BY - AYUSH SHARMA (2019101004)

Q. 1 Prove that :- If L is Regular, $\exists$ an integer p (Pumping length) s.t. $\forall$ $s \in L$, $|s| \geq p$, $s = xyz$ s.t.

(a) $\forall i \geq 0$ $xy^iz \in L$
(b) $|xy| \leq p$
(c) $y \neq \epsilon$.

Sol^n :- The above statement is known as "pumping Lemma". (P.L.).

Proof (of P.L.) :-

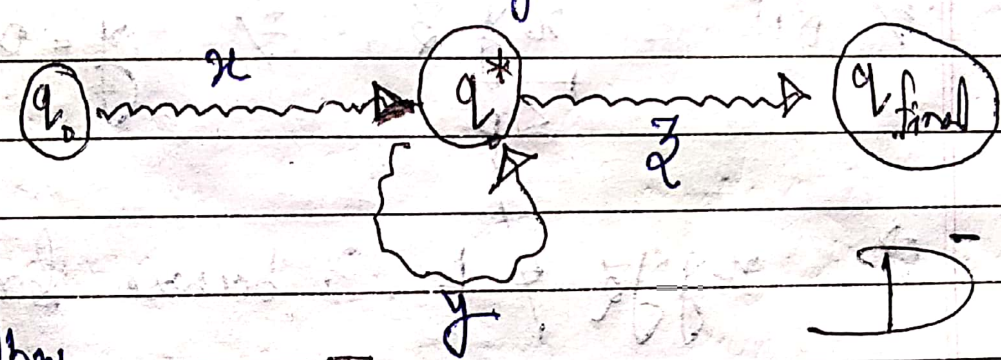
If L is Regular then it has a DFA. Let that DFA $D = \langle Q, \Sigma, \delta, q_0, F \rangle$ s.t. $L(D) = L$.

Consider $p \geq |Q|$. Let $s = s_1 s_2 \dots s_{p \geq |Q|}$ s.t. $s \in L$. Also, $|s| \geq |Q|$.

$$q_0 \xrightarrow{s_1} q_1 \xrightarrow{s_2} q_2 \xrightarrow{s_3} \cdots \xrightarrow{s_{|Q|}} q_{final}$$

Above is the state diagram for DFA possible but since formal definition of DFA contains only 'Q' states & total '$Q+1$' states has been listed in above state diagram, using pigeon-hole principle we can claim, for a particular i & j s.t. q_i & q_j ∈ states in above state diagram, $q_i = q_j = q^*$.

Thus, atleast one state get REPEATS.
We reduce state diagram to :—



Thus, $\boxed{S = xyz}$.

$\Rightarrow \forall i, \; xy^iz \in L$ { Because D ⇒ accept xy^iz.

⇒ As repetion occurs when we read confirmly $|Q|$ symbols (inputs) & $|Q| \le p$

Thus, $|xy| \le |Q| \le p$.

⇒ As there needs to be atleast 1 input for repetion → $|y| \ge 1$.

Thus, $y \ne \epsilon$.

Q.2 $L = \{ a^n b^n \mid n \geq 0 \} \cancel{\in}$ REGULAR. Prove using pumping lemma.

Sol^n → Assume $L \in$ REGULAR then L must satisfy the pumping lemma that means $\exists\, p$ s.t. $\forall\, s \in L$, $|s| \geq p$, $s = xyz \Rightarrow xy^iz \in L$ and $|xy| \leq p$ and $y \neq \epsilon$.

Let us consider 0^p1^p. clearly $s \in L$, $|s| \geq p$.
No way of tri-partioning $s \to xyz$

Because, if $0^p1^p = xyz$ then we can say $y = 0^k$ due to condition that $|xy| \leq p$ and $y \neq \epsilon$ where $1 \leq k \leq p$.

That means $xyy \cancel{\in} L$, because

xyy will be of form 0^x where $p+1 < x \leq 2p+1$, since $xyy = 0^py$.
And for a string a^mb^n to be in the given language, $m = n$ is must which is not true in case of xyy for claimed $s = 0^p1^p$.
Hence, L is not REGULAR.

Q. 3 Show that $L = \{ w \in \{0,1\}^* \mid w$ has equal no. of 0 & 1 $\}$ is not Regular.

Sol^n ⇒ Suppose $L \in$ REGULAR. Then $\exists\, p$ s.t. if $s \in L$, $|s| \geq p$.

Consider $s = 0^p 1^p$, $s \in L$, $|s| \geq p$.

$\Rightarrow 0^p 1^p = \underbrace{xy}_{\leq p} z \Rightarrow y = 0^k$.

$\Rightarrow xyyz \notin L$, because for above xyz, $xyyz$ has more 0s than 1s.

Therefore, $xy^2z \notin L$ for $xyz = 0^p 1^p \in L$.

Hence,

L is not REGULAR.

Q.4 $L = \{ ww \mid w \in \{0,1\}^* \} \notin$ Regular.
Prove it.

Sol^n ⇒ Say, $L \in$ Regular, from pumping lemma, $\exists p \; \forall s \in L$, $|s| \geq p$, $s = xyz$ s.t. $\forall i \geq 0 \;\; xy^iz \in L$, $|xy| \leq p$ & $|y| \geq 1$ i.e. $y \neq \epsilon$.

Consider, $S = 0^p 1 0^p 1$, $|s| = 2p+2 > p$

$S = xyz \Rightarrow y = 0^+$

Then $xyyz = 0^{p+i} 1 0^p 1 \notin L$
Therefore, L is not regular.

Q.5 $L = \{ a^{n^2} \mid n \geq 0 \} \notin$ Regular. Prove it.

Sol^n ⇒ Say $L \in$ Regular, then L satisfy pumping lemma. Consider $s = a^{p^2}$, $|s| \geq p$;
$s \in L$.

Now, $S = xyz = a^{p^2}$

then, $xyyz = a^{p^2 + |y|} = a^{\Delta}$

As, $1 \leq |y| \leq p \Rightarrow 1 + p^2 \leq \Delta \leq p + p^2$
$< (p+1)^2$

Thus, Δ is not a perfect square. Hence, $xyyz \notin L$. Thus, L is not regular.

Q. 6 Show that there are languages that are not regular but satisfy pumping lemma (P.L.).

Soln :- Consider the language :

$L = \{ ab^j c^j \mid j \geq 0 \} \cup \{ a^i b^j c^k \mid i, j, k \geq 0 \wedge i \neq 1 \}$

Showing that it satisfy P.L. :-
We need to show $\exists p$ s.t. if $s \in L$ & $|s| \geq p$ then $s = xyz$, $xy^iz \in L$, $|xy| \leq p$ & $|y| \geq 1$. $\forall i \geq 0$.

Let $p = 2$;

→ $s = ab^jc^j$; $j \geq 1$. We can write $s = xyz$; $x = \epsilon$, $y = a$, $z = b^jc^j$.
⇒
$|xy| = |a| = 1 \leq 2$
$|y| = |a| = 1 \geq 1$
$\forall i \geq 0$, $xy^iz \in L$ i.e. $a^ib^jc^j \in L$.

