pgf@stop

THEORY OF COMPUTATION - CS340

2022-09-08

CS340 Assignment1

Prof. Manindra Agarwal

Name: Jaya Gupta
Roll Number: 200471

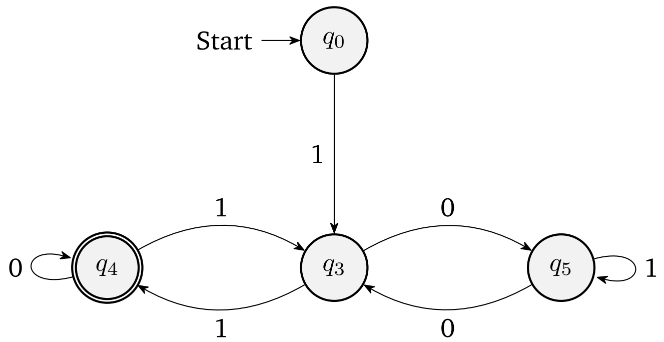
Name: Harshit Bansal
Roll Number: 200428

Name: Avi Bansal
Roll Number: 200229

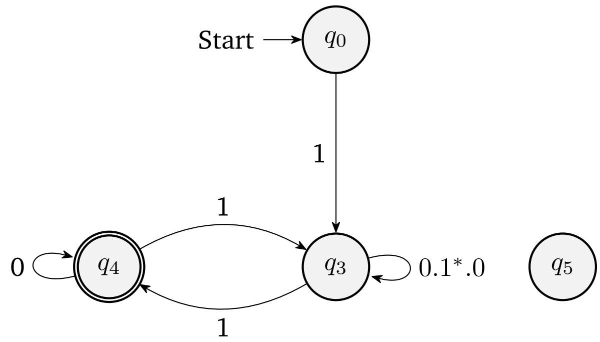
Q1. Language Accepted by DFA

To find the regular expression for the given DFA, we follow the recursive approach of removing the states one by one, till only start and final states are left.

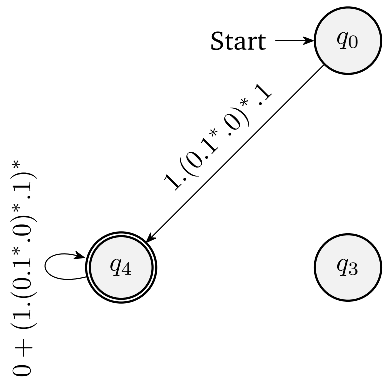
- Remove $q1$ and q_2: Now q_1 is final state and it accepts only set $\{0\}$. We can safely remove both theses states as they do not interfere with the rest of DFA.



- Remove q_5: New transitions are:



- Remove q_3:



Hence the language accepted by this DFA is union of those accepted by states q_1 and q_4.

$$r = 0 + (1.(0.1^*.0)^*.1).(0 + (1.(0.1^*.0)^*.1)^*)^*$$

where $L(r)$ is the regular set accepted by DFA.

Q2. If L is regular, oddL is also regular

Q3. Minimum State DFA (Minimization)

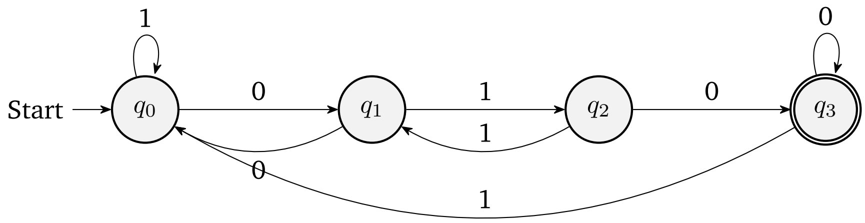


Figure: **Minimal DFA**

- Remove all the nodes unreachable from the start state. In this question, those nodes are q_4, q_5, q_6, and q_7.
- Lets make the transition table for each state.

Transition Table		
State	0	1
q_0	q_1	q_0
q_1	q_0	q_2
q_2	q_3	q_1
q_3	q_3	q_0

- Combine all the sinks (i.e. the nodes that can't reach to final state) into one. Here there are no sinks. We can reach the final state from all the states through some transition.
- Merge non-distinguishable states or states belonging to same equivalence classes whereby every two states of the same equivalence class are equivalent if they have the same behavior for all the input sequences.
- That is, for all states p_1, p_2 belonging to the same equivalence clas E, the transition taken on reading input w should be equivalent for both - transition to state that both rejects or accepts.
- Divide the states into accepting and rejecting.

$$R = \{q_0, q_1, q_2\}$$

$$A = \{q_3\}$$

- Since q_0 and q_1 on reading 0 or 1 remains in rejecting state, whereas q_2 on reading 0 reaches the accepting state. Hence behavior of q_2 is different from q_0 and q_1.

$$R_1 = \{q_0, q_1\}$$

$$R_2 = \{q_2\}$$

$$A = \{q_3\}$$

- q_0 on reading 1 remains in $R1$ only, whereas q_1 on reading 1 makes transition to R_2. By the definition of equivalence classes, if p and q are two strings in same equivalence class E, so if any extension of $p(pw)$ is in E, then qw will also be in E.

$$\forall p, q \in \text{E}, pw \in E \text{ iff } qw \in E$$

- q_0 and q_1 does not follow this property, hence they will be in different equivalence class.

$$R_1 = \{q_0\}$$

$$R_2 = \{q_1\}$$

$$R_3 = \{q_2\}$$

$$A = \{q_3\}$$

Hence in the final minimal DFA, there will be four states q_0, q_1, q_2, q_3.

Q4. L_1 and L_2 are regular. Prove Mix(L_1, L_2) is regular.

Mix$(L_1, L_2) = \{w \in \sum^* | w = x_1y_1x_2y_2......x_ky_k$ where $x_1x_2....x_k \in L_1$ and $y_1y_2......y_k \in L_2$, each $x_i, y_i \in \sum^*\}$

We need to show that if L_1 and L_2 are regular implies that Mix(L_1, L_2) is also regular.

Construction of DFA of Mix(L_1, L_2)

- Let $\mathcal{F}_1$ and $\mathcal{F}_2$ are DFA of L_1 and L_2 respectively.

$$\mathcal{F}_1 = \{q_0^1, Q_1, \sum, \delta_1, F_1\}$$
$$\mathcal{F}_2 = \{q_0^2, Q_2, \sum, \delta_2, F_2\}$$

- Let $\mathcal{F}_m$ be the NFA of Mix$(L_1, L2)$, such that

$$\mathcal{F}_m = \{\{q_0^1, q_0^2\}, Q_1 \times Q_2, \sum, \delta_m, F_m\}$$

  where

$$F_m = F_1 \times F_2$$
$$\delta_m : \delta_m(\{q_1, q_2\}, a) = \{\delta_1(q_1, a), q_2\}$$
$$OR$$
$$\delta_m : \delta_m(\{q_1, q_2\}, a) = \{q_1, \delta_2(q_2, a)\}$$

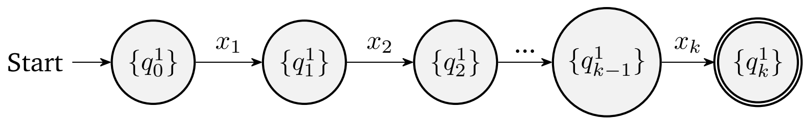
- Since the transition from a single state $\{q_1, q_2\}$ on reading input symbol a can be to two different states, $\mathcal{F}_m$ is a NFA.

Proof: Mix(L_1, L_2) **is accepted by** $\mathcal{F}_m$

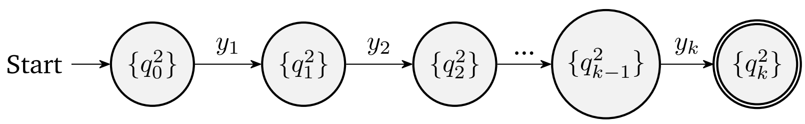
- Inorder to show that Mix(L_1, L_2) is accepted by NFA, we need to show that there exists at least one transition which accepts Mix(L_1, L_2).

  w=x$_1y_1x_2y_2......x_ky_k$ where $x_1x_2....x_k \in L_1$ and $y_1y_2......y_k \in L_2, w \in Mix(L_1, L_2)$

  The corresponding transition in L_1 and L_2 are

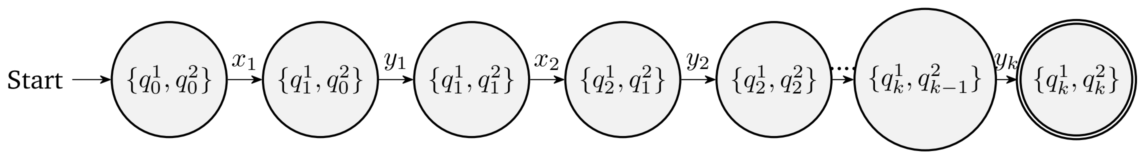


Transition in $\mathcal{F}_1$



Transition in $\mathcal{F}_2$

  Consider the following transition, which is one possible transition in the above NFA $\mathcal{F}_m$



One

Transition in $\mathcal{F}_m$

  So this transition accepts w.

  Note: Here the transition on any input is $q_i^j \rightsquigarrow q_{i+1}^k$. Means there are multiple transition involved between any two states. (Shown like that because was unable to show squiggly arrow in DFA diagram).

Proof2: $\mathcal{F}_m$ accepts strings which are in Mix(L_1, L_2) only.

- Let the string accepted by $\mathcal{F}_m$ be

$$w = m_1 m_2 m_3 m_4 .... m_k | m_i \in \sum$$

- Consider two empty strings A and B.

$$A = \epsilon$$

$$B = \epsilon$$

- We will concat a alphabet m_j in string A if after reading m_j, $\mathcal{F}_m$ takes transition of the type

$$\delta_m : \delta_m(\{q_1, q_2\}, a) = \{\delta_1(q_1, a), q_2\}$$

  means, the transition is taken is $\mathcal{F}_1$ and the state in $\mathcal{F}_2$ remains same.

- We will concat a alphabet m_j in string B if after reading m_j, $\mathcal{F}_m$ takes transition of the type

$$\delta_m : \delta_m(\{q_1, q_2\}, a) = \{q_1, \delta_2(q_2, a)\}$$

  means, the transition is taken is $\mathcal{F}_2$ and the state in $\mathcal{F}_1$ remains same.

- Since w is accepted by $\mathcal{F}_m$(final state $= F_1 \times F_2$), means string A takes $\mathcal{F}_1$ to state in F_1 (final state), and string B takes $\mathcal{F}_2$ to state in F_2(final state).

$$A \in L_1$$

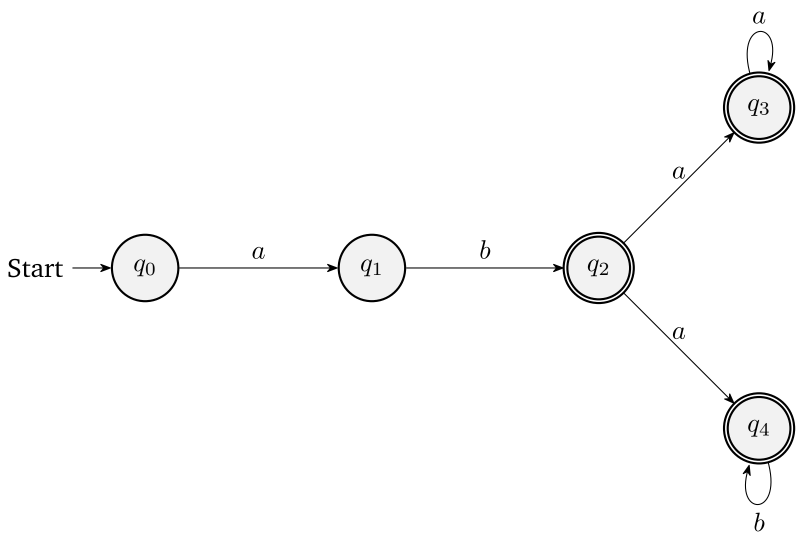
$$B \in L_2$$

- Hence this concludes that string $w \in Mix(L_1, L_2)$. It is a mix of string in L_1 and L_2.

Q5. Design NFA

$$L = \{w \mid w \text{ is } abab^n \text{ or } aba^n \text{ where } n \geq 0\}$$

NFA accepting above language is:



Note: The transitions not shown for any state for any alphabet are assumed to go into sink.

Q6. Regular or Non-Regular

The strings in language L will have difference between number of $0's$ and $1's$ equal to 0 or 1. Now consider following strings,

$$x = 0^{m_1},\ y = 0^{m_2},\ \text{where } m_1, m_2 \in N \text{ and } m_1 - m_2 \geq 2$$

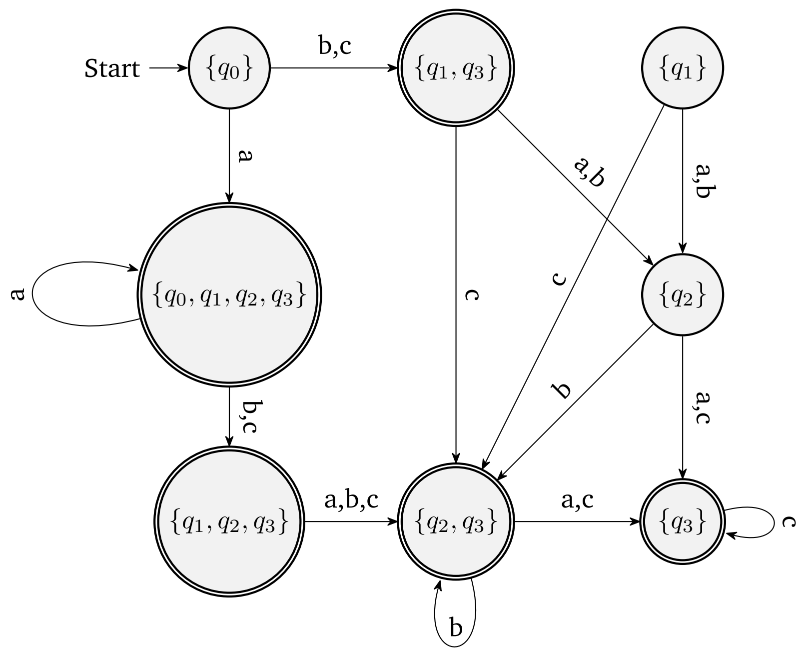
Now take $w = 1^{m_1}$. Observe that $xw = 0^{m_1}1^{m_1}$ will belong in L, whereas $yw = 0^{m_2}1^{m_1}$ wont belong in L because difference in number of $0's$ and $1's$ is more than 1. Hence by definition, x and y will belong to different equivalence classes. Also we can generate infinite number of m such that when taken any two at a time, x and y will lie in different equivalence classes. Hence there will be infinite equivalence classes for the given language L, which proves this is **not** a regular language.

Q7. Convert NFA to DFA.

- Lets make the transition table containing all the states a particular state can reach to on reading a symbol.

Transition Table			
State	a	b	c
q_0	$\{q_0, q_1, q_2, q_3\}$	$\{q_1, q_3\}$	$\{q_1, q_3\}$
q_1	$\{q2\}$	$\{q2\}$	$\{q_2, q_3\}$
q_2	$\{q_3\}$	$\{q_2, q_3\}$	$\{q_3\}$
q_3	Sink	Sink	$\{q_3\}$
$\{q_1, q_3\}$	$\{q_2\}$	$\{q_2\}$	$\{q_2, q_3\}$
$\{q_2, q_3\}$	$\{q_3\}$	$\{q_2, q_3\}$	$\{q_3\}$
$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$	$\{q_1, q_2, q_3\}$	$\{q_1, q_2, q_3\}$
$\{q_1, q_2, q_3\}$	$\{q_2, q_3\}$	$\{q_2, q_3\}$	$\{q_2, q_3\}$

- Convert the above transition table into DFA.



