$

for A2

$$S_2 \rightarrow aS_2 \mid F \mid \varepsilon$$

$$F \rightarrow bFc \mid \varepsilon$$

Since generating a string $w = a^n b^n c^n$ using language A, either S_1 or S_2 can be used

Therefore, the context free grammar for language A is ambiguous.

Scanned with CamScanner

Q 44

Let $G = (V, \Sigma, R, \langle STMT \rangle)$ be following grammar

$\langle STMT \rangle \rightarrow \langle ASSIGN \rangle \mid \langle IF\text{-}THEN \rangle \mid \langle IF\text{-}THEN\text{-}ELSE \rangle$

$\langle IF\text{-}THEN \rangle \rightarrow$ if condition then $\langle STMT \rangle$

$\langle IF\text{-}THEN\text{-}ELSE \rangle \rightarrow$ if condition then $\langle STM \rangle$ else $\langle STMT \rangle$

$\langle ASSIGN \rangle \rightarrow a := 1$

$\Sigma = \{$if, condition, then, else, $a := 1\}$

$V = \{\langle STMT \rangle, \langle IF\text{-}THEN \rangle, \langle IF\text{-}THEN\text{-}ELSE \rangle, \langle ASSIGN \rangle\}$

Show that G is ambigous & give new unambigous grammar for same language

Sol

1st derivation

$\langle STMT \rangle$

$\Rightarrow \langle IF\text{-}THEN \rangle$

$\Rightarrow$ if condition then $\langle STMT \rangle$

$\Rightarrow$ if condition then $\langle IF\text{-}THEN\text{-}ELSE \rangle$

$\Rightarrow$ if condition then if condition then $\langle STMT \rangle$ else $\langle STMT \rangle$

$\Rightarrow$ if condition then if condition then $\langle ASSIGN \rangle$ else $\langle STMT \rangle$

$\Rightarrow$ if condition then if condition then $a := 1$ else $\langle STMT \rangle$

$\Rightarrow$ if condition then if condition then $a := 1$ else $\langle ASSIGN \rangle$

$\Rightarrow$ if condition then if condition then $a := 1$ else $a := 1$

Scanned with CamScanner

Sencond Derivation :

<STMT>

=> <If - THEN - ELSE>

=> if condition then <STMT> else <STMT>

=> if condition then <IF - ELSE> else <STMT>

=> if Condition then if condition then <STMT> else <STMT>

=> if condition then if Condition then <ASSIGN> else <STMT>

=> if Condition then if condition then a:1 else <STMT>

=> if condition then if condition then a:1 else <ASSIGN>

=> if condition then if conditi then a:1 else a:1

In both cases when it takes 'if then' or 'if - then - else' result is same. Hence both are same left derivation & it is ambiguous.

New Grammer

<STMT> => <IF - THEN> | <ASSIGN> | <IF - THEN - ELSE>

<IF - THEN - ELSE> => if condition then <STMT 1> else <STMT>

<STMT 1> => <IF - THEN - ELSE> | <ASSIGN>

<IF - THEN> => if condition then <STMT>

<ASSIGN> => a:=1

Scanned with CamScanner

Q 45

Show that if G is CFG in chomsky normal form, then string $w \in L(G)$ of length $n \geq 1$, exactly $2n-1$ steps required for any derivation w.

Sol

For $n=1$; Steps = 1

String 'a' of length 1

Derivation will be $S \to a$

Hence true for $(n=1)$

Lets say it is valid for $k=n$

Now for $n=k+1$

Since $n>1$, derivation starts from S

$S \to BC$

$B \to {}^*x$

$C \to {}^*y$

So length $|w| = xy$ $\quad |x|>0$; $|y|>0$

Using inductive hypothesis, for above language in CNF the length of any derivation of string w

$1 + (2|x|-1) + (2|y|-1) = 2|x| + 2|y| + 1 - 1 - 1$

$= 2(|x|+|y|) - 1$

Here $n = |x| + |y|$

Since $B \to {}^*x$ has length $|x|$ & $C \to {}^*y$ has length $|y|$

Hence, it is proved that it requires $2n-1$ steps required for derivation of string $w \in L(G)$ in CNF

Scanned with CamScanner

Q 46

Let G be CFG in CNF that contains b variables
Show that if G generates some string with a derivation having atleast 2^b steps, L(G) is infinite

Sol

CFG

- A finite set of grammar rules is known as CFG
  It consisting of that is quadruple (N, T, P, S)
- T is group of terminal $N \cap T = NULL$
- P is group of rule $P: N \rightarrow (N \cup T)^*$
- S is start symbol.

CNF

$U \rightarrow u$

$U \rightarrow VW$

where U, V, W are non-terminal and u is terminal

- At most 2 terminal can generate in every deviation
- In any parse string use of G
- Internal node can have at most 2 children
- Parse tree with height k has at most 2^{k-1} internal node
- If several string generated G with a derivation taking at least 2^b steps
- At least consuming b+1 height in parse tree
- It exists a path from root to leaf containing b+1 variable
- In this one variable occuring twice

Scanned with CamScanner

Q 47

Convert CFG for G_4 below to equivalent PDA

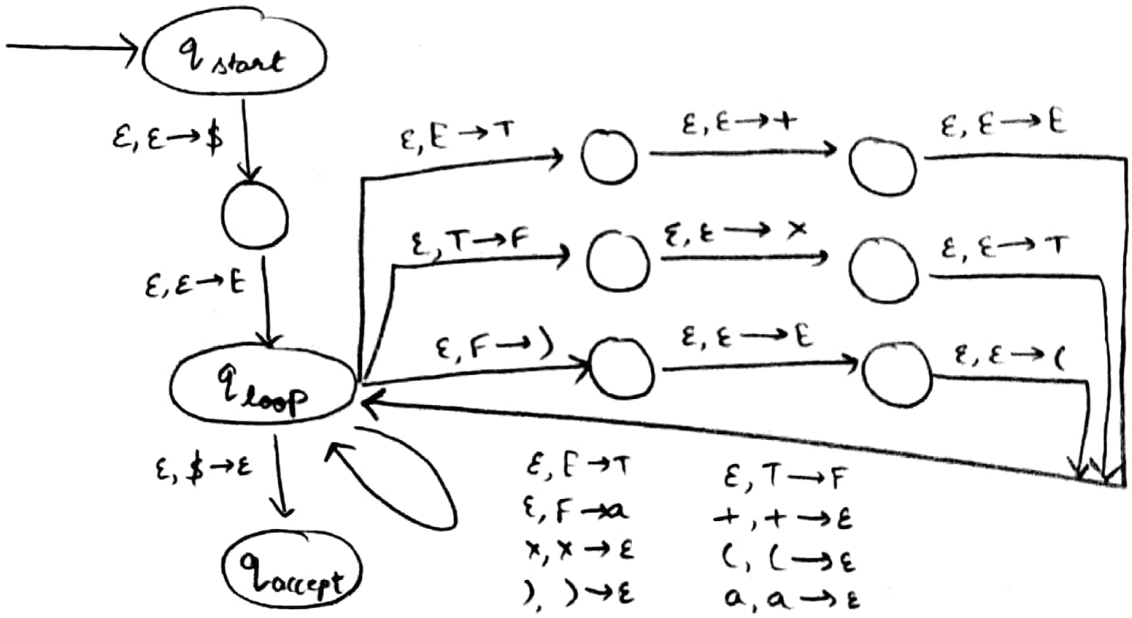
$$E \rightarrow E+T \mid T$$

$$T \rightarrow T \times F \mid F$$

$$F \rightarrow (E) \mid a$$

Sol

Equivalent PDA for CFG is



Explanation:

- A shorthand notation for pushing multiple symbol onto stack
- Initially '\$' is inserted q_{start} is start state.
- Transition function $\delta(q_{start}, \varepsilon, \varepsilon) = \{(q_{loop}, S\$)\}$
- Stack top in non-terminal variable E
- Transition function $\delta(q_{loop}, \varepsilon, E) = \{(q_{loop}, w) \mid E \rightarrow w \text{ is rule of CFG}\}$

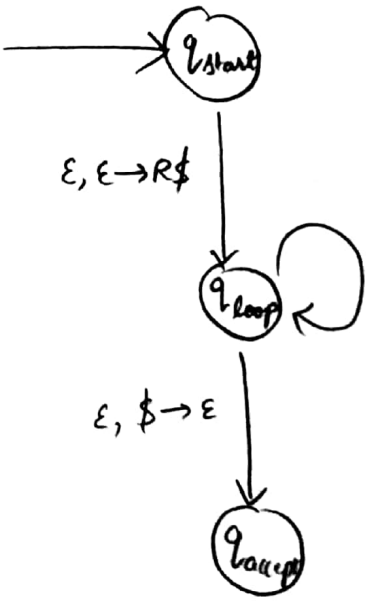
Scanned with CamScanner

Q 48 Convert CFG below to equivalent PDA

$$R \rightarrow XRX \mid S$$
$$S \rightarrow aTb \mid bTa$$
$$T \rightarrow XTX \mid X \mid \varepsilon$$
$$X \rightarrow a \mid b$$

Sol

Equivalent PDA for CFG:



$\varepsilon, R \rightarrow XRX$
$\varepsilon, R \rightarrow S$
$\varepsilon, S \rightarrow aTb$
$\varepsilon, S \rightarrow bTa$
$\varepsilon, T \rightarrow XTX$
$\varepsilon, T \rightarrow X$

$\varepsilon, T \rightarrow \varepsilon$
$\varepsilon, X \rightarrow a$
$\varepsilon, X \rightarrow b$
$a, a \rightarrow \varepsilon$
$b, b \rightarrow \varepsilon$

Explanation

- replace non-terminals or variables to right hand side terminal
- Transition for terminal symbol such as (a, b) permits someone for matching the input symbol to terminal symbol.
- A path of PDA of string w can only be accepted if an input string w can only be generated by gramar G

Scanned with CamScanner

Q 49

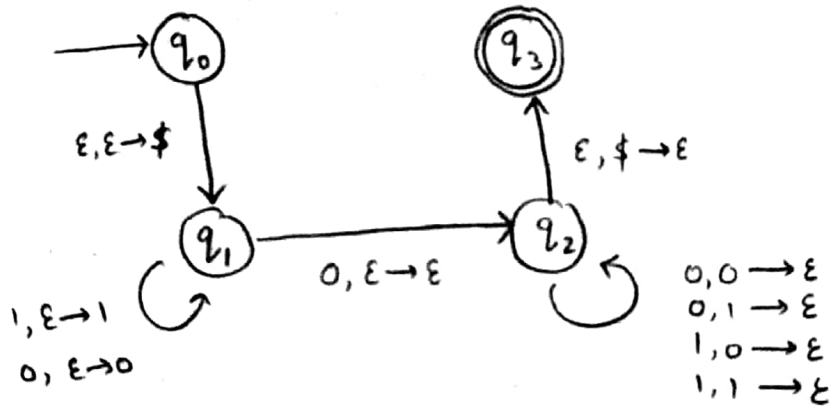
Give informal description & state diagrams of pushdown automata for language

$\{w \mid$ odd length and middle symbol is $0\}$

Sol Grammar:

$$S \rightarrow 0S0 \mid 0S1 \mid 1S0 \mid 1S1 \mid 0$$

Pushdown automata:



Discription:

initial state q_0

final state q_3

In each step it checks whether the middle symbol is 0. If stack is empty & middle symbol is 0 then string is accepted by pushdown automata otherwise pushdown automata will reject the input.

Scanned with CamScanner

Q50 Let $\Sigma = \{0,1\}$ show this language is **CFL**

$C = \{xyz \mid x, z \in \Sigma^* \text{ and } y \in \Sigma^* 1 \Sigma^*, \text{ where } |x| = |z| \geq |y|\}$

Sol

a push down automata needs to construct which help in determining language C.

PDA is recognize C is $(Q, \Sigma, \Gamma, \delta, q_0, F)$ where

$Q = \{q_0, q_1, q_2\}$

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

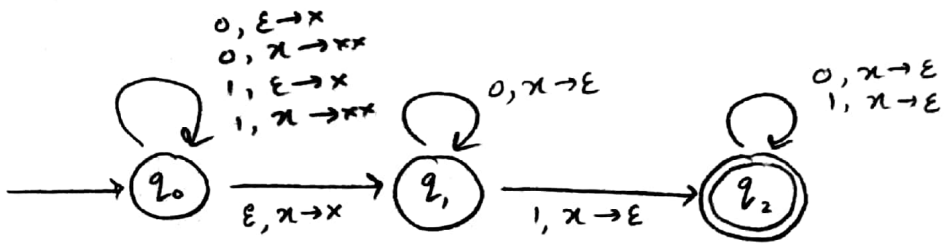
$\Gamma = \{x\}$

$F = \{q_2\}$

δ = Transition fuction

Input	0		1		ε	
Stack	x	ε	x	ε	x	ε
q_0	$\{(q_0, xx)\}$	$\{(q_0, x)\}$	$\{(q_0, xx)\}$	$\{(q_0, x)\}$	$\{(q_1, x)\}$	
q_1	$\{(q_1, \varepsilon)\}$		$\{(q_2, \varepsilon)\}$			
q_2	$\{(q_2, \varepsilon)\}$		$\{(q_2, \varepsilon)\}$			

State Diagram:



Hence from explanation there exists a turing machine which accepts given language

Hence, C is a CFL.

Scanned with CamScanner

Q 51

Q 33

Use the language $A = \{a^m b^n c^n \mid m,n \geq 0\}$ and $B = \{a^n b^n c^m \mid m,n \geq 0\}$ together to show that the class of context-free language is not closed under intersection

Sol

The two languges are

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

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

Constructing grammar that recognize A

$$S \rightarrow UT$$

$$U \rightarrow aU \mid \varepsilon$$

$$T \rightarrow bTc \mid \varepsilon$$

Observing this we can say language A is a context free language

Similary for B

$$S \rightarrow TU$$

$$T \rightarrow aTb \mid \varepsilon$$

$$U \rightarrow cU \mid \varepsilon$$

By this we can say language B is also context free language

Scanned with CamScanner

So consider $A \cap B = \{a^n b^n c^n \mid n \geq 0\}$

Now using pumping lemma for string s & pumping length p

$$s = a^p b^p c^p \quad ; \; s \in A \cap B$$

Divide s into $uvxyz$ either v or y is non-empty

<u>Case I</u>

If both v & y contains only one type of symbol v does't contains both a's & b's or both b's & c's Here uv^2xy^2z can't contain equal number of a's, b's & c's. Therefore it can't be member of $A \cap B$ which violates I^{st} condition of pumping lemma hence contradiction.

<u>Case II</u>

either v or y contains more than one type of symbol uv^2xy^2z may contain equal number but not in correct order. Hence contradiction

So we alway get a contradiction Hence this assumption of $A \cap B$ is a context-free language is false.

$A \cap B$ is not context-free language but A & B are, so we can say language obtained by intersection of 2 context-free language is not context-free

Therefore A & B are not closed under intersection

Proved

Scanned with CamScanner