→ $S = aab^jc^k$; $S = xyz$, $x = \epsilon$, $y = aa$ & $z = b^jc^k$.
⇒ $|xy| = |aa| = 2 \leq 2$
$|y| = |aa| = 2 \geq 1$

& $\forall i \geq 0,\ xy^iz = (aa)^i b^j c^k \in L$.

→ $w = a^r b^j c^k$ with $j, k \geq 0$ & $r \geq 2$.

$s = x, y, z$, $x = \epsilon$, $y = a$, $z = a^{r-1} b^j c^k$.

$|xy| = |a| = 1 \leq 2$, $|y| = |a| = 1 \geq 1$

$\forall i \geq 0,\ xy^iz = a^{i+r-1} b^j c^k \in L$.

→ $w = b^j c^k$; $j \geq 1$ & $k \geq 0$, $j+k \geq 2$.

$s = x, y, z$, $x = \epsilon$, $y = b$, $z = b^{j-1} c^k$.

$|xy| = |b| = 1 \leq 2$, $|y| = |b| = 1 \geq 1$.

$\forall i$, $xy^iz = b^{i+j-1} c^k \in L$.

→ $w = c^k$; $k \geq 2$.

$s = x, y, z$, $x = \epsilon$, $y = c$, $z = c^{k-1}$

$|xy| = |c| = 1 \leq 2$

$|y| = |c| = 1 \geq 1$

$\forall i$ $xy^iz = c^{i+k-1} \in L$.

Thus the language satisfy pumping lemma.

We can prove that this language is not regular by contradiction.

Let L is regular Language described by each regular is also regular. Also, regular language are closed under intersection.

Thus, $L(ab^*c^*)$ is regular. And $L \cap L(ab^*c^*)$ should be regular. But

$\Delta = L \cap L(ab^*c^*) = \{ab^jc^j \mid j \geq 0\}$ which does not follow Pumping Lemma. Because of following :-

Let $p = n$, consider string $ab^nc^n \in \Delta$.

$|ab^nc^n| = 2n+1 \geq (n = p)$.

Partioning; $ab^nc^n = xyz$, only 2 possible.

① $x = ab^r$, $y = b^s$ with $s \geq 1$, $z = b^{n-r-s}c^n$

Now,

$ab^{n-s}c^n = xz \in \Delta$ is must but does not.

② $x = \epsilon$, $y = ab^j$ $(j \geq 0)$ & $z = b^{n-j}c^n$

Now,

$b^{n-j}c^n = xz \in \Delta$ is must but wrong.

In both case $xz \notin \Delta$.

Thus, no partion exist $\forall$ $s \in L \cap L(ab^*c^*)$

& does not satisfy Pumping lemma.

Hence, our assumption that 'L' is regular is wrong.

Q.7 $L = \{ 0^i 1^j \mid i > j \} \notin$ Regular

Prove it.

Soln = Suppose L is regular, then $\exists p$ s.t. $\forall s \in L$, $|s| \geq p$ we can partion of such that $s = xyz$, $xy^iz \in L \; \forall i \geq 0$, $|xy| \leq p$ & $|y| \geq 1$ i.e. $y \neq \epsilon$

Consider $s = 0^{p+1} 1^p \in L$, $|s| \geq p$ because $|s| = 2p+1$.

$s = xyz = 0^{p+1} 1^p \Rightarrow y = 0^+$

$\forall i \geq 0$; $xy^iz \in L$. (claim)

then $xz = 0^{p+1-|y|} 1^p \notin L$

as $|y| \geq 1$. $\Rightarrow p+1-|y| \leq p$,

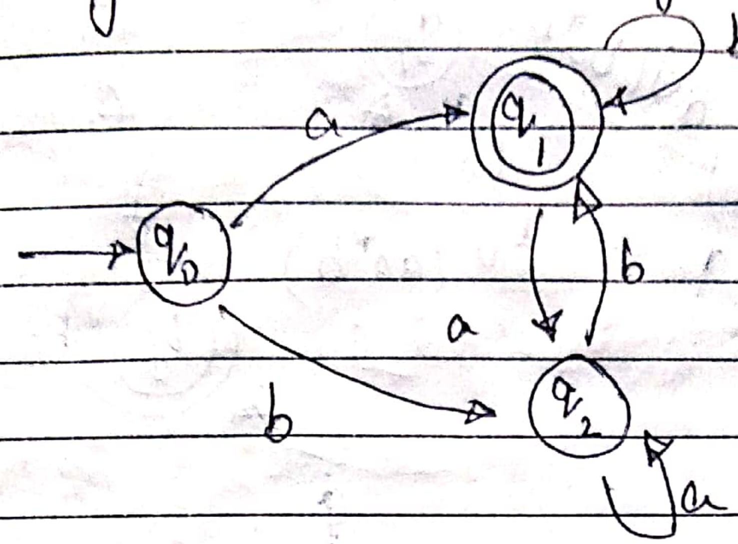
So, number 0 in xz becomes equal or less than number of 1.

Hence, $xz \notin L$,

Therefore, $L \notin$ Regular.

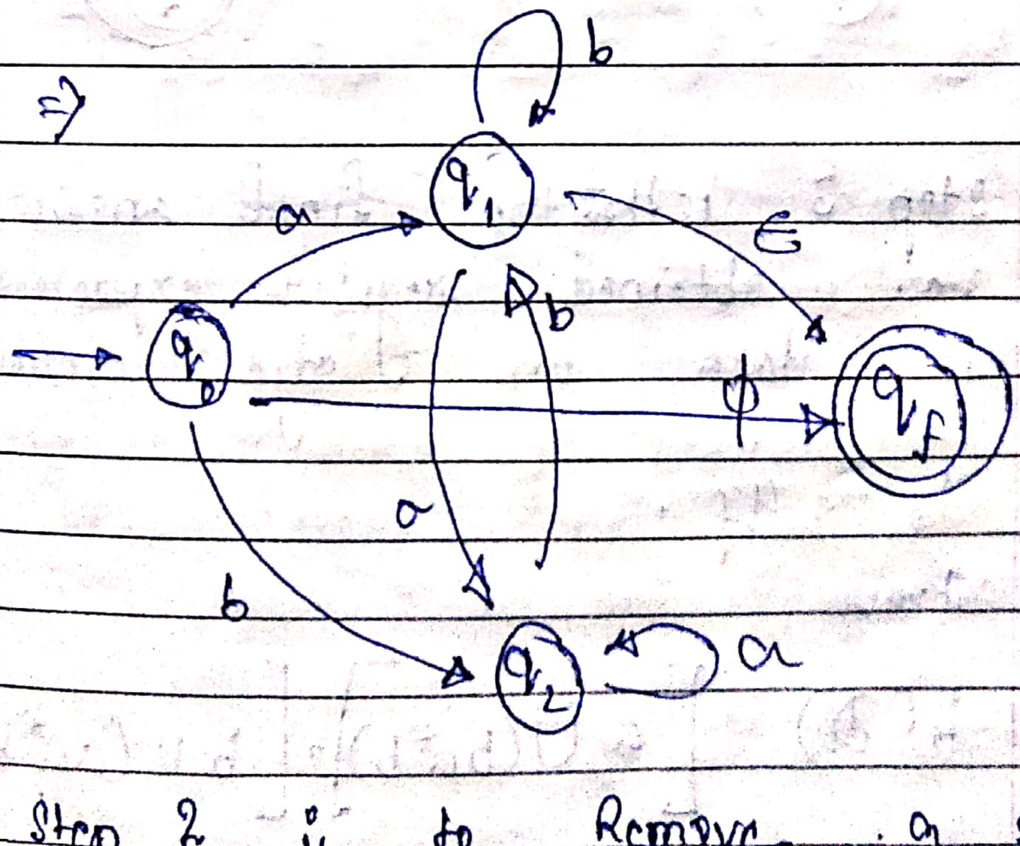
Scanned with CamScanner

Q. 8 Using the concept of GNFA, convert the given DFA to Regular Expression.



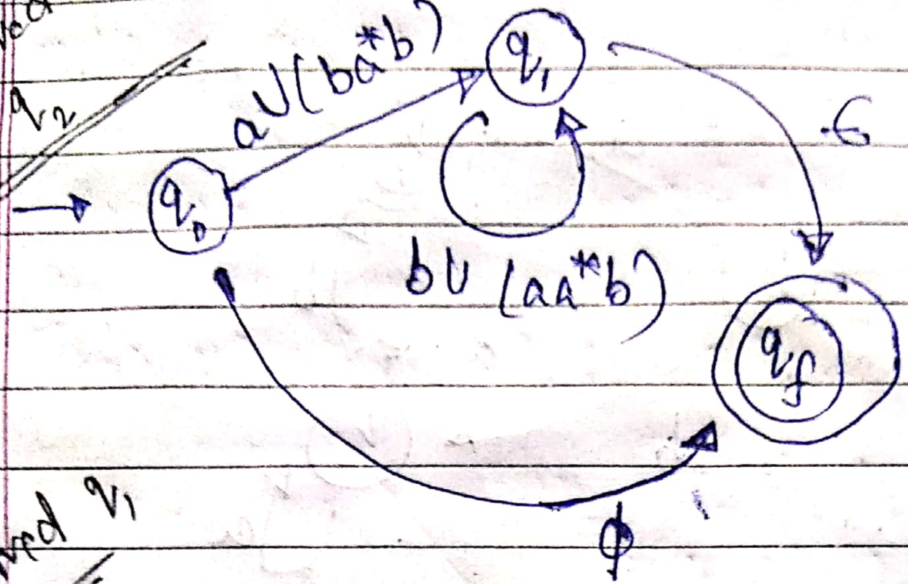
Sol => Step 1 is to convert above DFA to GNFA i.e. 1 special start state, 1 other final state.

=>

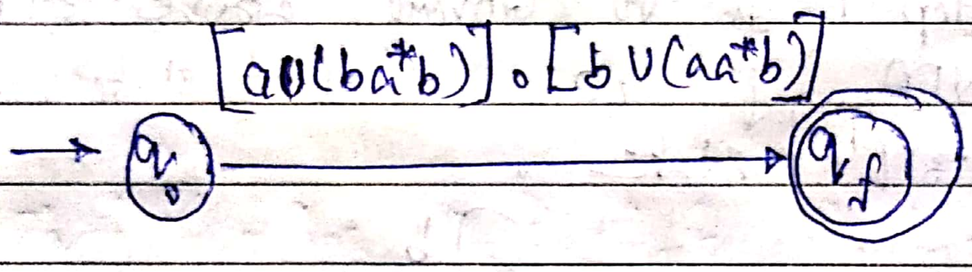


Step 2 is to Remove a state until only q_0 & q_f remain s.t. only 1 edge is maintained between any two state.

Removed q_2



Removed q_1



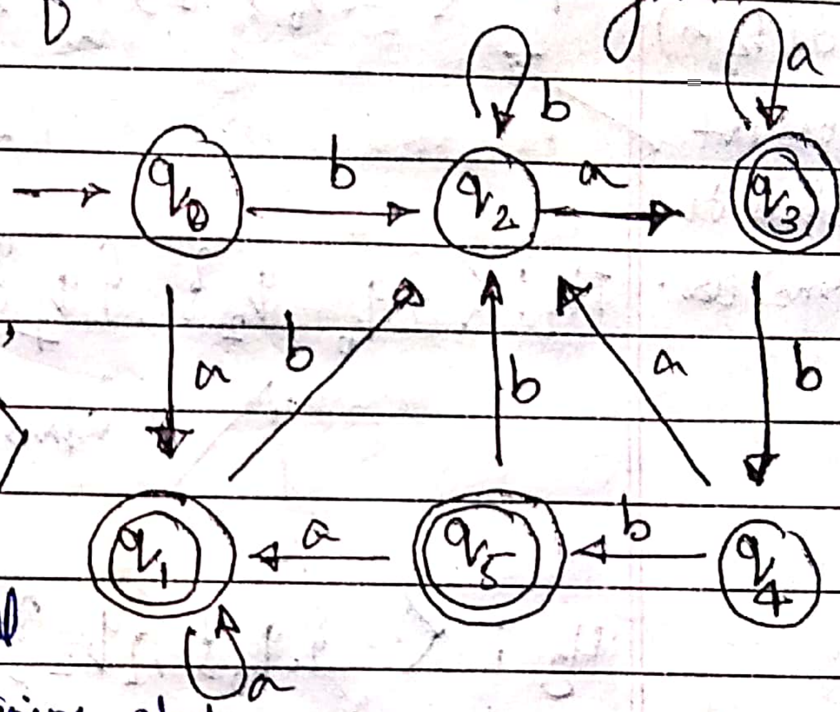
step 3 is that the final answer i.e. obtained regular expression is data on edge connecting q_0 & q_f.

Thus,

$$L(D) = \left[a \cup (ba^*b) \right] \circ \left[b \cup (aa^*b) \right]$$

is regular expression

Q. 9 Find the minimal state DFA for the given state diagram by minizing the number of state in given DFA.



Sol^n = $\Pi_0 = \langle \{q_0, q_2, q_4\},$
$\{q_1, q_3, q_5\} \rangle$

Will divide all state in 2-set, one with all final & other set remaining state.

Sol^n = Partion as Q $\Big\}$ $\Pi_i \longrightarrow \Pi_{i+1}$

We have to terminate when $\Pi_i = \Pi_{i+1}$.

This transition can be done :-

$P = \{q_i, q_j, \ldots\}$ for some $P \in \Pi_i$.

Then if $\exists a \in \Sigma$ s.t. $\delta(q_i, a) \in P_1$ & $\delta(q_j, a) \in P_2$ and $P_1 \neq P_2$.

Divide P s.t. q_i & q_j go to different class.

The whole transition is as follows :-

$\delta(q_0, a)$ & $\delta(q_4, a)$ does not belong to same class.

