Minimization of DFA

Raju Hazari

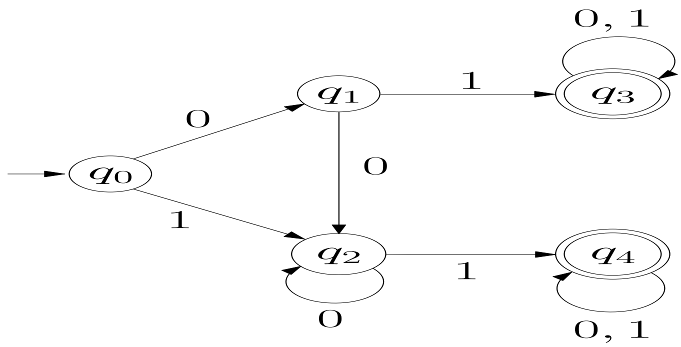
Department of Computer Science and Engineering
National Institute of Technology Calicut

November 5, 2020

Proof of Myhill-Nerode Theorem

- We stated with a machine M and defined a equivalence relation R_M on M. R_M was a finite index, it is right invariant and L is a union of some of the equivalence classes of R_M.
- Then we saw that if we take R_L, R_M will be a refinement of R_L. So, the index of R_L or number of equivalence classes in R_L will be less than equal to the number of equivalence classes in R_M.
- Then from R_L, we constructed a DFA $M^{'}$. Corresponding to each equivalence class of R_L we have a state in $M^{'}$. Equivalence class which contains the empty string it corresponds to start state. If x in L then an equivalence class corresponding to that will be a final state.

Minimization of DFA



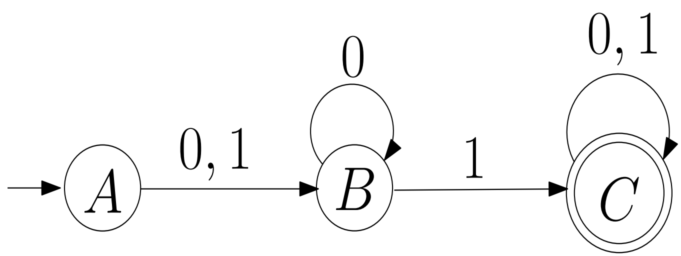
- This is a state diagram of DFA. Here, the alphabet $\Sigma^* = \{0, 1\}$
- Now, consider this as a machine M induces an equivalence relation (R_M) on Σ^*.
- In R_M we have define the set of strings which take you from initial state to a particular state and the each of corresponding to a equivalence class. So, there is five equivalence classes, J_0, J_1, J_2, J_3 and J_4.
  - J_0 is a equivalence class which contains a set of strings which take you from q_0 to q_0
  - J_1 is a equivalence class which contains a set of strings which take you from q_0 to q_1

Minimization of DFA

- L contains two class, i.e., J_3 and J_4, because these two are final state. So, L is a union of these two equivalence classes, $L = J_3 \cup J_4$
- Now, consider R_M will be refinement of R_L. So, some of these equivalence classes can be merged in R_L. See which of them can be merged.
- First consider two equivalence class J_1 and J_2, and two string x and y. Consider x belongs to J_1 and y belongs to J_2. If I take a z, and if z contain just one 0 then both xz and yz will not be accepted. If z contains just one 1, both of them will be accepted. So, whatever may be z, xz and yz both will be accepted or both will be rejected. So, in R_L we can put J_1 and J_2 in same equivalence class.
- Next consider two equivalence class J_0 and J_1, and two string x and y. Consider x belongs to J_0 and y belongs to J_1. If I take a z, and if z contain just one 1, then yz will be accepted but xz not be accepted. So, in R_L we can not merge J_0 and J_1 in same equivalence class.

Minimization of DFA

- Similarly, we can see that in R_L, we can merge J_3 and J_4 in same equivalence class. But we can not merge J_1 and J_3 in same equivalence class.
- So, in R_L there are only three equivalence class J_0, $J_1 \cup J_2$ and $J_3 \cup J_4$.
- Now, from this R_L when we construct a DFA M', it has three state. Let us consider the three states are A, B and C respectively.
- Equivalence state J_0 contains the empty string, so A is the initial state.



This is the mimimum state automata.

Minimization of DFA

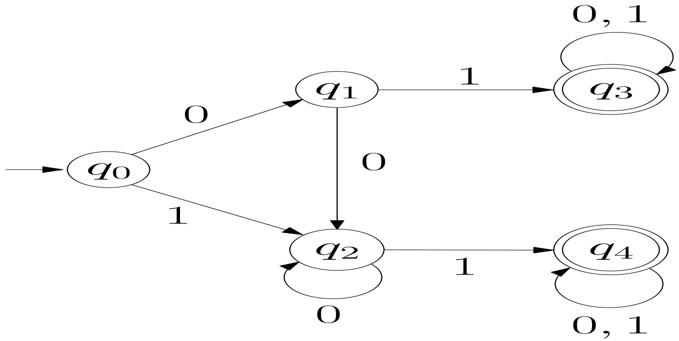
- **Definition:**

  Two states p and q of a DFA are called **indistinguishable** if $\delta^*(p, w) \in F$ implies $\delta^*(q, w) \in F$, and $\delta^*(p, w) \notin F$ implies $\delta^*(q, w) \notin F$, for all $w \in \Sigma^*$.

  On the other hand, if there exists some string $w \in \Sigma^*$ such that $\delta^*(p, w) \in F$ and $\delta^*(q, w) \notin F$ or vice versa, then the states p and q are said to be **distinguishable** by a string w. Here, w is the distinguishing sequence.

Minimization of DFA

- **Example1:**



- We have $q_0\, q_1\, q_2\, q_3\, q_4$ in one block
- Separate them into two block, having final and non-final states.

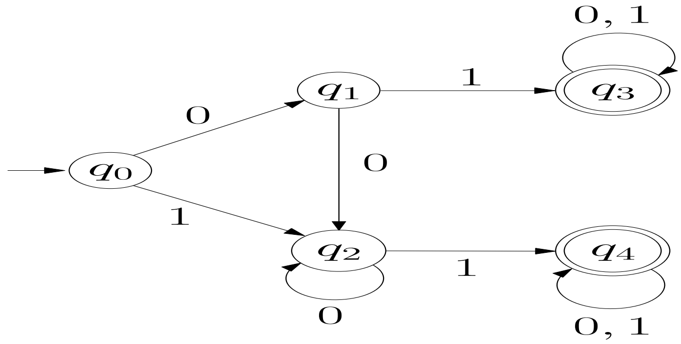
$$\overline{q_0\, q_1\, q_2} \qquad \overline{q_3\, q_4}$$

ϵ take you from each one of them to a final state and also ϵ take you from each one of them to a non-final state. So, ϵ is the distinguishable sequence.

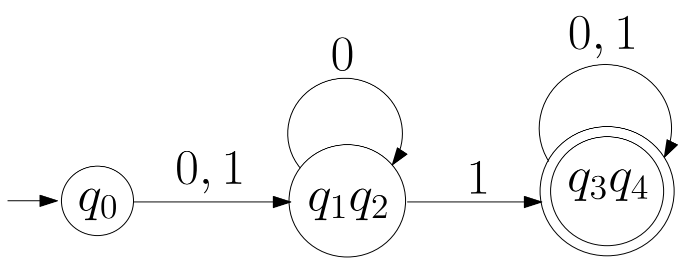
- Again separate first block into two block (based on 0-successor or 1-successor)

$$\overline{q_0} \qquad \overline{q_1\, q_2} \qquad \overline{q_3\, q_4}$$

Minimization of DFA

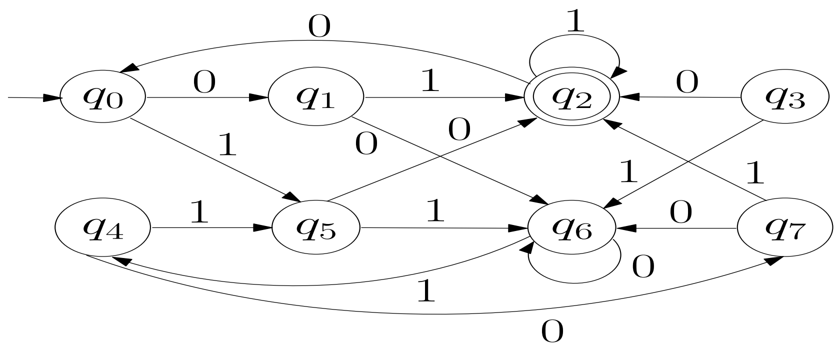


The corresponding minimum state automata:



Minimization of DFA

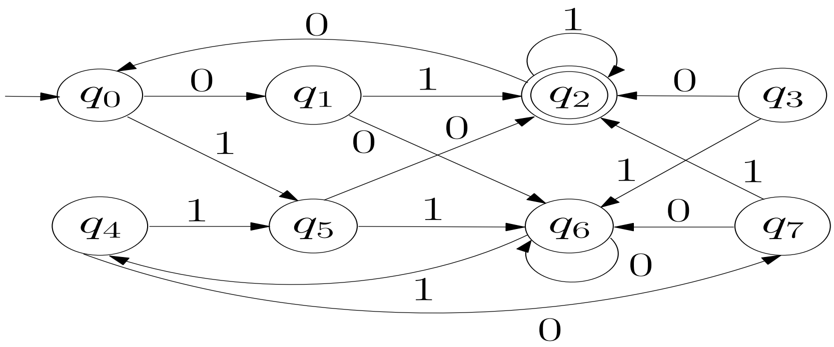
- **Example2:**



- We have $q_0\, q_1\, q_2\, q_3\, q_4\, q_5\, q_6\, q_7$ in one block
- Separate them into two block, having final and non-final states.

  $\overline{q_0\, q_1\, q_3\, q_4\, q_5\, q_6\, q_7}$ $\quad$ $\overline{q_2}$
- $\overline{q_3\, q_5}$ $\quad$ $\overline{q_0\, q_1\, q_4\, q_6\, q_7}$ $\quad$ $\overline{q_2}$
- $\overline{q_3\, q_5}$ $\quad$ $\overline{q_0\, q_4}$ $\quad$ $\overline{q_1\, q_7}$ $\quad$ $\overline{q_6}$ $\quad$ $\overline{q_2}$

Minimization of DFA



The corresponding minimum state automata: