Nirjhar Nath
BMC202239

① For my birthdate 27-Dec-2003,

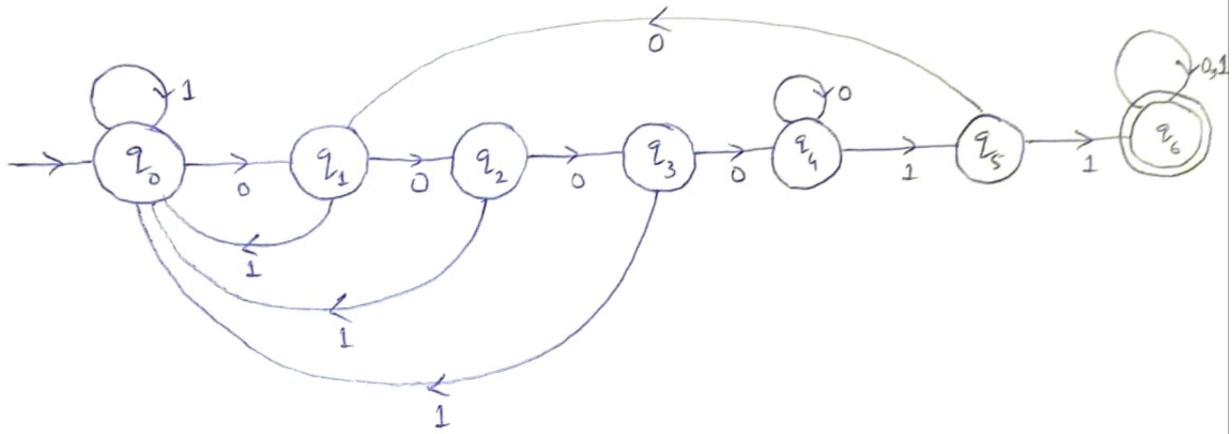
$$n = 12272003$$

$\therefore \quad m = n \pmod{64} = 3$

$\therefore$ $w = 000011$ is the 6-bit binary representation of m.

Now, $L_w := \{ x \in \{0,1\}^* \mid w \text{ is a substring of } x \}$

$\therefore$ A DFA that accepts L is shown below:



② Given, $\Sigma = \{0,1,2\}$ and $w \in \{0,1,2\}^*$ is a ternary representation of a number enc(w) (by dropping the leading zeros). Given language is

$$L = \{ w \in \Sigma^* \mid \text{enc}(w) \text{ is divisible by } 5 \}$$

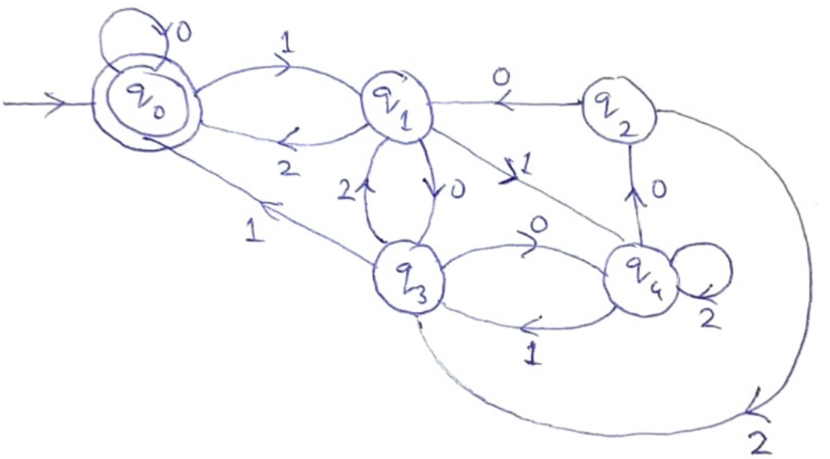
Let $Q = \{q_0, q_1, q_2, q_3, q_4\}$ be the states, where q_0, q_1, q_2, q_3, q_4 represent remainders 0, 1, 2, 3, 4 respectively, when divided by 5.

$\therefore$ The state q_0 is an accept state.

We can now construct a DFA from the transition function $\delta : Q \times \Sigma \to Q$ defined as below:

	0	1	2
q_0	q_0	q_1	q_2
q_1	q_3	q_4	q_0
q_2	q_1	q_2	q_3
q_3	q_4	q_0	q_1
q_4	q_2	q_3	q_4

The corresponding DFA is given below:



~~Here, $\delta(q_i, a)$ represents the remainder when $(3enc(w) + a)$ is divided by ~~3~~ 5, where w is the input string read so far and $enc(w)$ its ~~ternary~~ representation.~~

(3) For a language $L \subseteq \Sigma^*$,

$$pref(L) := \{ w \in \Sigma^* \mid ww' \in L \text{ for some } w' \in \Sigma^* \}$$

If L is regular, then then let $M = (Q, \Sigma, \delta, q_0, F)$ be the machine that recognizes L.

To show that pref(L) is regular, we find a DFA $\hat{M}$ that will accept it.

Let $\hat{M} = (Q, \Sigma, \delta, q_0, \hat{F})$ be the DFA that accepts pref(L), which is the same as that for L, except the set of accepting states $\hat{F}$. We say:

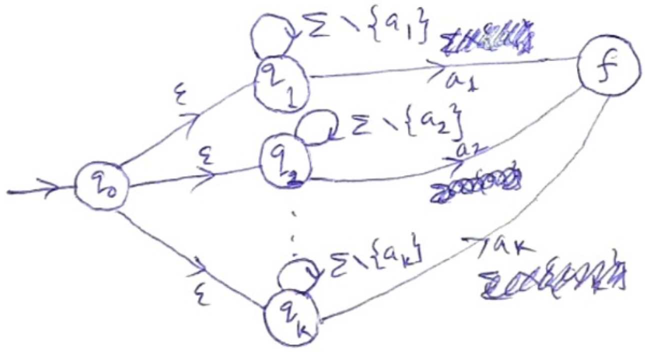
$$q \in \hat{F} \iff \exists \text{ a path from } q \text{ to an accepting state } f \text{ of } M.$$

i.e., there is a string $w' \in \Sigma^*$ so that $\delta(q, w') = f \in F$.

Clearly, this works because if $\delta(q_0, w) = q$, then $w \in \text{pref}(L)$ since $\delta(q_0, ww') = f \in L$; and conversely.

④ Let $\Sigma = \{a_1, a_2, \ldots, a_k\}$ be the input alphabet and let L consist of strings $w \in \Sigma^*$ such that the last symbol of w does not occur elsewhere in w; i.e., if $w \in L$ then $w = xa$ where $x \in (\Sigma \setminus \{a\})^*$.

An NFA that accepts L is shown below:



In the above NFA, $M = (Q, \Sigma, \delta, q_0, F)$:

where,

$Q = \{q_0, q_1, \ldots, q_k, f\}$, $F = \{f\}$

$\delta : Q \times \Sigma \cup \{\varepsilon\} \to 2^Q$ defined by

$$\delta(q_0, \varepsilon) = q_i \quad \forall \; i = 1, 2, \ldots, k$$

$$\delta(q_i, a) = \begin{cases} f & \text{if } a = a_i \\ \{q_i\} & \text{if } a \neq a_i \end{cases}$$

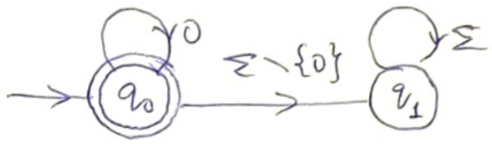
(5) Let $L = \{ 0^{k^2} \mid \forall \text{ positive integers } k \}$.

for $k=0$, $0^{k^2} = \varepsilon$, and for $k=1$, $0^{k^2} = 1$.

$\therefore 0^* \subseteq L^*$. Now any element of L^* is of the form 0^n, n positive integer. $\therefore L^* \subseteq 0^*$.

$\therefore L^* = 0^*$.

We give a DFA which shows that L^* is regular.



Here, $M = (Q, \Sigma, \delta, q_0, F)$

$Q = \{q_0, q_1\}$ is the set of states.

$F = \{q_0\}$

and $\delta : Q \times \Sigma \to Q$ is defined as:

$\delta(q_0, 0) = q_0$,

$\delta(q_0, \underset{\Sigma \setminus \{0\}}{\underset{\cap}{a}}) = q$

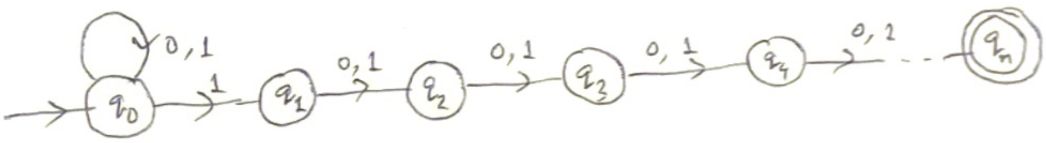
$\delta(q_1, \underset{\Sigma}{\underset{\cap}{b}}) = q_1$

$\therefore L^*$ is regular.