$\Pi_0 : \langle \{q_0, q_2, q_4\}, \{q_1, q_3, q_5\} \rangle$

$\Pi_1 : \langle \{q_0, q_2\}, \{q_4\}, \{q_1, q_3, q_5\} \rangle$

(for q_1 & q_3)

$\Pi_2 : \langle \{q_0, q_2\}, \{q_4\}, \{q_1, q_5\}, \{q_3\} \rangle$

$\left.\begin{array}{l} \delta(q_0, a) = q_1 \\ \delta(q_2, a) = q_3 \end{array}\right\} \rightarrow$ different class

$\Pi_3 : \langle \{q_0\}, \{q_2\}, \{q_4\}, \{q_1, q_5\}, \{q_3\} \rangle$

$\Pi_4 : \langle \{q_0\}, \{q_2\}, \{q_4\}, \{q_1, q_5\}, \{q_3\} \rangle$

As, $\Pi_3 = \Pi_4$. Hence, final partitioning is Π_3.

Total number of minimum state is 5 for given DFA state diagram.

Q.10 Proove the following :-

If $A = L(M)$ for a DFA M then for any $x, y \in \Sigma^*$ if $x \; R_M \; y$ then $x \; R_A \; y$ where R is equivalence relation. $M = \langle Q, \Sigma, \delta, s, F \rangle$

Soln:- Suppose that $A = L(M)$. Therefore $w \in A \Leftrightarrow \delta^*(s, w) \in F$. Suppose also that $x \; R_M \; y$. Then, $\delta^*(s, x) = \delta^*(s, y)$.

Let $z \in \Sigma^*$. clearly $\delta^*(s, xz) = \delta^*(s, yz)$. Therefore,

$$\begin{aligned} xz \in A &\Leftrightarrow \delta^*(s, xz) \in F \\ &\Leftrightarrow \delta^*(s, yz) \in F \\ &\Leftrightarrow yz \in A \end{aligned}$$

It follows that $x \; R_A \; y$.

Above statement says that whenever two elements arrive at the same state M they are in the same equivalence class of R_M. This means that each equivalence class R_A is union of equivalence classes of R_M.

Nerode -

Q.11 State Myhill-Theorem. And then prove it.

Sol^n :- Nerode -

Myhill-Theorem states that a language 'L' is regular iff the number of equivalence classes in R_L is finite.
Moreover, if finite, the no. of classes is the no. of state in the smallest DFA for L.

* Equivalence Relation R_L : string $x R_L y$ if :

Not [$\exists z$ st exactly one of $xz, yz \in L$] is true.

<u>Proof</u> :- If $L \in$ Regular $\exists$ DFA $D = \langle Q, \Sigma, \delta, q_0, F \rangle$. Consider the following partition of all finite length strings $\langle S_1, S_2, \ldots, S_n \rangle$ where $Q = \{1, 2, \ldots, 3\}$

Basically, $S_i = \left\{ x \;\middle|\; \begin{array}{l} x \text{ takes D from} \\ \text{state } 0 \rightarrow i \end{array} \right\}$

So, for $x \in S_i$ & $y \in S_j \Rightarrow x R_L y$

but for $x \in S_i$ & $y \in S_j$ & $i \neq j \Rightarrow x \cancel{R_L} y$

Thus the discussion above reduce to the fact that if $x \; R_D \; y$ then $x \; R_L \; y$ as discussed in previous question.

Hence, if language is regular then it has finite equivalence classes which will be number of state in smallest DFA.

Now if R_L has finite equivalence classes then L is regular.

Say R_L has class $\rightarrow \langle E_1, E_2, \ldots, E_n \rangle$

Consider states $Q' \longrightarrow \langle 1, 2, \ldots, n \rangle$

We can say,

$\delta(i,a) = j$; if $\exists x \in E_i$ s.t. $xa \in E_j$

q_0 : i s.t. $\epsilon \in E_i$

ϵ ← null string.

F : all i s.t. $\exists x \in L$ & $x \in E_i$.

It is easy to argue that $\delta^*(q_0, x) = j \Leftrightarrow x \in E_j$.

Thus, $L(D) = L$.

Hence, L is regular.

Also at start there are n states in DFA & n eq. classes in L.

Q. 12. Without using Pumping Lemma and Myhill-Nerode theorem, prove the following :-

The language $L = \{ w \,?\, w \mid w \in \{a,b\}^* \}$ is not regular.

* L is the language of all strings consisting of the same strings of a's & b's twice, with a ? symbol in-between.

Sol^n = Suppose D is a DFA for L where D ends in the same state when run on two distinct strings say a^n & a^m. Since D is deterministic, D must end in the same state when run on strings $a^n ? a^m$ & $a^m ? a^n$. If this state is accepting, then D accepts $a^m ? a^n$, which is not in L. Otherwise, the state is rejecting, so D rejects $a^n ? a^m$, which is in L.
Both cases contradict that D is

a DFA for L. so our assumption was wrong. Thus D ends in different states.

Proof :- Assume for the sake of contradiction that L is regular. Since L is regular, there must be a DFA D for L. Let n be the number of states in D, when D is run on the strings $a^0, a^1, \ldots, a^m$, by the pigeonhole principle since there are n+1 strings & n states, at least two these strings must end in the same state. Because of the result proven, this is impossible.

Contradiction happens, hence L is not regular.

Q. 13 Prove the following using Myhill-Nerode Th. :-

$L = \{ w \overset{?}{\pm} w \mid w \in \{a,b\}^* \}$ is not regular.

Soln ⇒ Consider $S = \{ a^n \mid n \in \mathbb{N} \}$.
S is infinite because it contains one string for each natural number.

Now consider a^n & $a^m \in S$, where $a^n \neq a^m$. then $a^n \overset{?}{\pm} a^n \in L$ &

$a^m \overset{?}{\pm} a^n \notin L$. so a^n & a^m are distinguishable relative to L i.e.

$a^n \not R_L a_m$. leading infinite

no. of Equivalence class. Therefore as Myhill-Nerode theorem requires necessary & sufficient condition both of finite Equivalence class which is not the case.

Hence, L is not regular.

Q.14- $L = \{a^n \mid n$ is power of $2\}$. Prove that L is not regular using Myhill Nerode Theorem. (MHT).

Sol^n ⇒ Let $S = \{a^{2^n} \mid n \in N\}$. This set is infinite because it contains one string for each positive natural number. Let $a^{2^n}, a^{2^m} \in S$ be any two strings in S where $a^{2^n} \neq a^{2^m}$. Assume without loss of generality that $n < m$, so $1 \leq n < m$.

Consider, the strings $a^{2^n} a^{2^n}$ & $a^{2^m} a^{2^n}$.
The string $a^{2^n} a^{2^n}$ has length 2^{n+1}, which is a power of two.
However, string $a^{2^m} a^{2^n}$ has length of $2^m + 2^n = 2^n(2^{m-n}+1)$.
Since, $m > n$, we know $2^{m-1}+1$ is odd; So $2^m + 2^n$ is not power of two.
Thus, $a^{2^m} a^{2^n} \notin L$. So, $a^{2^m} \not R_L\, a^{2^n}$
This leads to infinite number of equivalence classes, hence, by MHT, L is not Regular.

Scanned with CamScanner

Q.15 = For given language find minimum pumping length 'p' with proper justification :-

(a) 0001^* (b) $0^* 1^*$ (c) $0^* 1^+ 0^+ 1^* \cup 10^* 1$
(d) 1011 (e) ε

Sol = (a) $p \neq 3$ because 000 being in the language cannot be pumped. Thus, if $s \in L(0001^*)$, $|s| \geq 4$, $s = xyz$ where $x = 000$, $y = 1$ & $z = 1^*$ satisfies every pumping condition. Thus $P_{min} = 4$.

(b) $P \neq 0$ because $\epsilon \in L(0^* 1^*)$ & ϵ cannot be pumped. Every non-empty string in the language can be divided into xyz where $x = \epsilon$, y is the first character & z is remainder. Hence, $P_{min} = 1$.

(c) $P_{min} = 3$. Because $P_{min} \neq 2$ as string $11 \in$ Language, & cannot be pumped. If s is generated by $0^* 1^+ 0^+ 1^*$, it can be written as xyz, where $x = \epsilon$, y is first symbol & z is remainder for $|s| \geq 3$. If s is generated by $10^* 1$ & $|s| \geq 3$

we can write $s = xyz$ where $x = 1$, $y = 0$ & z is the remainder. This way we can pump s.
Thus, $P_{min} = 3$.

(d) If s be a string language then $s = 1011$ only.
If we set $P_{min} = 4$, then claim s is pumpable (which is not, as it is the only string in the language).
This should be 5.
Hence, $P_{min} = 5$.

(e) Let s be a string in the language then s is ε & acc. to pumping lemma it cannot be pumped. As per the pumping lemma, pumping length should greater 0.

Hence, $P_{min} = 0$.

Scanned with CamScanner

Q.16 = If A is any language, let $A_{\frac{1}{3}-\frac{1}{3}}$ be the set of all strings in A with their middle thirds removed so that

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

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

Sol^n = As language described each regular is also regular. Hence, $L(a^* c^*)$ is regular.
Now consider $A = a^* b + c^*$.
A is regular language.
Also we know regular language are closed under intersection. So.
Taking intersection :-

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

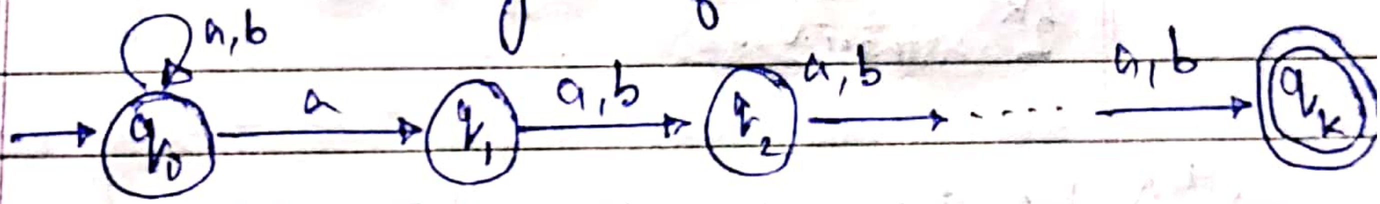
As we have proved many times than $\{a^n c^n\}$ is not regular. Hence, $A_{\frac{1}{3}-\frac{1}{3}}$ is not regular. Thus,

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

Q.17 = Let $\Sigma = \{a, b\}$. For each $k \geq 1$, let C_k be the language consisting of all strings that contain an 'a' exactly k places from the 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 a state diagram & a formal description.

Sol^n => The state diagram of NFA is as follows:—



The formal description of NFA N :—

$N = \langle Q, \Sigma, \delta, q_0, F \rangle$

Q = set of states = $\{ q_i \mid i = 0, 1 \ldots, k \}$

$\Sigma = \{a, b\}$ = set of alphabets.

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

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

δ = Transition function :—

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

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

$$\delta(q_i, \epsilon) = \phi \;\; \forall i = \{0,1,2,\dots,k\}.$$

Q.18 ⇒ Consider the language C_k in Q.17. Prove that for each k, no DFA can recognize C_k, with fewer than 2^k states.

Sol^n ⇒ We have to prove that minimal states in DFA to recognise C_k is 2^k. We know,

$$\delta^*(q_0, lz) = \delta^*(q_0, mz) \Leftrightarrow \delta^*(q_0, l) = \delta^*(q_0, m)$$

for a DFA $D = \langle Q, \Sigma, \delta, q_0, F \rangle$.

Consider, $s_1 = x_1 x_2 \dots x_k$ s.t. $|s_1| = k$ & $s_1 \in \Sigma^*$. Similarly consider $s_2 = y_1 y_2 \dots y_k$ s.t. $|s_2| = k$ & $s_2 \in \Sigma^*$.

We know, $x_i = a$ or b and

$y_i = a$ or b

$\forall i = \{1, 2, \dots k\}$

Let $x_i \neq y_i$ for some i then. Either S_1 or S_2 contains a at i^{th} position & other will have b at i^{th} position.

We can say $S_1 z \not\equiv_D S_2 z$ for $z = \{a \cup b\}^{i-1}$.
because, either the string $S_1 z$ or $S_2 z$ has the k^{th} bit from the end as 'a'.

Therefore, total no. of equivalence classes can said to be 2^k i.e. all possible no. of string of length k.

Using Myhill-Nerode theorem, no. of state in minimal possible DFA i.e. smallest DFA is = no. of equivalence class in corresponding language.

So, minimum state for DFA = 2^k.

Hence, no DFA can recognize C_k with fewer than 2^k states.

Q.19 = Let N be an NFA with k states that recognizes some language A. Show that, if A is non-empty, A contains some string of length at most k.

Sol^n = Let $N = (Q, \Sigma, \delta, q_0, F)$ with k states. s.t. $L(N) = A$.

Suppose A is non-empty.

→ Then there must be an accept state $q \in F$ that can be reached from the start state q_0.

→ Let w be the string that can be accepted by N when travelling along the shortest path q_0 to q. Let n be the length of w.

→ Then, the sequence of state $q_0 q_1 \cdots q$ in shortest path has length '$n+1$'.

→ Note, all state q_1 to q must be distinct, otherwise there will shorter path by removing a repeating state.

→ As $|Q| = k$ $\Rightarrow$ $n \le k$.

→ Thus, we can say w is accepted by N because q is an accept state.

→ Hence, A contains a string of length at most k.

Scanned with CamScanner

8.20 = Prove that, for each $n > 0$, a language B_n exist where
(a) B_n is recognizable by a NFA that has n states
(b) if $B_n = A_1 \cup \cdots \cup A_k$, for regular language A_i, then at least one of the A_i requires a DFA with exponentially many states.

Sol^n => (a) We will do proof by Induction:—

Base case: $n=1 \Rightarrow B_n = \{ \epsilon, 0, 1 \} = B_1$

Then, formally NFA can be

$N = \langle \{q_0\}, \Sigma, \delta, q_0, \{q_0\} \rangle$ s.t.

$L(N) = B_1$, where $\delta(q_0, \epsilon|0|1) = q_0$.

Proof :— As in base case we can see B_1 can be recognised by NFA with only 1 state.

Let assume B_n can be recognised by NFA with 'n' state.

We have to prove for B_{n+1}.

So, we can write $s \in B_{n+1}$ as

$S = b_1 b_2 \ldots b_n b_{n+1}$

As regular lang. is closed under the concatenation, so, we can break S into $(b_1 b_2 \cdots b_n) \circ (b_{n+1}) \equiv R_1 \circ R_2$

As due to inductive hypothesis R_1 can be recognized by NFA with 'n' state & R_2 is our base case which can be recognised by NFA with 1 state. We can concatenate both NFA (due to closure property).

And final no. of state in N obtained will be '$n+1$'.

Hence, $\exists$ NFA N s.t. $L(N) = B_{n+1}$ and states in $N = n+1$.

(b) $B_n = A_1 \cup A_2 \cup \cdots \cup A_k$, A_i is regular.
If DFA is constructed which is equivalent to the DFA of given NFA.
There could be at least n & at most 2^n states in the resultant equivalent DFA.
$\forall i \in \{1, 2, \cdots, k\} \rightarrow \exists$ DFA M s.t. $L(M) = A_i$. Now using pigion hole principle, we can say that there is atleast one DFA that requires 2^i state to recognise A_i.

Scanned with CamScanner

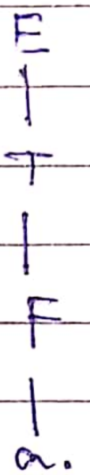
Q.21 For the given CFG, give parse tree & derivations for each string.

$$E \longrightarrow E + T \mid T$$
$$T \longrightarrow T \times F \mid F$$
$$F \longrightarrow (E) \mid a$$

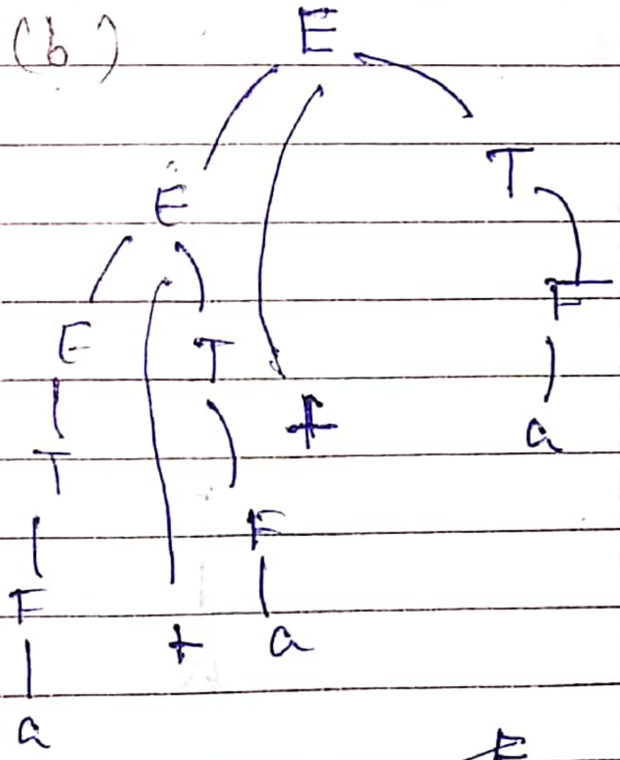
(a) → a (b) → a+a+a

(c) → a+a (d) → ((a))

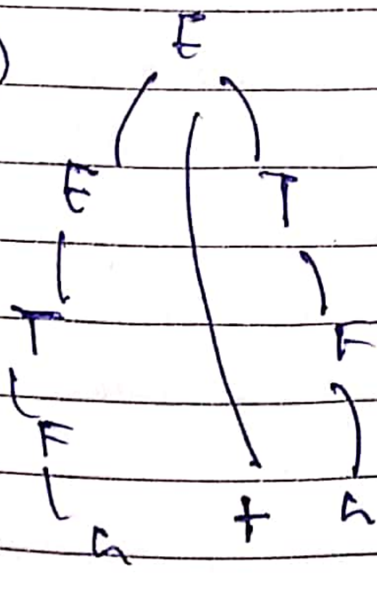
Sol → (a)



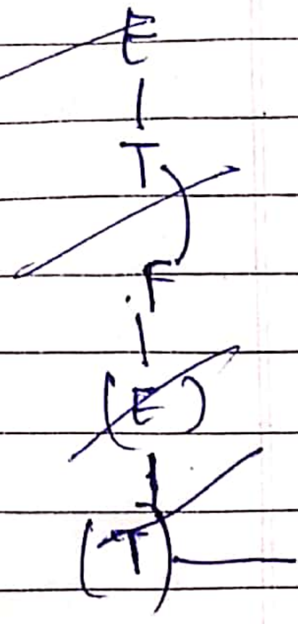
(b)



(c)

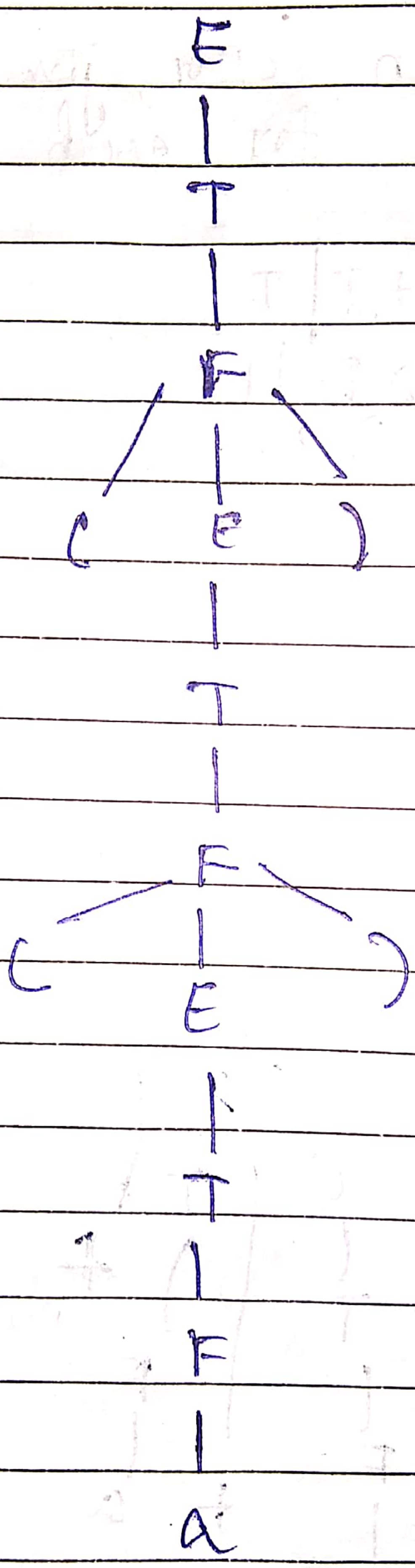


(d)



Scanned with CamScanner

(d)



Q.22 Answer each part for the following CFG :-

$R \rightarrow XRX \mid S$

$S \rightarrow aTb \mid bTa$

$T \rightarrow XTX \mid X \mid \varepsilon$

$X \rightarrow a \mid b$

(a) Write variable, terminals & start variable of G.
(b) Give 3 strings in L(G).
(c) $T \Rightarrow aba$ is true or not.

Solⁿ:- (a) Variables = R, S, T & X
Terminals = a, b
start variable = R

(b) Case 1: Consider the rule $R \rightarrow S$.
Substitute S with rule $S \rightarrow aTb$
$R \rightarrow aTb$
Substitute T with rule $T \rightarrow \varepsilon$
$R \rightarrow a\varepsilon b$, $R \rightarrow ab$

Case 2: Consider the rule $R \rightarrow S$
Substitute S with rule $S \rightarrow bTa$
$R \rightarrow bTa$

Substitute T with rule $T \to \varepsilon$

$R \to b\varepsilon a$

$R \to ba$.

Case 3: Consider the rule $R \to S$

substitute S with rule $S \to aTb$

$R \to aTb$

substitute T with rule $T \to X$.

$R \to aXb$

substitute X with rule $X \to a$.

$R \to aab$

Therefore, the 3 strings in $L(G)$ are ab, ba & aab.

(c) False.

$T \to aba$, the string cannot be derived using G.

Q.23= Give context free grammars generating the following language :-

(a) The set of strings over the alphabet $\{a,b\}$ with more a's than b's.

(b) $\{ w\#x \mid w^R$ is a substring of $w, x \in \{0,1\}^* \}$

Sol^n ⇒ (a) We can say the context free grammar is :—

$S \rightarrow Aa \mid BS \mid SBA$
$A \rightarrow Aa \mid \varepsilon$
$B \rightarrow \varepsilon \mid BB \mid bBa \mid aBb$

In the above grammar S will generate all strings with as many a's as b's. R forces an extra 'a' which gives the required strings of the language.

(b) We can say context free CFG is :—

$R \rightarrow SX$
$S \rightarrow 0S0 \mid 1S1 \mid \#X$
$X \rightarrow XX \mid 1 \mid 0 \mid \varepsilon$

The non-terminal S ends only with #X. S must generate a string whose beginning & end are mirror images.

Scanned with CamScanner

Since, X generate $(0 \cup 1)^*$ the symbol S generates all strings of the form $w \# (0 \cup 1)^* w^R$. The above grammar generates all the substrings of X for $w, x \in \{0,1\}^*$.

Q.24 ⇒ Give a context free grammar that generates the 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 ambiguous? why or why not?

Sol^n ⇒ The language A can be split into 2 languages which are defined as follows :-

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

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

Using the language A_1 & A_2 we can construct a CFG for A_1 & A_2.

The grammar for language A

is the union of grammar of two language which is defined as follows :-

$$S \rightarrow S_1 | S_2$$

In the language A_1 the values of i & j are equal so there must be equal number of a's & b's in the language A_1.

CFG for language A_1 is as follows :-

$$S_1 \rightarrow S_1 c | E | \varepsilon$$

$$E \rightarrow aEb | \varepsilon$$

Similarly, lang. A_2 the values of j & k are equal so there must be equal number of b's & c's in A_2.

CGF for the lang. A_2 is as follows :-

$$S_2 \rightarrow aS_2 | F | \varepsilon$$

$$F \rightarrow bFc | \varepsilon$$

Since, for generating a string $w = a^n b^n c^n$ using the lang. A either S_1 or S_2 can be used. Thus, the CFG for language A is ambiguous.

Scanned with CamScanner

Q.25 = Give an informal description of a pushdown automaton that recognize the language A in Q.24.

Solⁿ ⇒

We can say $A = A_1 \cup A_2$, where

$A_1 = \{a^i b^i c^k \mid i,k \geq 0\}$ & $A_2 = \{a^i b^k c^k \mid i,k \geq 0\}$

The informal description of PDA that recognize lang. A_1. In more detail it operates as follows :—

→ Read & push a's.
→ Read b's while popping a's.
→ If b's finish when stack is empty, skip c's on input & accept.

The informal description of PDA for A_2 :— In more details it operates as follows :—
Skip a's on input.
Read & push b's.
Read c's while popping b's.
If c's finish when stack is empty, accept.

The informal description of the PDA that recognize the language A is the combination of both the language A_1 & A_2

In more details, it operates as follows :-

- Non-deterministically branch to either step 2 or step 6.

- Read & push a's.
- Read b's while popping a's.
- If b's finish when stack is empty, skip c's on input & accept.

- Skip a's on input.
- Read & push b's.
- Read c's, while popping b's.

- If c's finish when stack is empty, accept.

Q.26 = Show that if G is a CFG in chomsky Normal form, then for any string $w \in L(G)$ of length $n \geq 1$, exactly $2n-1$ steps are required for any deviation of w.

Sol^n => Given that G is a CFG in chomsky Normal form (CNF). The length of string $w \in L(G)$ is $n \geq 1$ for the string w. It is required to show that exactly $2n-1$ steps are required for the derivation of string w. It can be proved applying the induction method by on the string w of length n.

For $n=1$; consider a string "a" of length 1 in CNF, so the valid derivation for this will be $S \rightarrow a$, where $a \in \Sigma$ & S is start symbol.

The no. of steps can be obtained as follows :-

$2n-1 = 2(1)-1) = 2-1 = 1$.

Hence, it is true that $2n-1$ (for $n=1$) steps are required to ~~solve~~ derive string "a".

For $n=k$: Take a string of length $k \geq n$ in CNF, so valid derivation for this will take $2k-1$ steps. The no. of steps are :-

$2n-1 = 2(k)-1 = 2k-1$.

Assume a string of length at most $k \geq 1$ terminal symbols & it has a string of length $n=k+1$ is in CNF.

Since $n>1$, consider a language as follows in CNF where derivation start with start symbol 'S'.

$S \to BC$, $B \xrightarrow{*} X$, $C \xrightarrow{*} Y$

So length of the string starting with start symbol S is $|w| = xy$ where $|x| > 0$ & $|y| > 0$.

Using the inductive hypothesis, for the above languages in CNF the length of any derivation of string w must be

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

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

Hence, $n = |x| + |y|$.

Since,

$B \rightarrow^* x$ has a length of $|x|$ & $C \rightarrow^* y$ has a length of $|y|$.

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

Q.27 = Let $C = \{ x\#y \mid x, y \in \{0,1\}^* \ \& \ x \neq y \}$. Show that C is a context-free language.

Sol^n ⇒ Given that a string $x\#y$ is in language C if & only if $x \neq y$ or strings x & y vary at some specific position. Such as for i-index value of x is different from the character value of y.

It is very easy to form context free grammar which produce all the strings of the form $x\#y$ with $x \neq y$. CFG grammar is as follows:-

$S \to A\#B \mid B\#A$

$A \to TAT \mid 0$

$B \to TBT \mid 1$

$T \to 0 \mid 1$

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

Hence, it is proved that C is context free language.

Q.28 = Give unambiguous CFGs for language L = { w| in every prefix of w the language of a's is at least the number of b's }

Solⁿ = The input alphabet a is prefix, at least the number of b's that means if the length of the input string is one then that must be a. For the two or more inputs the possible string are aa, ab, aaa, aab & so on. So, the unambiguous grammar (CFG) is given below.

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

The production for a regular expression aaab :-

$$S \rightarrow aS$$
$$\rightarrow aaS$$
$$\rightarrow aaaS$$
$$\rightarrow aaab.$$

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

Q.29= Use the pumping lemma to show that the following language is not context free.

$$L = \{0^n 1^n 0^n 1^n \mid n \geq 0\}$$

Solⁿ = Consider the language $B = \{0^n 1^n 0^n 1^n \mid n \geq 0\}$

Let p be the pumping length of B given by pumping lemma. To show that B is not CFL, it is enough to show that a string $S = 0^p 1^p 0^p 1^p$ cannot be pumped.

Consider S is of the form uvxyz.

If both v & y contain at most one type of alphabet symbol, the string will be of the form uv^2xy^2z runs of 0's & 1's of unequal length. Hence the string s cannot be a member of B.

If either v or y contains more than one type of alphabet symbol, the string will be of the form uv^2xy^2z which does not contain the symbols in correct order. Hence, the string s cannot be member of B.

Since the string s cannot be pumped without violating the pumping lemma condition. B is not a CFL.

Q.30 = Say that a language is prefix-closed if all prefix of every string in the language are also in the language. Let C be an infinite, prefix-closed, context free language. Show that C contains an infinite regular ~~sheet~~ subset.

Soln => Consider a language L which is context free & prefix closed. As the language L is context free & therefore, the principle of pumping lemma can be applied to it. Let the length of the pumping lemma be P. & the strings that needs to be considered in the language is S. The length P is lesser or smaller than length of the string. Now it is possible to break the string into 'abcde' s.t. $ab^k cd^k e \in L$ $\forall k \geq 0$ & $|bd| \geq 1$.

Now it is already given that L is prefix closed & all the prefixes of the string S are also present in L. Thus, $ab^k \in L$ $\forall k \geq 0$. Thus it can be inferred that the language formed from the ab^* is regular & it is subset of L. Also, if $b \neq \epsilon$ then it implies ab^* is an ∞ regular subset of L and thus it proves the required statement.

Q.31 ⇒ Let $\Sigma = \{0,1\}$ and let B be the collection of strings that contain atleast one 1 in their second half.

$$B = \{ uv \mid u \in \Sigma^*, v \in \Sigma^* 1 \Sigma^* \text{ \& } |u| \geq |v| \}.$$

Give a CFG that generate B.

Solⁿ ⇒ The CFG for the language B is following :-

$S \rightarrow UV$

$U \rightarrow AB$

$V \rightarrow A1A \mid A1B \mid A1U \mid B1U \mid U1U$

$A \rightarrow 00^+ \mid \varepsilon$

$B \rightarrow 11^* \mid \varepsilon$

Explanation :-

- The string S is concatenation of U & V.
- The string U may consist of any no. of 0's & 1's.
- The string V may consist of atleast one ~~zero~~ ~~or~~ one or only one 1.
- The string A may consists of atleast one zero & more than one zeros.
- The string B may consist of atleast one 1 or more than one 1's.

Q.32 ⇒ We defined the rotational closure of language A to $RC(A) = \{yx \mid xy \in A\}$ show that the class of CFLs is closed under rotational closure.

Soln ⇒ CFLs is closed under rotational closure.
As the class of CFLs is closed under concatenation operation. So if L_1 & L_2 are CFLs then the language $L = L_1 \cdot L_2$ is also CFL. Converse is also true. If $L = L_1 \cdot L_2$ is CFL then L_1 & L_2 are also CFL.
Consider $s = xy$ in A. It can be formed by concatenation of two languages X & Y s.t. $x \in X$ & $y \in Y$.
As the language $A = X \cdot Y$ is a CFL, the languages X & Y will also be CFL.

The rotational closure of the string $s = xy$ will be $RC(s) = RC(xy) = yx$. It can be formed by concatenating Y & X.
$RC(s) = Y \cdot X$
As both Y & X are CFL, the language RC(s) is CFL for any string $s \in A$. The rotational closure has been proven for class of CFLs.

Q.33 ⇒ Show that every DCFG is an unambiguous CFG.

Soln ⇒ An ambiguous grammar is defined as "a context-free grammar" for which there exists a string & that string may contains greater than one left-most derivation. An unambiguous grammar is defined as "a context free grammar" for which all justifiable string has an individual leftmost derivation.

→ As CFG's is proper superset of deterministic CFG's i.e. DCFG's & It can be derived from DFA & it can be used to general DCFL.

→ DCFG always shows an unambiguous behaviour & an unambiguous CFG's is an important super class of DCFGs.

The above statement can be proved in following way :-

As it is known that for every pushdown automaton M ∃ an equivalent CFG G.

Therefore, M recognizes L\$ ⇒ G generates L\$.

But M deterministic ⇒ G is an unambiguous. Hence, replacing \$ by ε in G ⇒ G generates L.

Thus, from the above explanation it can be said that every DCFG is an unambiguous CFG.

Q.34 → show that every DCFG generates a prefix-free language.

Solⁿ ⇒ DCFG (deterministic CFG) is a subset of CFG and are derived from deterministic pushdown automata (variation of pushdown automata) for generating deterministic CFG.

Prefix free language refer to language in which any of it member doesnot have any prefix. By contradiction it can be proved that DCFG always generates prefix-free language. It is given below:-

Consider there are 2 string w & wz in $L(G)$, $|w| \neq |wz|$, $L(G)$ is language of grammar & G is DCFG. As both strings are valid, handles of them will exist. Both can be written as:-

$w = xhy$ & $wz = xhz = xh\hat{z}$

h refers to handle of w.

Now consider v & vz valid strings as first reduced step of w & wz. The process is continued till S_1 & $S_1 z$ where S_1 refers to start variable. Since, S_1 never appears at the right side, so $S_1 z$ cannot be reduced. This leads to contradiction.

Q. 35 ⇒ Let $B = \{a^i b^j c^k \mid i,j,k \geq 0$ & $i=j$ or $i=k\}$
Prove that B is not a DCFL.

Soln ⇒ The following fact can be used to proof B is not a DFCL.

→ B is context free.

→ If B is deterministic context free then $\overline{B} = \{a,b,c\}^* - B$ is deterministic context free.

→ $B_1 = \{a^i b^j c^k \mid i \neq j$ & $j \neq k\}$ is not context free.

Here it is recorded that $\overline{B} \cap a^*b^*c^* = B_1$. Suppose that $\overline{B}$ was a DCFL implying that $\overline{B}$ must be a deterministic context free language. which in turn implies that $\overline{B}$ is CFL. As the intersection of a CFL & regular lang. is CFL. However, B_1 is not CFL that shows a contradiction.

Therefore, B is not deterministic context free language i.e. DCFL.