Q-1

(a) consider alphabets $\Sigma = \{0, 1, \$\}$

$$L = \{x\$y \mid x, y \in \{0,1\}^*, |x| = |y|\}$$

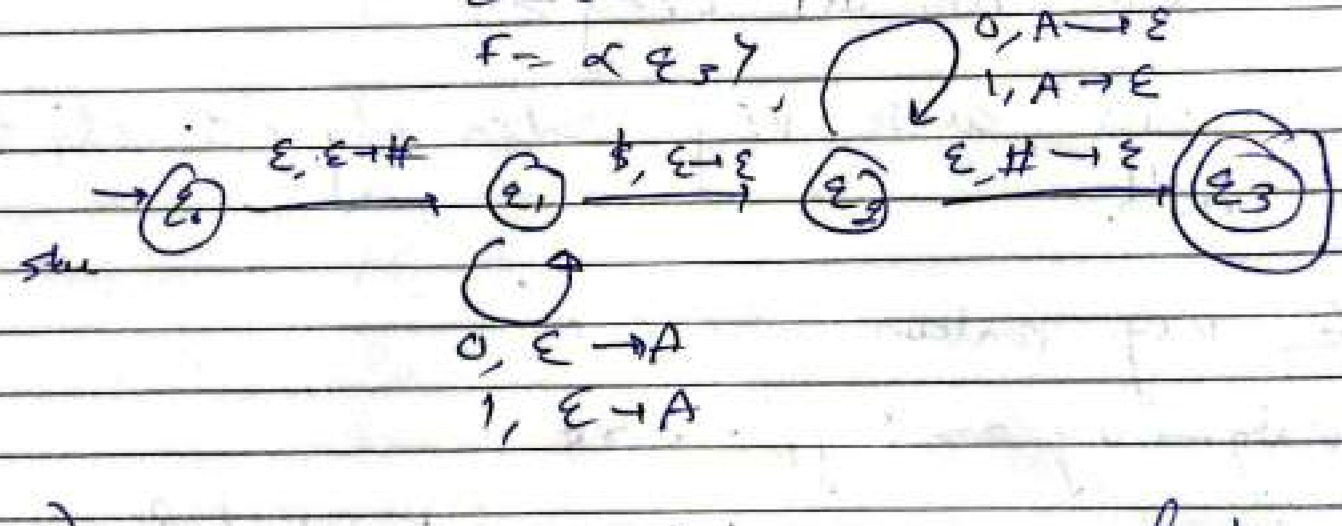
PDA $M = \{Q, \Sigma, \Gamma, \delta, q_0, F\}$

where $\Sigma = \{0, 1, \$\}$, $\Gamma = \{A, \#\}$

$Q = \{q_0, q_1, q_2, q_3\}$

$q_0 = q_0$

$F = \{q_3\}$



(b) The language that we are looking for

$$L_2 = \{ww \mid w \in (0,1)^*\}$$

So, the required concatenation of a & b should be accepted by L_2

Camlin

Scanned with CamScanner

(6) No, we cannot design PDA for L_2

We will try to prove that L_2 is not a CFL by using contrapositive of pumping lemma.

Consider $w = 0^p 1^p 0^p 1^p$ where $p > 0$.

Clearly $w \in L_2$ & $|w| > p$.

Consider $w' = uv^i xy^i z$

- Partition w as $uvxyz$.

three case arise vxy contain only 0's, only 1's or both.

C-1 vxy contain only 0's.

Clearly $vy = 0^k$ for $k > 0$

This implies $uxz = 0^{p-k} 1^p 0^p 1^p$

or $0^p 1^p 0^{p-k} 1^p$

In first case, if it is divided in to two equal length part, one ends with 0 & other 1, hence not equal

In second case: Right half have less zeroes. So not equal.

C-2 vxy only comb. of 1's

This is similar to C-1.

$vy = 1^k$ for $k>0$

$uxz = 0^p 1^{p-k} 0^p 1^p$ or $0^p 1^p 0^p 1^{p-k}$

- first case if divided into two parts ends with different alphabets 0 & 1
- Second case is Number of 0's in left half & right are differ.

So $w' \notin L_2$

C-3. vxy has both 0's and 1's

In $w' = uv^i xy^i z$, set $i=0 \Rightarrow w' = uxz$

- Assume first boundary b/w 0's & 1's lies in vxy.
- let $vy = 0^{k_1} 1^{k_2}$

$w = 0^{p-k_1} 1^{p-k_2} 0^p 1^p$

- Left Half $= 0^{p-k_1} 1^{p-k_2} 0^{(k_1+k_2)/2}$
- Right Half $= 0^{p-(k_1+k_2)/2} 1^p$

$0 < k_1 + k_2 < p$

Both halves end with different alphabets.

Let boundary lies b/w ~~1's & 0~~ 1's & 0's

Let $v_y = 1^{t_1} 0^{t_2}$

$\rightarrow w' = 0^p 1^{p-t_1} 0^{p-t_2} 1^p$

~~left half $= 0^p 1^{p-(t_1-t_2)/2}$ if $(t_1$~~

left half $= 0^p 1^{p-(t_1+t_2)/2}$ if $t_1 \le t_2$

$0^p 1^{p-t_1} 0^{(t_1-t_2)/2}$ if $t_1 > t_2$

Right half $= 1^{(t_2-t_1)/2} 0^{p-t_2} 1^p$

if $t_1 \le t_2$

$= 0^{p-\frac{(t_2-t_1)}{2}} 1^p$ if $(t_1 > t_2)$

$0 < k_1 + k_2 \le p$

This implies left half & Right half

cannot be equal. $w' \notin L_2$

Camlin

Scanned with CamScanner

if their is any other boundary, arguments are similar.

L_2 is non-CFL by pumping lemma.

$\boxed{Q-2}$ $L = \{a^n b^j c^k \mid k = jn \text{ and } j, k, n \geq 0\}$

T.P. L is not Context Free

We will try to use pumping lemma for context free language to prove L is not context free.

We assume that L is a context-free language. Consider the string when $n = p$, $n = j = p$ & $k = jn = p^2$ where $p > 0$

let string w be $a^p b^p c^{p^2}$.

$|w| \geq p$ & $w \in L$, we divide the string w into 5 parts uvxyz, which satisfies the following conditions:

(i) $|vxy| \leq p$

(ii) $|vy| > 0$ v & y cannot be empty

(iii) $\forall i \geq 0$ $uv^i xy^i z \in L$

From the following above condition, we can observe that vxy cannot contain a & c together as minimum b's between them are p

so vxy can contain a & c, b & c or only b.

Camlin

Consider condition (iii) & put $i=0$, we will get uv^0xy^0z.

Now we consider all the cases one by one.

Case-1 : vxy only contain b. $|vy|>0$

So $uxz = a^p b^r c^p$

$p\,r \neq p$ $\quad uxz \notin L$.

Case-2 : vxy contain a & b. or only a.

$|vy|>0$ (condition ii)

So ~~vxy~~ vy either contain some a's or b's or both

So in uxz, c remain p

but a or b or both are reduced

So $uxz \notin L$.

Case-3 : vxy contains b & c or only c.

Similar to Case-2.

$|vy|>0$

So vy either contain some b's or c's or both.

P.T.O.

Camlin

So, let string uxz be $a^p b^{p-i}$

$< p^2 - m$

$0 < i + m \le p,\ i \ge 0$
$n \ge 0$

To $uxz \in L$, $p(p-i) = p^2 - m$,

m must be of form ip, which means multiple of p.

but $0 < i + m \le p$

$m = 0p, 1p$

two cases

(i) $m = 0 \rightarrow i = p$.

$p(p-i) = 0$

$p^2 - 0 = p^2$

So $uxz \notin L$.

(ii) $m = p \rightarrow i = 0$

$p^2 \neq p^2 - p$

so $uxz \notin L$.

So by pumping lemma, so L is not a context free Language

Camlin

Q-3

(a) CFG for $L = \{a^i b^j c^k \mid i = j$ or $i \neq k$, where $i, j, k \geq 0\}$

To ~~defi~~ design CFG for the Language L, we will try to solve two different problems:

(i) $i = j$

(ii) $i \neq k \rightarrow i > k$ or $i < k$

and combine after solving them individually

Production Rules P:

$S \rightarrow S_1 \mid S_2$

$S_1 \rightarrow A_1 A_2$

$A_1 \rightarrow a A_1 b \mid \epsilon$

$A_2 \rightarrow c A_2 \mid \epsilon$

$S_2 \rightarrow A_3 A_4 \mid A_4 A_3$

$A_3 \rightarrow a A_3 \mid a$

$A_4 \rightarrow a A_4 c \mid A_5$

~~$A_5 \rightarrow b \mid \epsilon$ cA_5 A $\mid A_3$~~)

$A_6 \rightarrow c A_3 \mid \epsilon$

$A_5 \rightarrow b A_5 \mid \epsilon$

P.T.O

3 - a ctd

Working of Non-terminal symboles:

S: Start variable which represents the language in question, which we use to branch to two subproblem

S_1: $\{a^i b^j c^k \mid i = j$ where $i, j, k \geq 0\}$

S_2: $\{a^i b^j c^k \mid i \neq k$ where $i, j, k \geq 0\}$

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

A_2: $\{c^i \mid i \geq 0\}$

A_3: $\{a^i \mid i > 0\}$

A_4: $\{a^i b^j c^k \mid i = k$ where $i, j, k \geq 0\}$

A_5: $\{b^i \mid i \geq 0\}$

A_6: $\{c^i \mid i > 0\}$

3(b) CFG to CFG in chomsky normal form.

(i) $S \to BSB / B / \epsilon$

(ii) $B \to 00 / \epsilon$

1. S appears in RHS of (i), introducing S_0

(i) $S_0 \to S$

(ii) $S \to BSB / B / \epsilon$

(iii) $B \to 00 / \epsilon$

2) Removing transition $B \to \epsilon$

(i) $S_0 \to S$

(ii) $S \to BSB / BS / SB / B / \epsilon$

(iii) $B \to 00$

3) Removing $S \to \epsilon$

(i) $S_0 \to S / \epsilon$

(ii) $S \to BSB / BS / SB / B$

(iii) $B \to 00$

4) Removing unit Rule $S \to B$

(i) $S_0 \to S / \epsilon$

(ii) $S \to BSB / SB / BS / 00 / BB$

(iii) $B \to 00$

(5) ~~BSB~~ Removing $S \to BSB$ & add variable for $B \to 00$

(i) $S_0 \to S_1B / BB / BS / SB / AA / \epsilon$

(ii) $S \to S_1B / BB / BS / SB / AA$

(iii) $B \to AA$

(iv) $S_1 \to BS$

(v) $A \to 0$

Camlin