(6) Consider the regular language

$L = \{ w \in \{0,1\}^* \mid n^{th} \text{ digit from the right is } 1 \}$

An NFA that accepts L is:



Now, any DFA of this NFA has to remember the last n bits of the string. Since any bit can be 0 or 1, $\therefore$ any 2^n bit string can be an acceptable string.

Consider a DFA with # states $< 2^n$. Since there are 2^n n-bit strings. $\therefore$ By PHP, at least two n-bit strings, say $x = x_1 x_2 \cdots x_n$ and $y = y_1 y_2 \cdots y_n$, go to the same state from q_0;

$$\text{i.e., } \delta(q_0, x) = \delta(q_0, y) = q, \quad x \neq y.$$

$\because x \neq y$, $\therefore$ $x_i \neq y_i$ for some $i \in \{1, 2, \cdots, n\}$.

- If $x_1 \neq y_1$, we can assume WLOG that $x_1 = 1, y_1 = 0$. Then $x \in L$ but $y \notin L$, a contradiction, $\because$ both go to the state q.
- If $x_i \neq y_i$ for some $i > 1$, then again WLOG we can assume $x_i = 1, y_i = 0$.

  Then $\delta(q_0, x') = \delta(q_0, y') = \delta(q, \underbrace{00\cdots0}_{(i-1)\ 0's}) = q'$ (say).

  where x' and y' are extensions of x and y by $(i-1)$ 0's i.e.,

  $$x' = x_1 x_2 \cdots x_n \underbrace{0 0 \cdots 0}_{(i-1)\ 0's}, \quad y' = y_1 y_2 \cdots y_n \underbrace{00\cdots0}_{(i-1)\ 0's}.$$

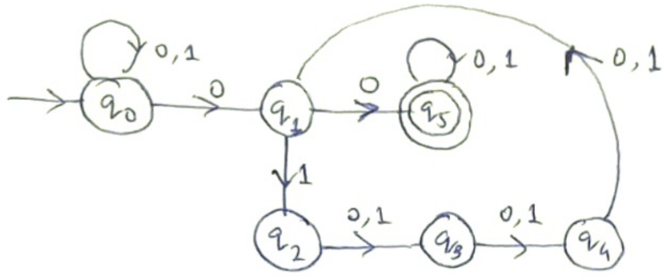
  Now, n^{th} bit of x' from the right is 1 but n^{th} bit of y' from the right is 0.

  $\therefore$ $x' \in L$ and $y' \notin L$, a contradiction, $\because$ both go to the same state q'.

$\therefore$ Any DFA for this must have at least 2^n states, i.e., we have a "small" NFA but a "large" DFA.

(7) $L \subseteq \{0,1\}^*$ such that L consists of all strings w such that there are two 0's in w separated by a number of positions that is a multiple of 4.

First we give an NFA that accepts L:



In the above NFA,

$$M = (Q, \Sigma, \delta, q_0, F) \;, F = \{q_5\}$$
$$Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$$
$\delta : Q \times \Sigma \to Q$ is defined as in the above NFA diagram.

Now the DFA that accepts L can be defined by:

