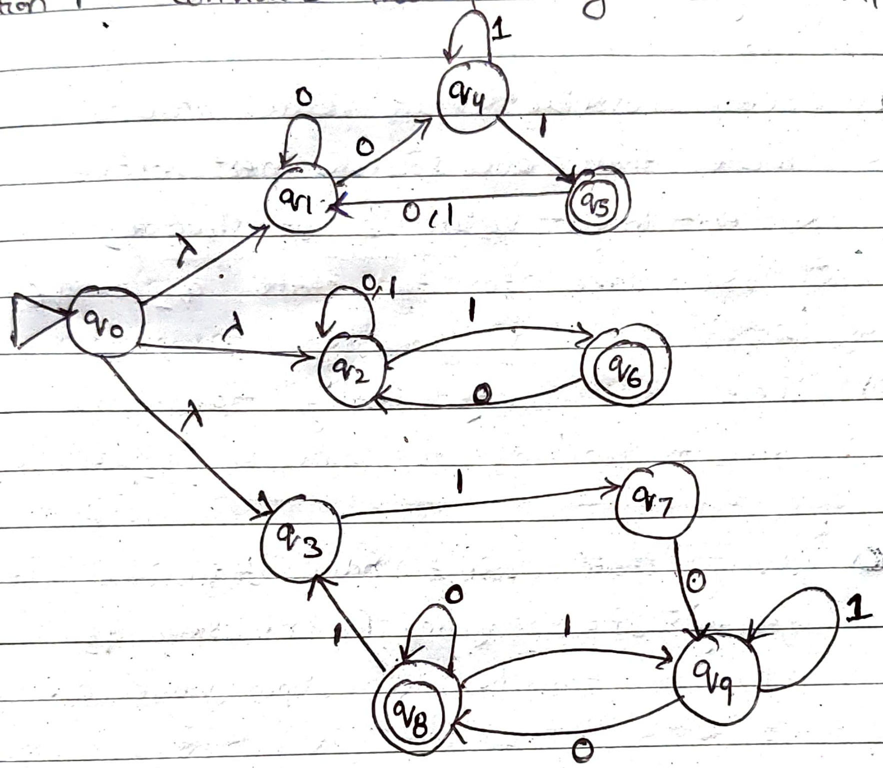
Archit Agrawal
202051213

CS305: Formal Language and Automata Theory

Assignment 1

Question 1: Consider the following λ-NFA M_1:



(i) Give three examples of a strings in $L(M_1)$ of length at least 5, one for each of the final states.

→ for final state q_5 : 011001
→ for final state q_6 : 0011101
→ for final state q_8 : 1010100

Archit Agrawal

202051213

(ii) Is M_1 deterministic or non-deterministic? Explain your answer.

M_1 is non-deterministic because of the following reasons:

(a) Every possible input value have more than one possible next state. For ex. Input value 1 at state q_2 can take to the state q_2 or q_6.

(b) There are empty string transitions from state q_0 to state q_1, q_2 and q_3.

(c) Every state doesn't have transition for every input value. For ex. state q_3 has no transition for input value 0.

(iii) What is $L(M_1)$? Explain your answer.

For finding $L(M_1)$, ~~we~~ ~~will~~ first convert λ-NFA to NFA.

We can convert λ-NFA to NFA by constructing transition table considering the λ transitions too.

The NFA for M_1 is represented by transition table on next page.

Archit Agrawal
202051213

State	0	1
$\rightarrow q_0$	$\{q_1, q_2, q_4\}$	$\{q_2, q_6, q_7\}$
q_1	$\{q_1, q_4\}$	ϕ
q_2	q_2	$\{q_2, q_6\}$
q_3	ϕ	q_7
q_4	ϕ	$\{q_4, q_5\}$
q_5	q_1	q_1
q_6	q_2	ϕ
q_7	q_9	ϕ
q_8	q_8	$\{q_3, q_9\}$
q_9	q_8	$\{q_9\}$

Writing incoming equation's for each state:

$$q_0 = \lambda \quad \text{—(I)}$$
$$q_1 = q_0 0 + q_1 0 + q_5 0 + q_5 1 \quad \text{—(II)}$$
$$q_2 = q_0 0 + q_0 1 + q_2 0 + q_2 1 + q_6 0 \quad \text{—(III)}$$
$$q_3 = q_8 1 \quad \text{—(IV)}$$
$$q_4 = q_0 0 + q_1 0 + q_4 1 \quad \text{—(V)}$$
$$q_5 = q_4 1 \quad \text{—(VI)}$$
$$q_6 = q_0 1 + q_2 1 \quad \text{—(VII)}$$
$$q_7 = q_0 1 + q_3 1 \quad \text{—(VIII)}$$
$$q_8 = q_8 0 + q_9 0 \quad \text{—(IX)}$$
$$q_9 = q_7 0 + q_8 1 + q_9 1 \quad \text{—(X)}$$

Let regular expression for states q_5, q_6 and q_8 be R_1, R_2 and R_3 respectively and that for M_1 be R:

$$\therefore \quad R = R_1 + R_2 + R_3$$

Archit Agrawal

202051213

$$q_1 = 0 + q_1 0 + q_5 0 + q_5 1 \qquad \left(\because q_0 = \lambda\right) \text{ eq}^n \text{ (II)}$$

$$\underbrace{q_1}_{R} = \underbrace{0 + q_5 0 + q_5 1}_{Q} + q_1 \underbrace{0}_{P}$$

Using Arden's Lemma, since $\lambda \notin \{0\} = P$ ∴

$$\therefore \quad q_1 = (0 + q_5 0 + q_5 1) 0^* \qquad \text{— (XI)}$$

$$q_4 = 0 + q_1 0 + q_4 1 \qquad (\text{Eq}^n \text{ (V)})$$

Substituting q_1 in (V) from (XI),

$$\underbrace{q_4}_{R} = \underbrace{0 + (00^* + q_5 00^* + q_5 10^*) 0}_{Q} + q_4 \underbrace{1}_{P}$$

$\because \lambda \notin \{1\}$, using Arden's lemma,

$$q_4 = (0 + 00^*0 + q_5 00^*0 + q_5 10^*0) 1^* \qquad \text{— (XII)}$$

Substituting q_4 from (XII) in (VI)

$$q_5 = (01^* + 00^*01^* + q_5 00^*01^* + q_5 10^*01^*) 1$$

$$\therefore \quad \underbrace{q_5}_{R} = \underbrace{01^*1 + 00^*01^*1}_{Q} + q_5 \underbrace{(00^*01^*1 + 10^*01^*1)}_{P}$$

Clearly $\lambda \notin P$, using Arden's lemma

$$\boxed{R_1 = q_5 = (01^*1 + 00^*01^*1) \cdot (00^*01^*1 + 10^*01^*1)^*} \qquad \text{— (XIII)}$$

Archit Agrawal
202051213

Now,

$$\underbrace{q_2}_{R} = \underbrace{0+1+q_6 0}_{Q} + q_2 \underbrace{(0+1)}_{P} \qquad (\text{Eq } (III), \because q_6 = \lambda)$$

$\because \lambda \notin P$, using Arden's Lemma,

$$q_2 = (0+1+q_6 0)(0+1)^* \quad \text{— (XIV)}$$

Substituting q_2 from (XIV) in (VII),

$$q_6 = 1 + \left(0(0+1)^* + 1(0+1)^* + q_6 0(0+1)^*\right)1$$

$$\therefore \underbrace{q_6}_{R} = \underbrace{1 + 0(0+1)^*1 + 1(0+1)^*1}_{Q} + q_6 \underbrace{0(0+1)^*1}_{P}$$

$\because \lambda \notin P$, using Arden's Lemma,

$$\boxed{R_2 = q_6 = \left(1 + (0+1)(0+1)^*1\right)\left(0(0+1)^*1\right)^*} \quad \text{— (XV)}$$

Now, substituting q_3 from (IV) in (VIII),

$$q_7 = 1 + q_8 11 \quad \text{— (XVI)}$$

Substituting q_7 from (XVI) to (X),

$$\underbrace{q_9}_{R} = \underbrace{10 + q_8 110 + q_8 1}_{Q} + q_9 \underbrace{1}_{P}$$

Archit Agrawal
202051213

$\because \lambda \notin P$, using Arden's lemma,

$$q_9 = (10 + q_8 110 + q_8 1) 1^* \quad \text{— (XVII)}$$

Substituting q_9 from XVII to (IX)

$$q_8 = (101^* + q_8 1101^* + q_8 11^*) 0 + q_8 0$$

$$\underbrace{q_8}_{R} = \underbrace{101^* 0}_{Q} + q_8 \underbrace{(1101^* 0 + 11^* 0 + 0)}_{P}$$

$\because \lambda \notin P$, using Arden's Lemma,

$$\boxed{R_3 = q_8 = (101^* 0)(1101^* 0 + 11^* 0 + 0)^*} \quad \text{— (XVIII)}$$

$\therefore$ the regular expression for $L(M_1)$ is :

$$R = R_1 + R_2 + R_3$$

where $R_1 = (01^*1 + 00^*01^*1)(00^*01^*1 + 10^*01^*1)^*$

$$R_2 = (1 + (0+1)(0+1)^*1)(0(0+1)^*1)^*$$

and $R_3 = (101^*0)(1101^*0 + 11^*0 + 0)^*$

(iv) Calculate $\hat{\delta}(q_0, w)$ where $w = w_1 w_2$, and w_1 is binary encoding of the last digit of your roll number and w_2 is binary encoding of your birth date?

Archit Agrawal
2020512213

Roll No. → 2020512213
Date of Birth → Sep 17, 2001

$\therefore\ w_1 = 11$
$w_2 = 10001$

$\therefore\ w = 1110001$

using the transition table in part (iii) of this question and the property

$$\hat{\delta}(q_0, aw) = \hat{\delta}(\delta(q_0, a), w)$$

$$\therefore\ \hat{\delta}(q_0, 1110001) = \hat{\delta}(\delta(q_0, 1), 110001)$$

$$= \hat{\delta}(\{q_2, q_6, q_7\}, 110001)$$

$$= \hat{\delta}(\{q_2, q_6\}, 10001)$$

$$= \hat{\delta}(\{q_2, q_6\}, 0001)$$

$$= \hat{\delta}(\{q_2\}, 001)$$

$$= \hat{\delta}(\{q_2\}, 01)$$

$$= \hat{\delta}(\{q_2\}, 1)$$

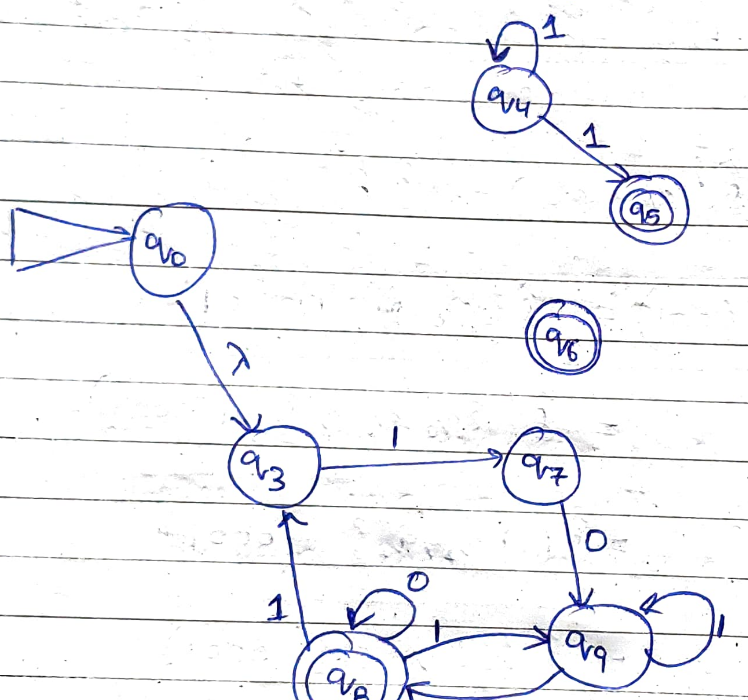
$$\boxed{\hat{\delta}(q_0, w) = \{q_2, q_6\}}$$

Archit Agrawal

202051213

(v) Consider M_2 which is obtained by deleting the states q_2 and q_1, and thereby all the transitions involving q_1 and q_2. What is $L(M_2)$?

Let us first delete the states q_1 and q_2 from given λ-NFA; and all transitions involving q_1 and q_2.



The NFA transition table for M_2 is given below

State	0	1
$\rightarrow q_0$	ϕ	q_7
q_3	ϕ	q_7
q_4	ϕ	$\{q_4, q_5\}$

Archit Agrawal
20205123

q_5	ϕ	ϕ
q_6	ϕ	ϕ
q_7	q_9	ϕ
q_8	q_8	$\{q_3, q_9\}$
q_9	q_8	q_9

Writing incoming equation's for the states.

$q_0 = \lambda$ — (I)

$q_4 = q_4 1$ — (II)

$q_5 = q_4 1$ — (III)

$q_3 = q_0 1$ — (IV)

$\boxed{q_6 = \lambda}$ (V)

$q_7 = q_0 1 + q_3 1$ (VI)

$q_8 = q_8 0 + q_9 0$ (VII)

$q_9 = q_7 0 + q_8 1 + q_9 1$ (VIII)

Substituting q_4 from (II) in (III),

$q_5 = q_4 11$

again substituting q_4 from (II) in (III),

$q_5 = q_4 111$

$\boxed{\therefore\ q_5 = 1^*}$ (infinite regression and $q_4 = \lambda$)

$q_7 = 1 + q_8 11$ (substituting q_3 from (IV))

$\therefore \underbrace{q_9}_{R} = \underbrace{10 + q_8 110 + q_8 1}_{Q} + q_9 \underbrace{1}_{P}$

Archit Agrawal

202051213

Since, $\lambda \notin P$, using Arden's Lemma,

$$q_9 = (10 + q_8 110 + q_8 1) 1^* \quad \text{(IX)}$$

Substituting q_9 from (IX) to (VII)

$$q_\emptyset = q_8 0 + (10 + q_\emptyset 110 + q_\emptyset 1) 1^* 0$$

$$\underbrace{q_\emptyset}_{R} = \underbrace{101^*0}_{Q} + q_\emptyset \underbrace{(0 + 1101^*0 + 11^*0)}_{P}$$

$\because \lambda \notin P$, using Arden's Lemma

$$\boxed{q_\emptyset = (101^*0)(0 + 1101^*0 + 11^*0)^*}$$

Since, q_5, q_6 and $q_\emptyset$ are final states. There fore, $L(M_2)$ is regular language with regular expression R.

$$\therefore R = R(q_5) + R(q_6) + R(q_\emptyset)$$

$$\boxed{R = \lambda + 1^* + (101^*0)(0 + 1101^*0 + 11^*0)^*}$$

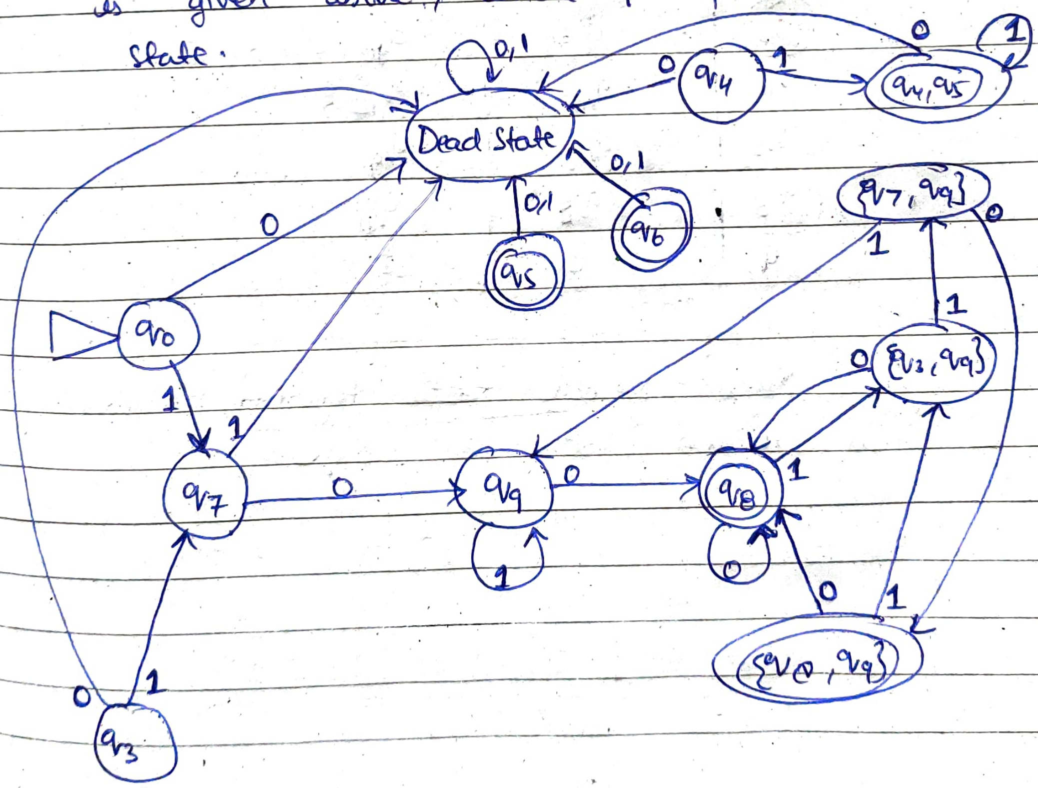
(vi) Convert the $L(M_2)$ to its equivalent DFA, say $L(M_3)$ and minimize $L(M_3)$.

Using subset construction method to form DFA, $L(M_3)$.

Archit Agrawal
20205123

State	0	1
→ q_0	ϕ	q_7
q_3	ϕ	q_7
q_4	ϕ	$\{q_4, q_5\}$
q_5	ϕ	ϕ
q_6	ϕ	ϕ
q_7	q_9	$\{q_3, q_9\}$
q_8	q_8	q_9
q_9	q_8	q_9
$\{q_4, q_5\}$	ϕ	$\{q_4, q_5\}$
$\{q_3, q_9\}$	q_8	~~$\{q_3, q_9\}$~~ $\{q_7, q_9\}$
$\{q_7, q_9\}$	$\{q_8, q_9\}$	q_9
$\{q_8, q_9\}$	q_8	$\{q_3, q_9\}$

Hence, the transition table for DFA, $L(M_3)$ is given above, where ϕ represents dead state.



Archit Agrawal
2020S1213

To check, or to draw minimized DFA, we use Myhill-Nerode theorem,

using DFA, $L(M_3)$, and renaming states, $L(M_3)$ now becomes,

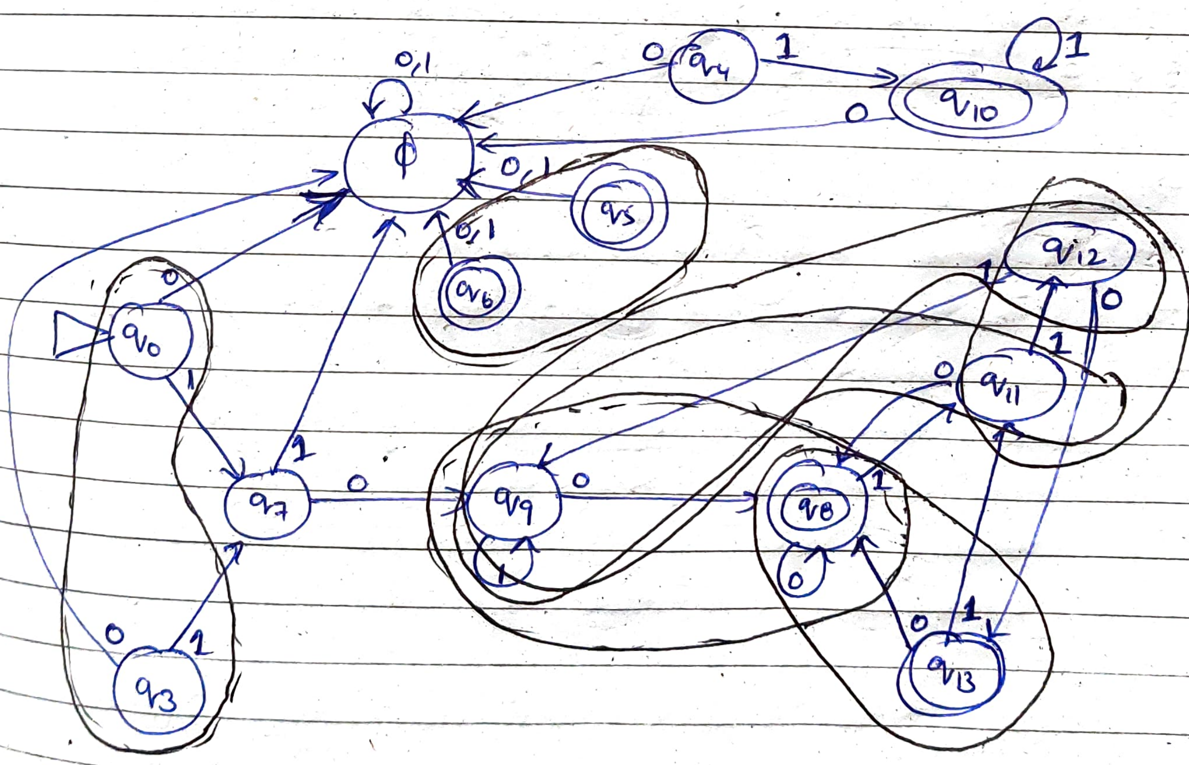
State	0	1
$\rightarrow q_0$	ϕ	q_7
q_3	ϕ	q_7
q_4	ϕ	q_{10}
q_5	ϕ	ϕ
q_6	ϕ	ϕ
q_7	q_9	ϕ
q_8	q_8	q_{11}
q_9	q_8	q_9
q_{10}	ϕ	q_{10}
q_{11}	q_8	q_{12}
q_{12}	q_{13}	q_9
q_{13}	q_8	q_{11}
ϕ	ϕ	ϕ

The Myhill-Nerode ~~matrix~~ theorem matrix is drawn on next page

Archit Agrawal

202051213

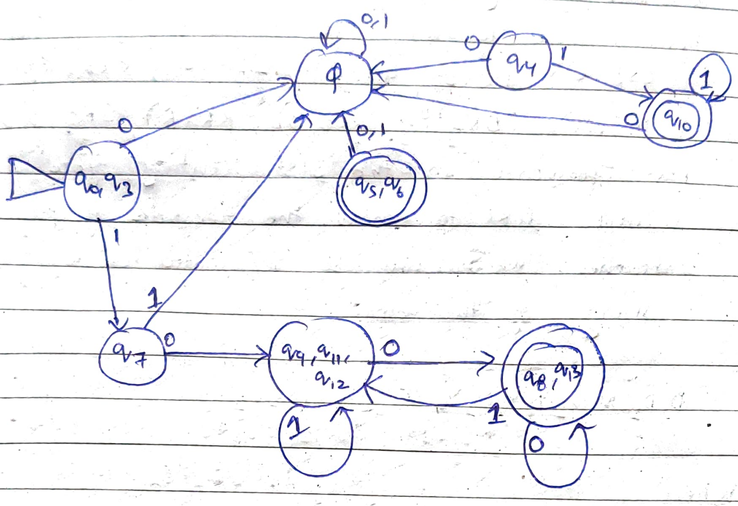
	q_0	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}	q_{13}	ϕ
q_0													
q_3	✓	✓											
q_4	✓	✓	✓										
q_5	✓	✓	✓										
q_6	✓	✓	✓										
q_7	✓	✓	✓	✓	✓								
q_8	✓	✓	✓	✓	✓	✓							
q_9	✓	✓	✓	✓	✓	✓							
q_{10}	✓	✓	✓	✓	✓	✓	✓	✓					
q_{11}	✓	✓	✓	✓	✓	✓	✓		✓				
q_{12}	✓	✓	✓	✓	✓	✓	✓		✓				
q_{13}	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		
ϕ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	



Archit Agrawal

2020512 13

After combining the possible states, the DFA that is minimized is



Hence, the minimized DFA is drawn above.

Archit Agrawal

202051213

Question 2 Consider the IPL tournament played in our country every year since 2007 between the following 10 teams:

CSK (C), DC (D), GT (G), LSG (L), KKR (K), MI (M), PBKS (P), RCB (B), RR (R), SRH (S).

Fast forward in the 31st century, you would like to know the records of previous matches which can be represented as a string of the form:

$$D_1 D_2 M_1 M_2 Y_1 Y_2 Y_3 Y_4 T_1 T_2 W R_1 R_2 R_3$$

where,

- → $D_1 D_2$ are two digits of date and D_1 can only be 0, 1, 2, 3
- → $M_1 M_2$ are two digits of month of the date, and the matches are played b/w March and May.
- → $Y_1 Y_2 Y_3 Y_4$ are digits of year of the date and Y_1 can only be 2 or 3.
- → T_1, T_2 are the two teams involved in a match
- → W is winning team
- → $R_1 R_2 R_3$ are three digit runs of the winning team, and you may assume that the winning team would score at least 50 runs and not more than 300.

Assume that the scores were written as a string one after the other as one enormous

Archit Agrawal

202051213

string, find the regular expressions to identify the following instances:

(i) any match in year 2022 won by MI.

If month is March or May, date can go from 0-31, if Month is April day lies in the range 0-30.

⇒ ∴ Regex for $D_1D_2M_1M_2$ is given as:

~~((0[1-9]|[12]\d|3[01])0[35])|((0[1-9]|[12]\d|3[0])~~

"((0[1-9]|[12]\d|3[01])0[35])|((0[1-9]|[12]\d|30)04)"

⇒ Regex for ~~[struck]~~ $Y_1Y_2Y_3Y_4$ will be "2022"

⇒ Regex for T_1T_2: either T_1 can be M (MI) or T_2 can be M(MI). If T_1 is M, T_2 is other team and if T_2 is M, T_1 is other team.

"(M[^M]|[^M]M)"

⇒ winning team is MI (M)
∴ regex for W is "M"

⇒ Score can be anything between 050 and 300
∴ regex for $R_1R_2R_3$ is

"(0[5-9]\d|[12]\d\d|300)

Archit Agrawal

202051213

therefore, regex for any match in 2022 won by MI is:

" ((0[1-9]|12\d|3[01])0[35]) | ((0[1-9]|[12]\d|30)04)

2022 (M[^M]|[^M]M) M (0[5-9]\d|[12]\d\d|300)"

(ii) Any match involving GT after 2021.

⇒ regex for $D_1D_2M_1M_2$ will be same as part (i)

$D_1D_2M_1M_2$: " ((0[1-9]|[12]\d|3[01])0[35]) | ((0[1-9]|[12]\d|30)04)"

: " 0[1-9]|[12]\d|(3[01]0[35]|3004)"

⇒ any year after 2021, and $Y_1 \in \{2,3\}$

∴ regex for $Y_1Y_2Y_3Y_4$.

$Y_1Y_2Y_3Y_4$: "$\underbrace{\text{202[2-9]}}_{2022-2029}|\underbrace{\text{20[3-9]\\d}}_{2030-2099}|\underbrace{\text{2[^0]\\d\\d}}_{2100-2999}|\underbrace{\text{3\\d\\d\\d}}_{3000-3999}$"

⇒ team GT (G) is involved

∴ regex for T_1T_2 : "(G[^G]|[^G]G)"

⇒ the winning team must be any of the playing team:

∴ regex for W : "[T_1T_2]"

⇒ regex for runs $R_1R_2R_3$ will be same as part i)

$R_1R_2R_3$: "(0[5-9]\d|[12]\d\d|300)"

Archit Agrawal

2020 51213

∴ regex for any match involving GT after 2021 is

" ~~(0[1-9]|[12]\d|3[01])0[35]~~

~~0[1-9]|[12]\d~~

" (0[1-9]|[12]\d|(3[01]0[35]|3004))

(202[2-9]|20[3-9]\d|2[^0]\d\d|3\d\d\d)

(G[^G]|[^G]G) [T, T2]

(0[5-9]\d|[12]\d\d|300)

(iii) any match played on 6th March, 3022 in which the winning team scored 180 runs:

⇒ $D_1 D_2 M_1 M_2 Y_1 Y_2 Y_3 Y_4$: "06033022"

⇒ regex for T_1, any team from given teams,

~~T_1 : ([B-D]|G|~~

T_1 : "([BCDGKLM PRS])"

⇒ T_2 must not be same as T_1, therefore, regex for T_2

T_2 : "[^T₁]"

⇒ winning team must be out of T_1 and T_2.

regex for W : "[T_1 T_2]"

⇒ regex for Runs $R_1 R_2 R_3$: "180"

Archit Agrawal
2020512 13

∴ regex for any match on 6^{th} March, 3022 in which winning team scored 180 runs
ie :

" 06033022 ([BCDGKLMPRS]) ([^T₁]) ([T₁T₂]) 180 "

(iv) Any sequence of wins by CSK.

~~date and month can be~~
date can be anything in month of March to May.

⇒ ∴ regex for $D_1 D_2 M_1 M_2$ is :

" 0[1-9] | [12]\d | (3[01] 0[35] | 3004) "

⇒ Year can be any year after 2006, since tournament started in 2007.

∴ regex for $Y_1 Y_2 Y_3 Y_4$: ($Y_1 \in \{2,3\}$)

" 200[7-9] | 20[1-9]\d | 2[1-9]\d\d | 3\d\d\d "

⇒ match will be played b/w T_1 and T_2, since winning team is CSK, therefore out of T_1 and T_2, one must be CSK and other team must not be CSK.

∴ regex for $T_1 T_2$: " C[^C] | [^C]C "

⇒ winning team is CSK ; regex for W : " C "

Archit Agrawal

20205121З

⇒ any no. of runs can be scored by winning team b/w 50 and 300.

∴ regex for $R_1 R_2 R_3$:

```
" 0[5-9]\d | [12]\d\d | 300 "
```

∴ regex for any sequence of wins by csk is, that is ~~eth~~ the sequence is repeated at least one time.

∴ ~~regex~~ will be : $(D_1 D_2 M_1 M_2 Y_1 Y_2 Y_3 Y_4 T_1 T_2 W R_1 R_2 R_3)^+$

```
((0[1-9]|[12]\d|(3[01]0[35]|3004))(200[7-9]|20[1-9]\d|
2[1-9]\d\d|3\d\d\d)(c[^c]|[^c]c)c
(0[5-9]\d|[12]\d\d|300))+
```

(V) ~~any match~~ any regex of your choice

Description: any match played between PBKS(P) and KKR(K) in which winning team scored more than 175 runs.

regex for $D_1 D_2 M_1 M_2 Y_1 Y_2 Y_3 Y_4$ will be same as part (iv).

Archit Agrawal

202051213

⇒ regex for $T_1 T_2$: "(PK | KP)"

⇒ regex for W : "[PK]"

⇒ regex for runs by winning team (>175)

"17[6-9] | 1[89]\d | 2\d\d | 300"

∴ regex for any match played between PBKS and KKR in which winning team scored more than 175 runs is:

"(0[1-9] | [12]\d | (3[01]0[35]) | (3004)) (200[7-9] | 20[1-9]\d | 2[1-9]\d\d | 3\d\d\d) (PK | KP) ([PK]) (17[6-9] | 1[89]\d | 2\d\d | 300)"

Archit Agrawal

202051213

Question 3 Consider a DFA with the following transition table. Convert the DFA to its equivalent regular expression using Arden's Theorem.

	a	b
* → A	B	C
B	A	C
C	A	A

Writing incoming equations for each state :

~~$A = \lambda + Ba +$~~

$$A = \lambda + Ba + Ca + Cb \tag{I}$$

$$B = Aa \tag{II}$$

$$C = Ab + Bb \tag{III}$$

Substituting $B = Aa$ in (III),

$$C = Ab + Aab \tag{IV}$$

Substituting B and C in (I) from (II) & (IV),

$$A = \lambda + Aaa + (Ab + Aab)a + (Ab + Aab)b$$

$$\underbrace{A}_{R} = \underbrace{\lambda}_{Q} + A\underbrace{(aa + ba + aba + bb + abb)}_{P}$$

Clearly $\lambda \notin P$, therefore, using Arden's Theorem,

$$A = \lambda(aa + ba + aba + bb + abb)^*$$

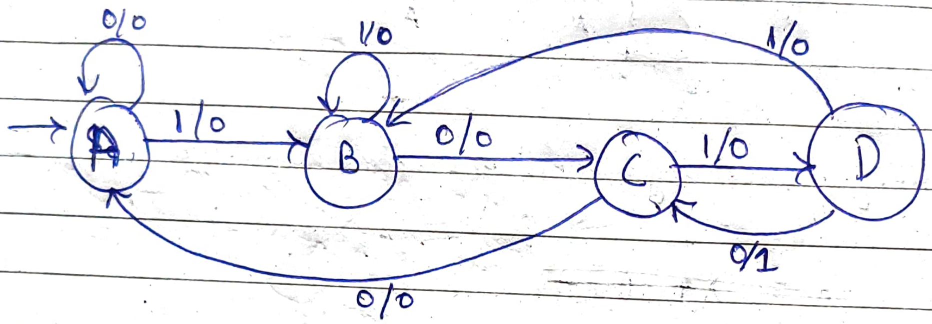
∴ regular expression for given DFA is

$$R = (aa + ba + aba + bb + abb)^*$$

Archit Agrawal
202051213

Question 4. Construct a Mealy Machine for detecting a sequence 1010 where the overlapping sequences are also allowed.

~~Let~~ Building the machine such that it gives output 1 when a sequence 1010 is encountered and 0 otherwise.



→ No. of occurences of 1 in the output is equal to the no. of times 1010 sequence appears in input.

→ a sequence of 101 in output suggests that the two sequences of 1010 overlap.

Ex. input : 1100 $\boxed{10\,10}$ 10 overlapping of sequence 1010
0 0 0 0 0 0 0 1 0 1

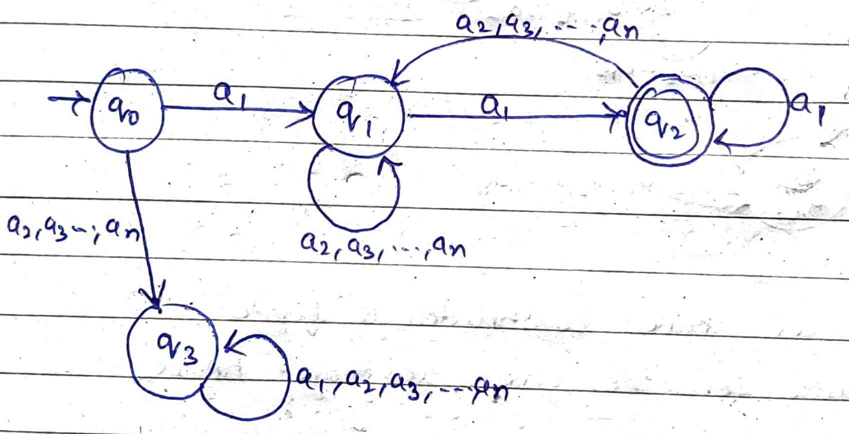
Archit Agrawal

2020S1213

Question 5. A word is said to be bordered if $\exists a \in \Sigma$ such that $w = axa$ for $x \in \Sigma^*$. Show that all bordered words over $\Sigma = \{a_1, a_2, \cdots, a_n\}$ is a regular language.

We will try to form a finite automata for the given language, if we succeeded, it means that L is regular language.

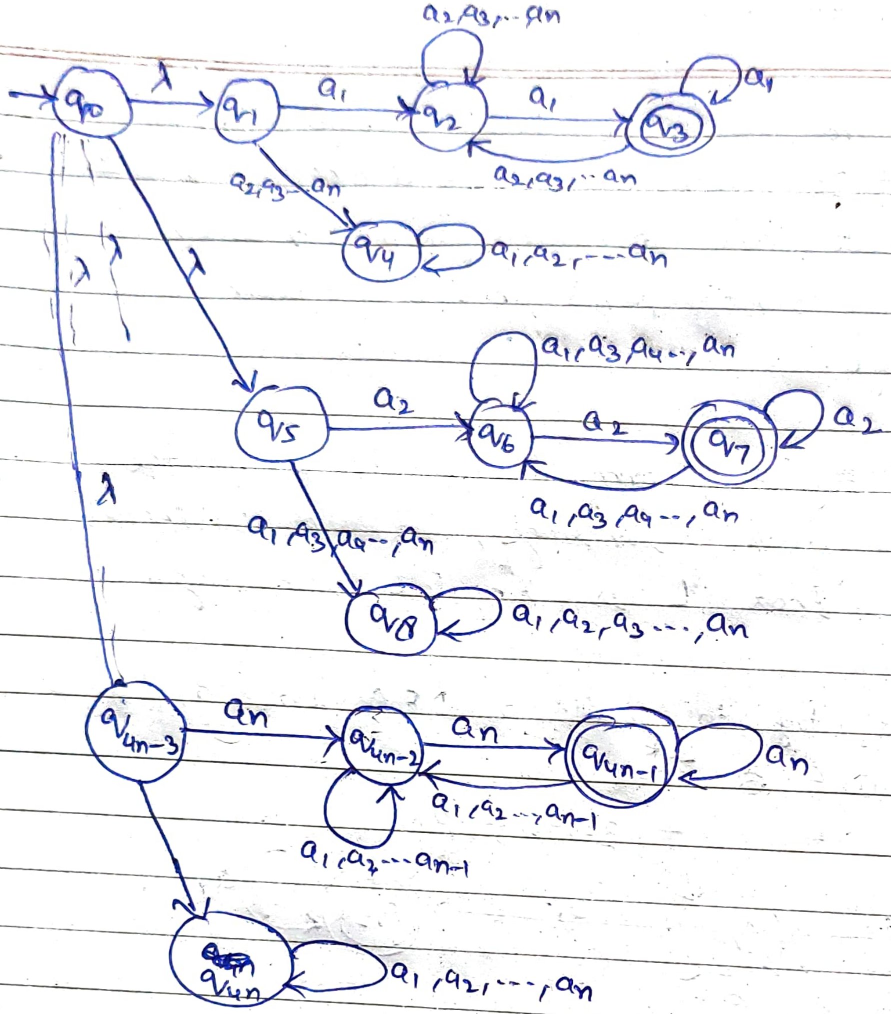
Let's first try to build a DFA for $a_1 w a_1$ where $w \in \Sigma^*$



As it can be seen, the DFA above accepts only the strings of the form $a_1 w a_1$ where $w \in \Sigma^*$

Similarly, we can build DFA for $a_2 w a_2$, $a_3 w a_3$ $\cdots$, $a_n w a_n$. And we can combine these DFA's using λ-transition to ~~build~~ make an λ-NFA for the language $L = awa$ where $w \in \Sigma^*$ and $a \in \{a_1, a_2, \cdots, a_n\}$

Archit Agrawal
202051213



Hence, we have constructed a finite automata for the given language. Hence, the given language is a regular language.

Archit Agrawal
2020S1213

Question 6. Prove that the language

$$L = \{a^n b^n c^n : n \geq 0\}$$

is not regular.

We can prove that L is not a regular language using Pumping Lemma.

Let us consider a pumping length of m.

Now, let us take a string w such that $w \in L$ and $|w| > m$.

Let $w = a^m b^m c^m$

clearly $|w| = 3m > m$

Let us decompose w as $w = xyz$ such that
$|xy| \leq m$
$|y| > 0$

$\cancel{w = a^{m-k} a^k}$

$w = a^{m-1} a b^m c^m$

where, $x = a^{m-1}$
$y = a$
$z = b^m c^m$

Now, we just need to find an i such that $xy^iz \notin L$. For $i=0$, $xy^iz = a^{m-1} b^m c^m$ which does not belong to L. Hence, L is not a regular language.