Name → Kavyansh Gangwar
Rollno → 18075027 CSE B.Tech

Mid Semester Examination

Q1 →

Ans 1 → Given →

$\Sigma = \{0,1\}$ and $A \subset \Sigma^*$ is a regular language.

To prove → $B = \{x \in \Sigma^* : \exists$ string $w \in A \mid x$ is obtained by deleting symbols from $w\}$ is regular or not.

Proof →

If we are able to find ~~A~~ a DFA or NFA that accepts ~~a~~ B then we can say it is regular.

We know that A is a regular language. This means there is a DFA say ~~B_0~~ $D_0 = (Q, \Sigma, \delta, q_0, F)$ that accepts A.

Now let us define a NFA $N_0 = (Q, \Sigma, \delta', q_0, F)$ where we define δ' as

i) $\delta'(q, x) = \delta(q, x)$ $\forall q \in Q$, $x \in \Sigma$
i.e if the input symbol belongs to Σ then the transition in the NFA N_0 is same as DFA D_0.

Kavyansh

ii) $\delta'(q, \epsilon) = \delta(q, y)$ $\forall q \in Q$, $y \in \Sigma$
i.e a ϵ-transition from a state to another state in NFA N_0 which is same as the transition in DFA D_0

Now this NFA N_0 is obtained by adding ~~ε~~ ε-transition to DFA D_0 and the ε transition is corresponding to transition in D_0. We can also state clearly that this NFA N_0 accepts B as we ~~to~~ can put ~~of~~ ~~ε~~ ε in place of any deleted symbol and use ε transition.

Since there exists a ε-NFA that accepts B. ~~we can say that the~~ this means there ~~is~~ also exists a NFA and DFA that accepts B

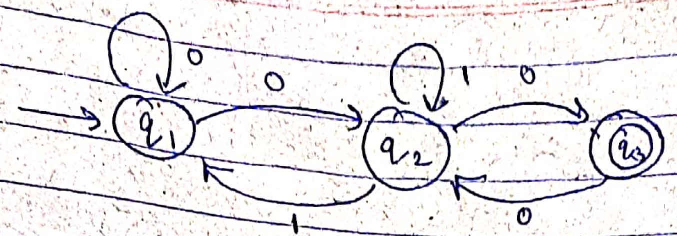
$\Rightarrow$ B is a regular language.

Proved

Kavyansh

Q2

Ans2 →



For the given finite automata, the equations for q_1, q_2 and q_3 can be found. The equations are:

$$q_1 = q_1 \cdot 0 + q_2 \cdot 1 + \varepsilon \quad \text{— (1)}$$

$$q_2 = q_1 \cdot 0 + q_2 \cdot 1 + q_3 \cdot 0 \quad \text{— (2)}$$

$$q_3 = q_2 \cdot 0 \quad \text{— (3)}$$

Now solving the equations

by using (3) in (2) we get

$$q_2 = q_1 \cdot 0 + q_2 \cdot 1 + q_2 \cdot 0 \cdot 0$$

$$\Rightarrow q_2 = q_1 \cdot 0 + q_2 \cdot (1 + 0 \cdot 0)$$

$$\Rightarrow \underbrace{q_2}_{R} = \underbrace{q_1 \cdot 0}_{Q} + \underbrace{q_2}_{R} \cdot \underbrace{(1 + 0 \cdot 0)}_{P}$$

the equation for q_2 is of the form

$$R = Q + RP$$

and we know in this case

$$R = QP^*$$

$$\Rightarrow q_2 = q_1 \cdot 0 \cdot (1 + 0 \cdot 0)^* \quad \text{— (4)}$$

Kavyansh

putting ④ in ①

$$q_1 = q_1 \cdot 0 + q_2 \cdot 1 + \varepsilon$$

$$\Rightarrow q_1 = q_1 \cdot 0 + q_1 \cdot 0 \cdot (1 + 0 \cdot 0)^* \cdot 1 + \varepsilon$$

$$\Rightarrow \underbrace{q_1}_{R} = \underbrace{q_1}_{R} \cdot \underbrace{(0 + 0 \cdot (1 + 0 \cdot 0)^* \cdot 1)}_{P} + \underbrace{\varepsilon}_{Q}$$

equation is again of the form

$$R = Q + RP$$
$$\Rightarrow R = QP^*$$

$$\Rightarrow q_1 = \varepsilon (0 + 0 \cdot (1 + 0 \cdot 0)^* \cdot 1)^*$$

$$\Rightarrow q_1 = (0 + 0 \cdot (1 + 0 \cdot 0)^* \cdot 1)^* \quad \text{—— ⑤}$$

putting ⑤ in ④

$$q_2 = (0 + 0 \cdot (1 + 0 \cdot 0)^* \cdot 1)^* \cdot 0 \cdot (1 + 0 \cdot 0)^* \quad \text{—— ⑥}$$

putting ⑥ in ③

$$q_3 = (0 + 0 \cdot (1 + 0 \cdot 0)^* \cdot 1)^* \cdot 0 \cdot (1 + 0 \cdot 0)^* \cdot 0$$

This is the accepting state. Hence Regular ~~RE~~ Expression accepted by Finite Automata

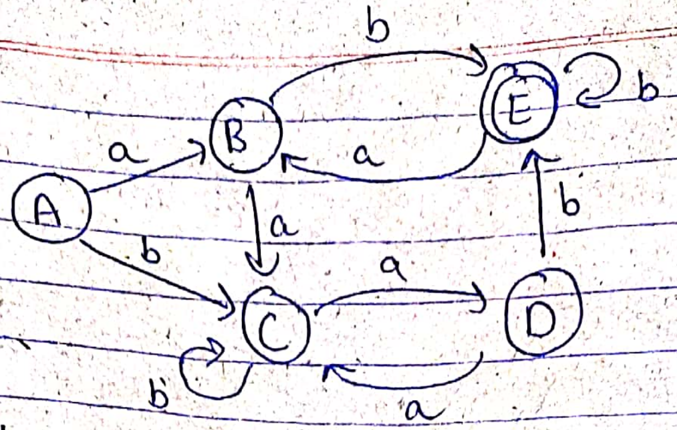
$$\Rightarrow (0 + 0 \cdot (1 + 0 \cdot 0)^* \cdot 1)^* \cdot 0 \cdot (1 + 0 \cdot 0)^* \cdot 0$$

Ans

Kavyansh

Scanned with CamScanner

Q3→
Ans 3→



For the given FA the transition table is

	a	b
A	B	C
B	C	E
C	D	C
D	C	E
E	B	E

	A	B	C	D	E
A	-	-	-	-	-
B		-	-	-	-
C			-	-	-
D				-	-
E	X	X	X	X	-

In thae above FA final state is {E} and Non final states are {A, B, C, D}. On combining final states with Non final states we get

(A, E), (B, E), (C, E), (D, E)

So we mark them with X

Kavyansh

for the remaining pairs we will try to find if it leads to any X. i.e. any state with a final state.

(A,B) $\delta((A,B),a) = (B,C)$ No
$\delta((A,B),b) = (C,E)$ Yes] X

(A,C) $\delta((A,C),a) = (B,D)$ No
$\delta((A,C),b) = (C,C)$ No

(A,BD) $\delta((A,D),a) = (B,C)$ No
$\delta((A,D),b) = (C,E)$ Yes] X

(B,C) $\delta((B,C),a) = (C,D)$ No
$\delta((B,C),b) = (E,C)$ Yes] X

(B,D) $\delta((B,D),a) = (C,C)$ No
$\delta((B,D),b) = (E,E)$ No

(C,D) $\delta((C,D),a) = (D,C)$ No
$\delta((C,D),b) = (C,E)$ Yes] X

⇒

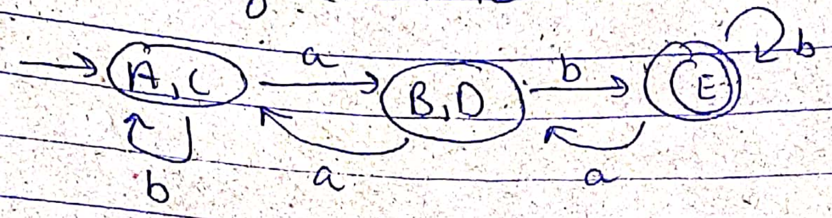
	A	B	C	D	E
A	–	–	–	–	–
B	x	–	–	–	–
C		x	–	–	–
D	x		x	–	–
E	x	x	x	x	–

Kavyansh

The unmarked states are to be combined for minimising.

⇒ The states of the minimizes machine are
$\{(A, C), (B, D), E\}$

⇒ The minimized Finite Automata is

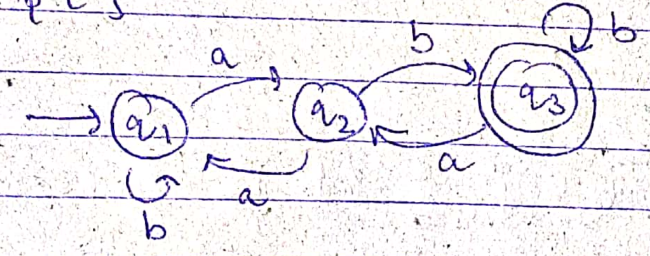


Transition table for this finite automata

	a	b
$\{A, C\}$	$\{B, D\}$	$\{A, C\}$
$\{B, D\}$	$\{A, C\}$	$\{E\}$
$\{E\}$	$\{B, D\}$	$\{E\}$

Let us represent the FA as
$q_1 = \{A, C\}$
$q_2 = \{B, D\}$
$q_3 = \{E\}$

⇒



Kavyansh

to find the regular expression of the FA. the equations are

$q_1 = q_1 \cdot b + q_2 \cdot a + \varepsilon$ —— (1)

$q_2 = q_1 \cdot a + q_3 \cdot a$ —— (2)

$q_3 = q_2 \cdot a + q_3 \cdot b$ —— (3)

Using ③

$$\underbrace{q_3}_{R} = \underbrace{q_2 \cdot b}_{Q} + \underbrace{q_3}_{R} \cdot \underbrace{b}_{P}$$

the equation is of the form $R = Q + RP$
we know in this case $R = QP^*$

$$\Rightarrow q_3 = q_2 \cdot b \cdot b^* \quad \text{—} \quad (4)$$

Putting ④ in ②.

$$\underbrace{q_2}_{R} = \underbrace{q_1 \cdot a}_{Q} + \underbrace{q_2}_{R} \cdot \underbrace{b \cdot b^* \cdot a}_{P}$$

again equation is of the form $R = RP + Q$
$\Rightarrow R = QP^*$

$$\Rightarrow q_2 = q_1 \cdot a \cdot (b \cdot b^* \cdot a)^* \quad \text{—} \quad (5)$$

Kavyansh

putting ⑤ in ①

$$q_1 = q_1 \cdot b + q_1 \cdot a \cdot (b \cdot b^* \cdot a)^* \cdot a + \varepsilon$$

$$\underbrace{q_1}_{R} = \underbrace{q_1}_{R} \underbrace{(b + a \cdot (b \cdot b^* \cdot a)^* \cdot a)}_{P} + \underbrace{\varepsilon}_{Q}$$

again eq. is of form $R = RP + Q$
$\Rightarrow R = QP^*$

$$\Rightarrow q_1 = \varepsilon \, (b + a \cdot (b \cdot b^* \cdot a)^* \cdot a)^*$$

$$\Rightarrow q_1 = (b + a\cdot(b\cdot b^*\cdot a)^*\cdot a)^* \quad \text{—} (6)$$

putting (6) in (5)

$$q_2 = (b + a\cdot(b\cdot b^*\cdot a)^*\cdot a)^*\cdot a\cdot(b\cdot b^*\cdot a)^* \quad \text{—} (7)$$

Putting (7) in (4)

$$q_3 = (b + a\cdot(b\cdot b^*\cdot a)^*\cdot a)^*\cdot a\cdot(b\cdot b^*\cdot a)^*\cdot b\cdot b^*$$

as q_3 is the final state $\Rightarrow$ language accepted by the FA has regular expression $\Rightarrow$

$$(b + a\cdot(b\cdot b^*\cdot a)^*\cdot a)^*\cdot a\cdot(b\cdot b^*\cdot a)^*\cdot b\cdot b^*$$

Ans

Kavyansh

$Q_4 \rightarrow$

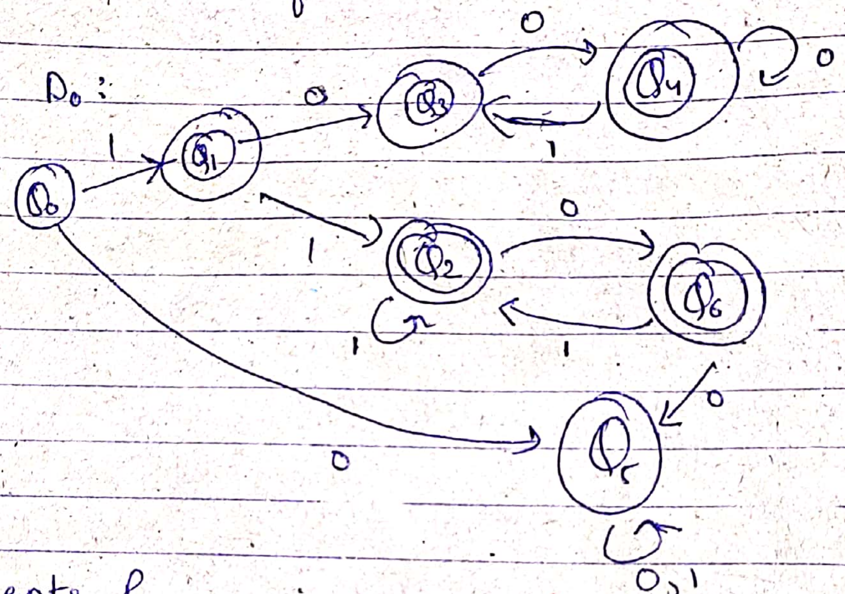
Ans 4 → __Given →__

Regular Expression $R = 1.(1+1.0)^* + 1.0\,(0+0.1)^*$

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

__Step 1 →__ Construction of DFA

DFA D_0 :



D accepts R

__Step 2 →__ No. of non-terminal states in the grammar is equal to the no. of states in the DFA

$|N| = |Q| = 7$

__Step 3 →__ Constructiong production rules for every transition $\delta(q_1, a) = q_2$ as $A \rightarrow aA_1$ and an extra $A \rightarrow a$ if q_2 is an accepting state.

Kaviyansh

State transition table

δ	0	1
q_0	q_5	q_1
q_1	q_3	q_2
q_2	q_6	q_2
q_3	q_4	q_5
q_4	q_4	q_3
q_5	q_5	q_5
q_6	q_5	q_2

Production rules:

$$P = \{ \; S \rightarrow 0S_1 \mid 1S_2 \mid 1,$$
$$S_2 \rightarrow 0S_3 \mid 1S_4 \mid 0 \mid 1,$$
$$S_4 \rightarrow 0S_5 \mid 1S_4 \mid 0 \mid 1,$$
$$S_3 \rightarrow 0S_6 \mid 1S_1 \mid 0,$$
$$S_1 \rightarrow 0S_1 \mid 1S_1,$$
$$S_6 \rightarrow 0S_6 \mid 1S_3 \mid 0 \mid 1,$$
$$S_5 \rightarrow 1S_4 \mid 0S_1 \mid 1 \}$$

$$V_N = \{S, S_1, S_2, S_3, S_4, S_5, S_6\}$$

$T = \{0, 1\}$

Kavyansh

$S = S$

The required regular grammar is

$$G = (V_N, T, P, S)$$

Ans

Q5 →

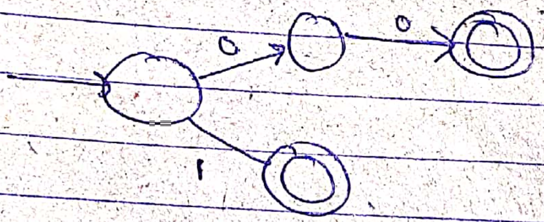
Ans5 → Given :→

A regular expression

$R = 0(11 + 0(00+1)^*)^*$

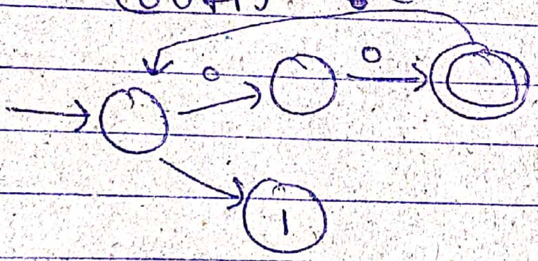
Step 1 →

Step 2 →

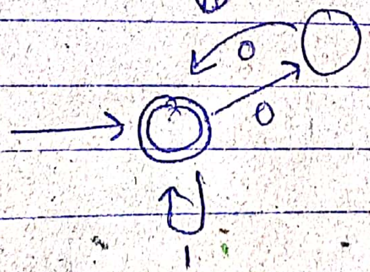


Step 3 →

$(00+1)^* \cdot \epsilon$



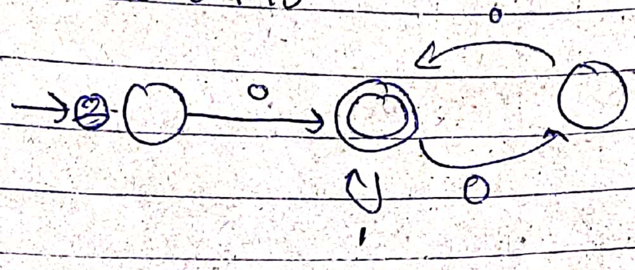
⇓



Kavyansh

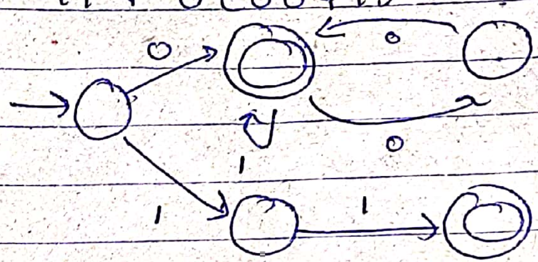
Step 4 →

$0(00+1)^*$



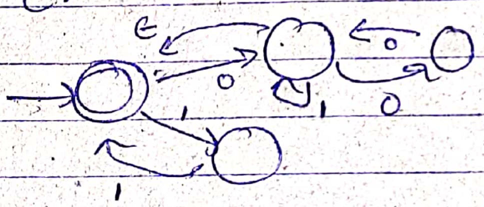
Step 5 →

$11 + 0(00+1)^*$



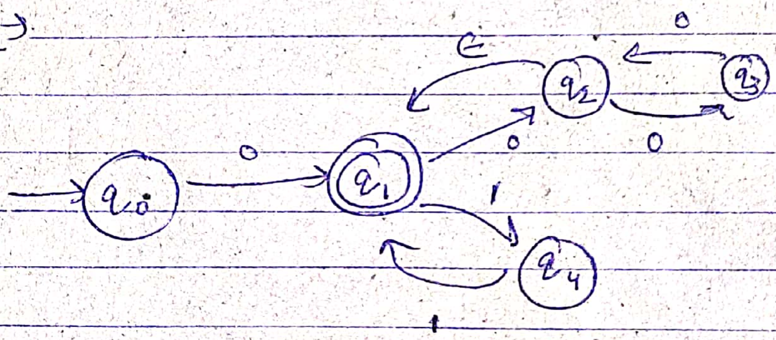
Step 6 →

$(11 + 0(00+1)^*)^*$



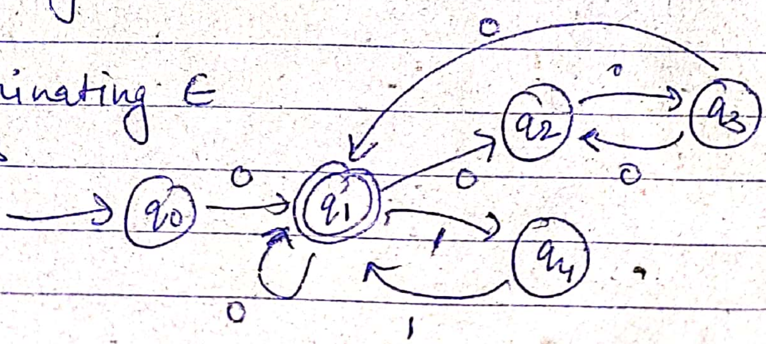
Kavyansh

Step 7 →



The above given is an E-NFA

after eliminating E
so NFA is



Any.

Q6→
Ans6 →

Given → ~~Middle~~ ~~= {~~

$MIDDLE = \{u0v : u, v \in \Sigma^* \text{ and } |u| = |v|\}$
$\Sigma = \{0, 1\}$

To prove →

MIDDLE is not regular

Proof →

We will prove this by method of contradiction but first let us state Pumping lemma.

For every alphabet Σ and every regular language $A \subseteq \Sigma^*$, there exists a pumping length $p \geq 1$ for A that statisfies the following property. For every string $w \in A$ with $|w| \geq p$, it is possible to write $w = xyz$ for strings $x, y, z \in \Sigma^*$ such that

(a) $y \neq \varepsilon$,
(b) $|xy| \leq p$ and
(c) $xy^iz \in A$ for every $i \in \mathbb{N}$

Karyansh

Now let us assume that MIDDLE is regular. By pumping theorem there exists a pumping length $p \geq 1$ with the property described above. Let $w = 1^p 0 1^p$ given that $w \in MIDDLE$ and $|w| = 2p + 1 \geq p$, there must exist string $x, y, z \in \Sigma^*$

such that $w = xyz$, and for which the above three item (a), (b) and (c) are satisfied. As $y \neq \varepsilon$ and $|xy| \leq p$, we conclude that $y = 1^k$ for some choice of an integer k with $k \geq 1$. As $xy^iz \in MIDDLE$ for every $i \geq 0$, we conclude that

$$1^{p+k}01^p = xy^2z \in MIDDLE$$

However, the string $1^{p+k}01^p$ is not contained in MIDDLE, as the only 0 in this string is not in the middle position. We have therefore proved by contradiction, that MIDDLE is not regular.

Proved. Kavyansh

Q7:
Ans 7 →

a → Given →

A language define by

$A = \{ w \in \{0,1\}^* : w = w^R$ and $|w|$ is divisible by 3 $\}$

We need to define context free grammar here for the language $w = w^R$ and $|w|$ is divisible by 3

<u>Grammar</u> →

<u>Production rules</u> →

Kanyansh

$S_0 \to 0S_10 \mid 1S_11 \mid \varepsilon$
$S_1 \to 0S_20 \mid 1S_21 \mid 0 \mid 1$
$S_2 \to 0S_00 \mid 1S_01 \}$

Conversion from context free grammar to CNF

<u>step 1</u> → eliminating ε productions

Nullable variables $= \{S_0\}$

Eliminating ε-production rules by adding

$S_2 \to 00 \mid 11$

$P = \{ S_0 \to 0S_10 \mid 1S_11$
$S_1 \to 0S_20 \mid 1S_21 \mid 0 \mid 1$
$S_2 \to 0S_00 \mid 1S_01 \mid 00 \mid 11 \}$

Scanned with CamScanner

Step 2 → Eliminating unit productions

No unitary productions

Step 3 → Restricting the right side of production to single transaction on string of two or more variables.

This step yields →

P : $S_0 \rightarrow X_a S_1 X_a \mid X_b S_1 X_b$

$S_1 \rightarrow 0 \mid 1 \mid X_a S_2 X_a \mid X_b S_2 X_b$

$S_2 \rightarrow X_b S_0 X_b \mid X_a S_0 X_a \mid X_a X_a \mid X_b X_b$

$X_a \rightarrow 1$; $X_b \rightarrow 0$

Step 4 → Final step to CNF

Reducing long productions

$S_0 \rightarrow X_a T_a$, $T_a \rightarrow S_1 X_a$

$S_0 \rightarrow X_b T_b$, $T_b \rightarrow S_1 X_b$

$S_1 \rightarrow 0 \mid 1$

$S_1 \rightarrow X_b T_c$, $T_c \rightarrow S_2 X_a$

$S_1 \rightarrow X_b T_D$, $T_D \rightarrow S_2 X_b$

$S_2 \rightarrow X_a T_E$, $T_E \rightarrow S_0 X_a$

$S_2 \rightarrow X_b T_F$, $T_F \rightarrow S_0 X_b$

$S_2 \rightarrow X_a X_a \mid X_b X_b$

Any Kavyansh

Q7.
Ans7 →

b → Given →

A language define by

$C = \{0^n 1^m : m \leq n \leq 3m\}$

$S \rightarrow \epsilon \mid 0S1 \mid 00S1 \mid 000S1$

Converting content free grammar to CNF

Step 1 → eliminating ϵ-productions

Nullable variable = {S}

To elliminate ϵ-production rules adding
$S \rightarrow 01$, $S \rightarrow 001$, $S \rightarrow 0001$

$P = \{ S \rightarrow 01 \mid 0001 \mid 0001 \mid 0S1 \mid 00S1 \mid 0001 \}$

Step 2 → Eliminating unit productions
No unitary productions

Step 3 → We restrict the R.H.S of production to single ~~trans~~ transaction on string of two or more variables.

$P: S \rightarrow X_b X_b X_a$; $S \rightarrow X_b X_b X_b X_a$
$S \rightarrow X_b X_a$; $S \rightarrow X_b S X_a$; $S \rightarrow X_b X_b S X_a$
$S \rightarrow X_b X_b X_b S X_a$

Kavyansh

$X_a \rightarrow 1$; $X_b \rightarrow 0$

Step 4)

Final step to CNF

Reducing long productions

$S \to Y_a X_a$; $Y_a \to X_b Y_b$

$S \to Y_a Y_b$; $Y_b \to X_b X_a$

$S \to X_b X_a$

$S \to X_b Y_C$; $Y_C \to S X_a$

$S \to Y_a Y_C$;

$S \to Y_D Y_C$; $Y_D \to Y_a X_b$

$X_a \to 1$; $X_b \to 0$

Kavyansh

Ans

Q8 →
Ans8 →

Given →

$\Sigma = \{0,1\}$
$C \subseteq \Sigma^*$ is a context-free language.

To prove →

$A = \{w : 0w0 \in C\}$ is context-free.

Proof →

If the language C is empty, then A is also empty and therefore context-free.

We will therefore assume that C is non-empty for the remainder of the solution.

Let G be a context-free grammar in chomsky normal form that generates C, and assume without loss of generality that G has no useless variables (i.e. variables that generate no strings).

We will first define a context-free grammar H that generates A as follows:

- For each variable X that appears in G, we will have 4 variables in H:

X, X_L, X_R and X_{LR}

Kavyansh

Naturally this assumes that none of the variable X_L, X_R, X_{LR} appears in G.

The meaning of these variables is as follows →

→ X will generate exactly same strings in H that it generates in G

→ X_L will generate those strings that result from taking any string that could be generated from X in G, except that a 0 is removed from the left-hand side of that string.

→ X_R is similar to X_L, except that a 0 is removed from the right-hand side of the string rather than the left.

→ X_{LR} generates strings that can be obtained by removing a 0 from both the left hand side and right hand side of a string generated by X in G.

The start variable of H is S_{LR}, for S being the start variable of G.

- For every rule of the form $X \to YZ$ in G, include these rules in H:

$$X \to YZ$$
$$X_L \to Y_L Z$$
$$X_R \to Y Z_R$$
$$X_{LR} \to Y_L Z_R$$

Kavyansh

- For every rule of the form $X \to 0$ in G, include these rules in H:

$$X \to 0$$
$$X_L \to \varepsilon$$
$$X_R \to \varepsilon$$

- for every rule of the form $X \to 1$ in G, include just this rules in H:

$$X \to 1$$

- If the rule $S \to \varepsilon$ appears in G, ignore it.

It is the case that $L(H) = A$ and therefore A is context-free.

Proved

Kavyansh