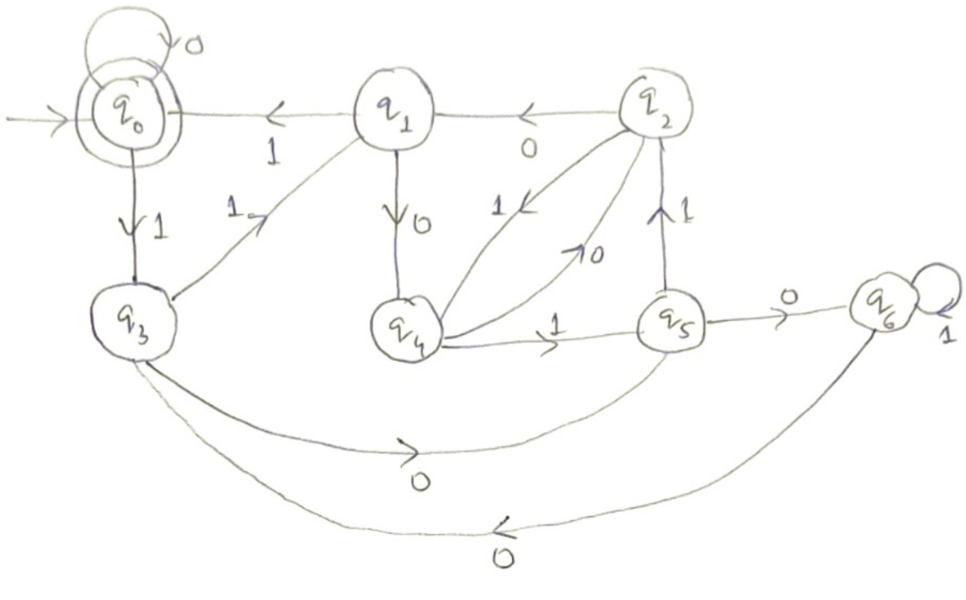
$$M' = (Q', \Sigma, \delta', q_0, F'),$$
$$Q' = 2^Q,$$
$$F = \{ S \in 2^Q \mid S \cap F \neq \phi \}$$
$\delta : Q' \times \Sigma \to Q'$ is defined as

$$\delta'(q, a) = \bigcup_{q \in S} \delta(q, a)$$

(8) Here, $L = \{ w \in \{0,1\}^* \mid enc(\hat{w}) \text{ is divisible by } 7 \}$.

The DFA for this is similar to that for $L = \{ w \in \{0,1\}^* \mid enc(w) \text{ is divisible by } 7 \}$ with just the transition function reversed.

It is shown below:



Here, $M = (Q, \Sigma, \delta, q_0, F)$

$Q = \{q_0, q_1, q_2, q_3, q_4, q_5, q_6\}$ are the states with q_i representing remainder i $(0 \le i \le 6)$ when divided by 7.

$F = \{q_0\}$, i.e., remainder 0.

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

$\delta : Q \times \Sigma \to Q$ as defined above.

(9) Given that L is regular. So there is a DFA $M = (Q, \Sigma, \delta, q_0, F)$ that accepts L.

Let $M_{min} = (Q_{min}, \Sigma, \delta_{min}, q_0, F)$ where

$Q_{min} = Q \cup \{q_{dead}\}$ with q_{dead} being a dead state and $q_{dead} \notin Q$. Consider the set of states

$$Q' = \{ q \in Q \mid \exists w \in \Sigma^*, f \in F \text{ s.t. } \delta(q, w) = f \}$$

and we define the transition function $\delta_{min} : Q_{min} \times \Sigma \to Q_{min}$ as:

$$\delta_{min}(q, a) = \begin{cases} \delta(q, a) & \text{if } q \notin Q' \cup \{q_{dead}\} \\ q_{dead} & \text{if } q \in Q' \cup \{q_{dead}\} \end{cases}$$

$\therefore$ $\min(L)$ is accepted by M_{min}, i.e., $\min(L)$ is regular.

In case of $\max(L)$, we can define

$$M_{max} = (Q, \Sigma, \delta, q_0, F_{max})$$

where $F_{max} = F \setminus Q'$. $\therefore$ $\max(L)$ is accepted by M_{max}, i.e., $\max(L)$ is regular.

(10) Given, L is regular. Thus, let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA that accepts L.

~~Since~~ To show that $L' = \{ww' \mid w \in L, |w'| = k\}$ is regular, we need to find a DFA M' that accepts L'.

~~We can~~ Let $M' = (Q', \Sigma, \delta', q_0, F')$ where

$Q' = Q \cup \{q_1, q_2, \dots, q_k\}$, $F' = \{q_k\}$,

$\delta' : Q' \times \Sigma \to Q'$ is defined as

$$\delta'(q, a) = \begin{cases} \delta(q, a) & \text{if } q \in Q \setminus F \\ q_1 & \text{if } q \in F \\ q_{i+1} & \text{if } q \in \{q_1, \dots, q_k\} \\ q_{k+1} & \text{if } q = q_{k+1} \end{cases}$$

The diagram of the DFA is shown below: