Theory of Computation: CS-202

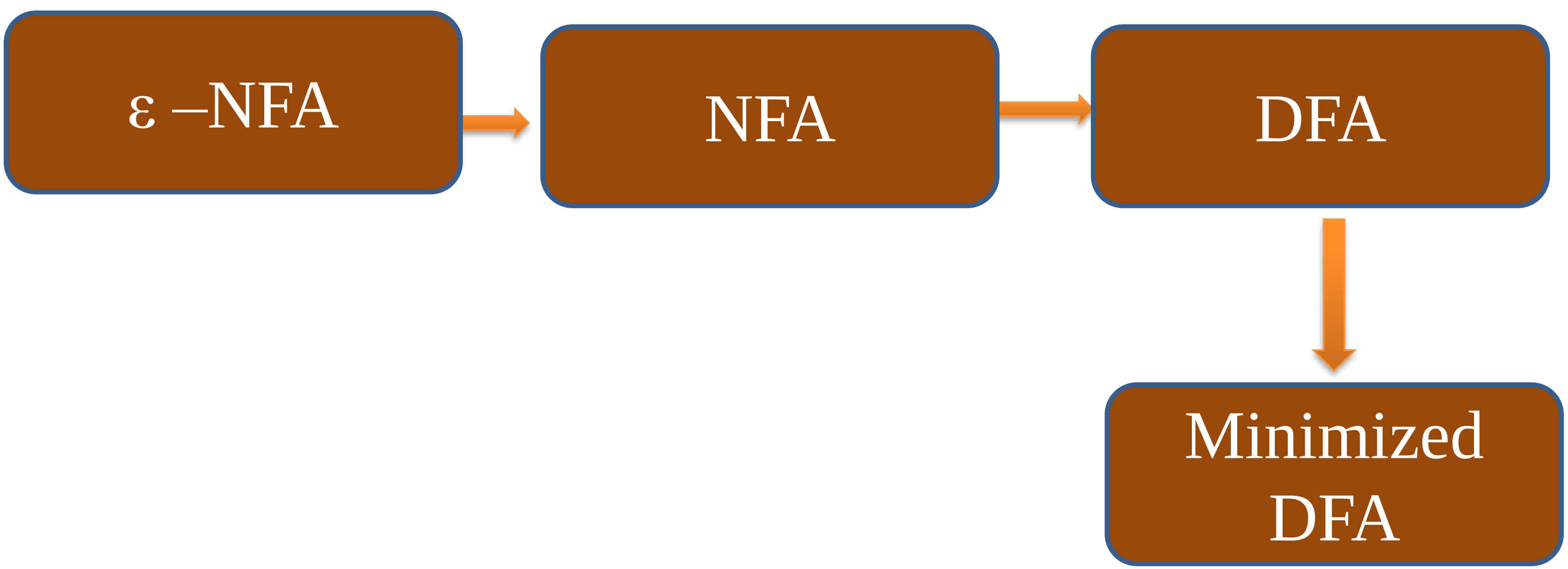
Deterministic Finite Automata Minimization

Outline

Deterministic Finite Accepters (DFA)

- Minimization of DFA
  - Equivalence Theorem or Set Partitioning method
  - Myhill- Nerode Theorem

Finite Automata



Minimization of DFA

- Minimization of DFA means reducing the number of states from the given DFA.

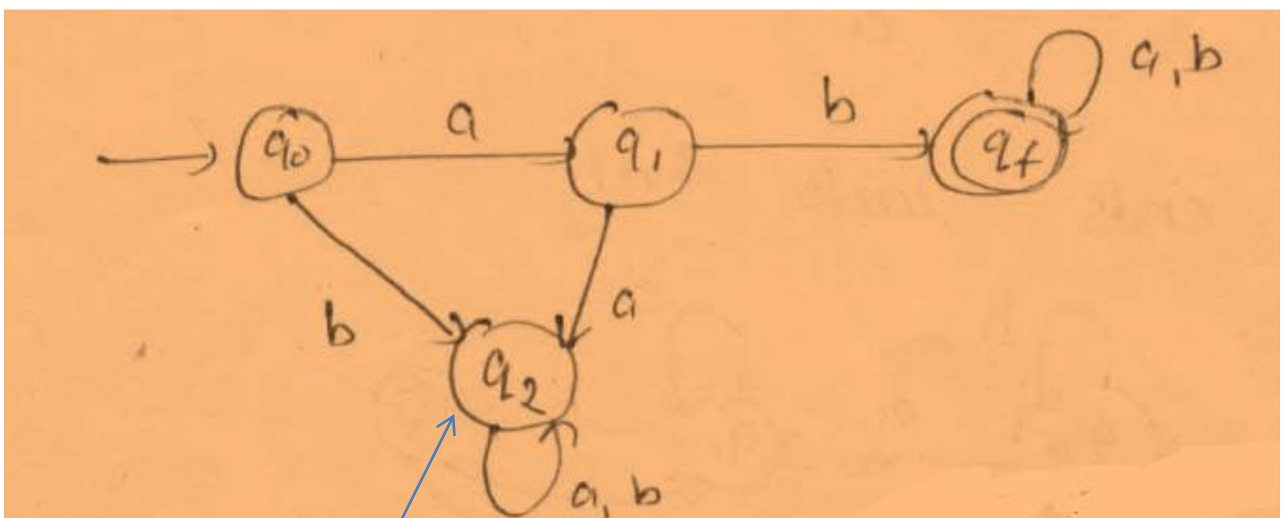
Minimization of DFA

DFA minimization is possible by considering following states:

1. Dead states
2. Unreachable states
3. Equal or indistinguishable states

Minimization of DFA

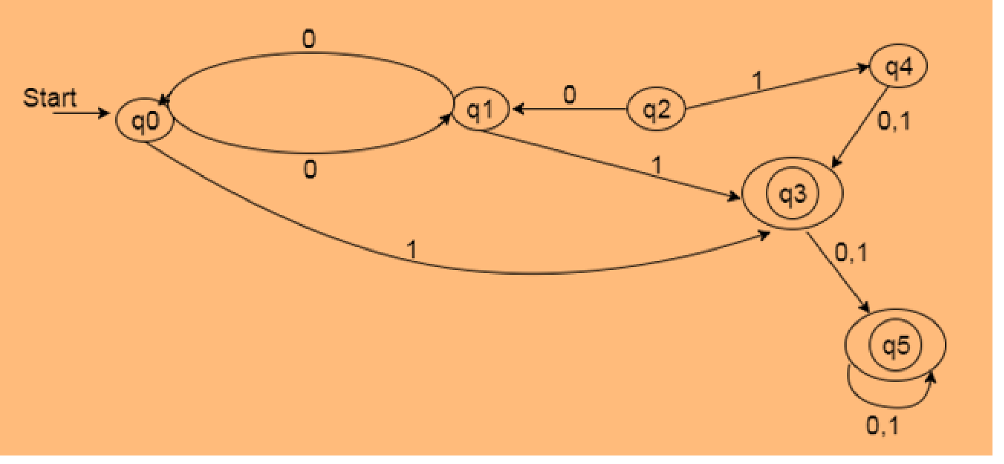
- **Dead states:** A dead state is non accepting state whose transitions for every input symbols terminate on themselves.
- There is no way for it to reach a final state.
- Merge all the dead states of DFA.



Dead state

Minimization of DFA

- **Unreachable states:** Unreachable states are the states that are not reachable from the initial state of the DFA for any input string.
- Remove unreachable states from DFA.



q2 and q4 are unreachable

Minimization of DFA

- **Equal or indistinguishable states:** Two states (q_i, q_j) of a DFA are said to be equal or indistinguishable if

  $\delta^*(q_i, w) \in F \Rightarrow \delta^*(q_j, w) \in F$

And

  $\delta^*(q_i, w) \notin F \Rightarrow \delta^*(q_j, w) \notin F$

$\forall$ $w \in \Sigma^*$, q_i, q_j are called equal or indistinguishable.

$\exists$, $w \in \Sigma^*$ such that

$\delta^*(q_i, w) \in F \Rightarrow \delta^*(q_j, w) \notin F$

Then q_i, q_j are called unequal or distinguishable.

Note: merge the equal states.

Myphill-Nerode Theorem or Table filling Method

Minimization of DFA using Myphill-Nerode Theorem or Table filling Method

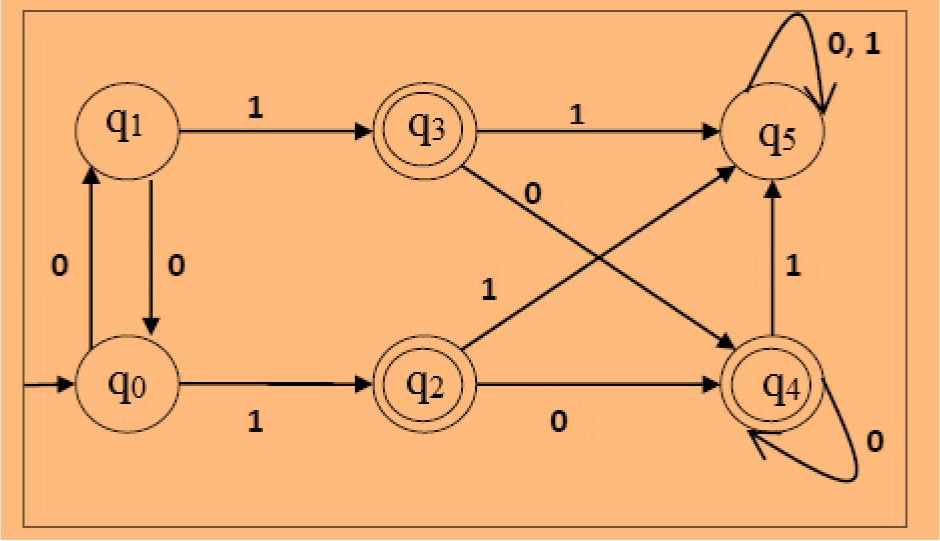
Minimization of DFA means reducing the number of states from given FA.

We have to follow the various steps to minimize the DFA. These are as follows:

- **Step 1** − Draw a table for all pairs of states (q_i, q_j) not necessarily connected directly [All are unmarked initially]
- **Step 2** − Consider every state pair (q_i, q_j) in the DFA where $q_i \in F$ and $q_j \notin F$ or vice versa and mark them. [Here F is the set of final states]
- **Step 3** − Repeat this step until we cannot mark anymore states −

  If there is an unmarked pair (q_i, q_j), mark it if the pair $\{\delta (q_i, X), \delta (q_i, X)\}$ is marked for some input alphabet.
- **Step 4** − Combine all the unmarked pair (q_i, q_j) and make them a single state in the reduced DFA.

Example: Minimization the DFA using Myphill-Nerode Theorem or Table filling Method

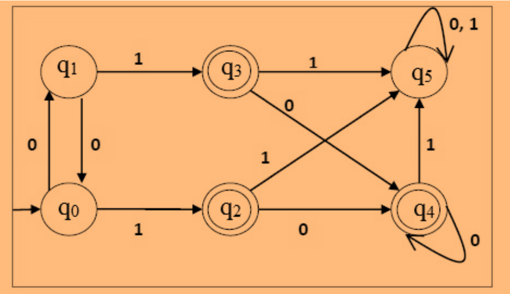


Step 1 – Draw a table for all pair of states.

	q0	q1	q2	q3	q4	q5
q0						
q1						
q2						
q3						
q4						
q5						

Step 2 – We mark the state pairs.

	q0	q1	*q2	*q3	*q4	q5
q0						
q1						
*q2	✓	✓				
*q3	✓	✓				
*q4	✓	✓				
q5			✓	✓	✓	



Step 3- If we input 1 to state ‘q0’ and ‘q5’, it will go to state ‘q2’ and ‘q5’ respectively.
Or $\delta(q_0, 1)$=q2, $\delta(q_5, 1)$=q5, (q2, q5) is already marked, hence we will mark pair (q0, q5).

Now, we input 1 to state ‘q1’ and ‘q5’; it will go to state ‘q3’ and ‘q5’ respectively. (q3, q5) is already marked, hence we will mark pair (q1, q5).

	q0	q1	*q2	*q3	*q4	q5
q0						
q1						
*q2	√	√				
*q3	√	√				
*q4	√	√				
q5			√	√	√	

	q0	q1	*q2	*q3	*q4	q5
q0						
q1						
*q2	√	√				
*q3	√	√				
*q4	√	√				
q5	√	√	√	√	√	

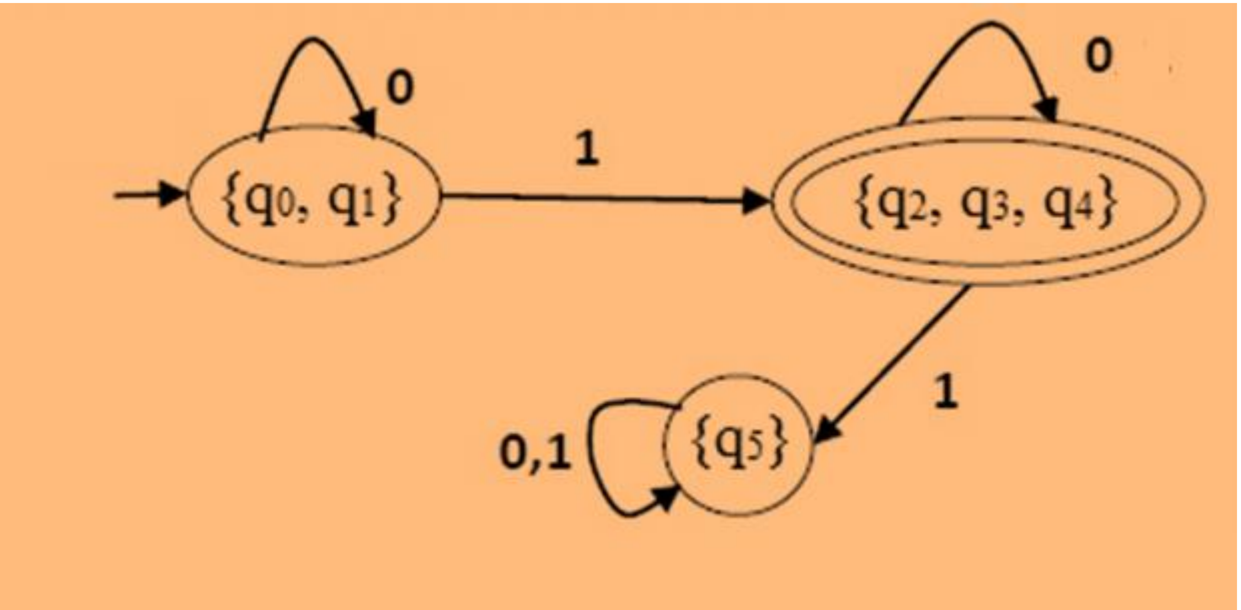
Step 4- After step 3, we have got state combinations {q0, q1} {q2, q3} {q2, q4} {q3, q4} that are unmarked.

We can recombine {q2, q3} {q2, q4} {q3, q4} into {q2, q3, q4}

Hence we got two combined states as {q0, q1} and {q2, q3, q4}

Transition Diagram of Minimized DFA

States (q)	$\delta(q_i, 0)$	$\delta(q_i, 1)$
→ $\{q_0, q_1\}$	$\{q_0, q_1\}$	{q2 , q3, q4}
*$\{q_2, q_3, q_4\}$	*$\{q_2, q_3, q_4\}$	q_5
q_5	q5	q5



Suggested readings

1. An introduction to FORMAL LANGUAGES and AUTOMATA by PETER LINZ.
2. Introduction to Automata Theory, Languages, And Computation by JOHN E. HOPCROFT, RAJEEV MOTWANI, JEFFREY D. ULLMAN
3. Theory of computer science: automata, languages and computation **by** K.L.P MISHRA

Thank you