Note: All the remaining transitions takes the state to a sink state.
Note: All the remaining states from 2^Q are not reachable from the start state. Hence Minimal DFA above does not contain them.

Q8. Let L be the language $L = \{w \in \{a, b\}^* | w$ contains equal number of occurences of ab and ba $\}$.

(a) **Regular Expression**

- The string will be of four types.
  - Beginning with a and ending with a.

    aaa....abbbbb....baaaa....abbb...baaa
  - Beginning with a and ending with b.

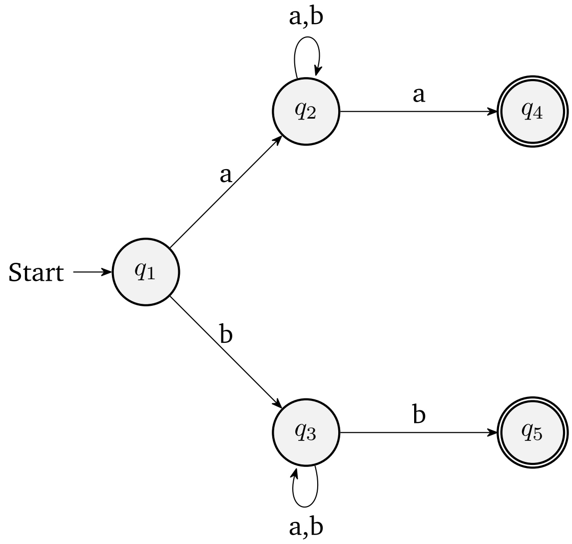
    aaa....abbbbb....baaaa....abbb
  - Beginning with b and ending with b.

    bbb....baaaaa.....abbbbb......baaa....abbb
  - Beginning with b and ending with a.

    bbb....baaa......abbbbb....baaa.......aaa
- The types of string with equal number of occurences of ab and ba will be the strings will be the one starting and ending with the same letter (either a or b). Regualar expression will be

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

(b) **DFA /NFA /ϵ−NFA**



Note: All the remaining transitions takes the state to a sink state.

Q9. DFA state-minimization

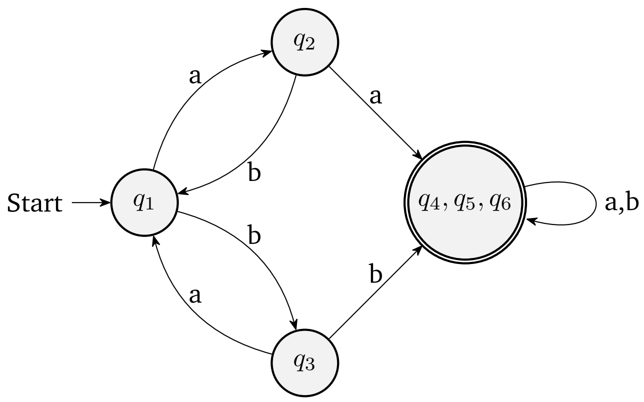


Figure: **Minimal DFA**

- Remove all the nodes unreachable from the start state. In this question, there are no such nodes.
- Lets make the transition table for each state.

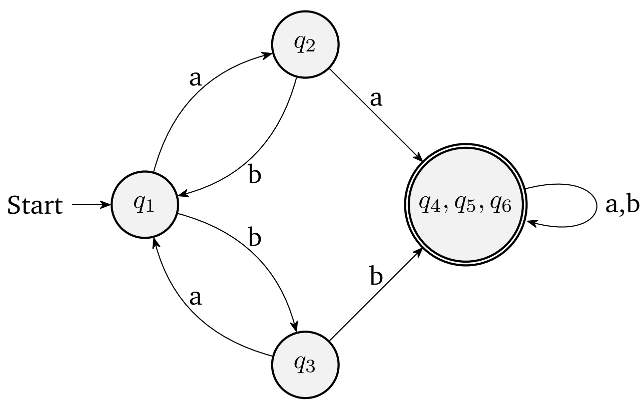
Transition Table		
State	a	b
q_1	q_2	q_3
q_2	q_4	q_1
q_3	q_1	q_5
q_4	q_6	q_4
q_5	q_5	q_6
q_6	q_6	q_6

- Combine all the sinks (i.e. the nodes that can't reach to final state) into one. Here there are no sinks. We can reach the final state from all the states through some transition.
- Merge non-distinguishable states or states belonging to same equivalence classes whereby every two states of the same equivalence class are equivalent if they have the same behavior for all the input sequences.
- That is, for all states p_1, p_2 belonging to the same equivalence class E, the transition taken on reading input w should be equivalent for both - transition to state that both rejects or accepts.
- Divide the states into accepting and rejecting.

$$R = \{q_1, q_2, q_3\}$$

$$A = \{q_4, q_5, q_6\}$$

- Since q_4, q_5, q_6 on either reading 0 or 1, remains in the accepting state, they belong to the same equivalence class. Hence we merge them into one state.



- We can clearly see that the behavior of q_1, q_2 and q_3 are different on reading a or b.
  - (q_2, q_3) : q_2 on reading a goes to the final state whereas q_3 remains in rejecting state. q_3 on reading b goes to final state whereas q_2 not. Hence, their behaviors are different.
  - (q_1, q_2) or (q_1, q_3) : q_2 or q_3 on reading either a or b goes to the final state, whereas q_1 on reading both a and b remains in the rejecting state. Hence behavior of q_1 differs from both q_2 and q_3.

Hence in the final minimal DFA, there will be four states $\{q_1, q_2, q_3, \{q_4, q_5, q_6\}\}$.

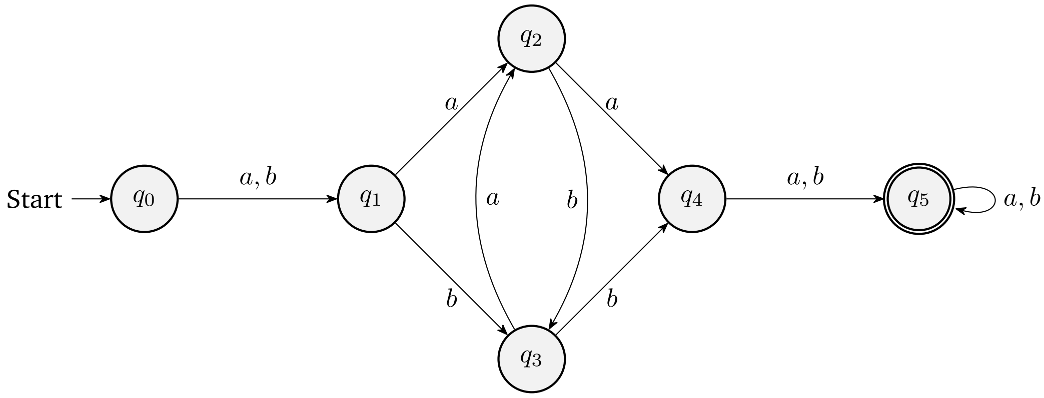
Q10. Regular Expression For Reverse

$$L = \{uvv^r w \mid u,\, v,\, w \in \{a, b\}^+\}$$

The regular expression which has language L is:

$$r = (a+b)(a+b)^*.(aa+bb).(a+b)^*.(a+b)$$

The equivalent NFA for above regular expression is:



Q13. Given that language L is regular, is the language $L_{\frac{1}{2}} = \{x | \exists y$ such that $|x| = |y|, xy \in L\}$ regular? If yes, give a formal proof.

Q15. DFA for no consecutive identical symbols in the string.

$$L = \{s | s \in (0,1,2)^* \text{ and } \{00, 11, 22\} \notin substring(s)\}$$

DFA design that accepts L is