Theory of Computation: CS-202

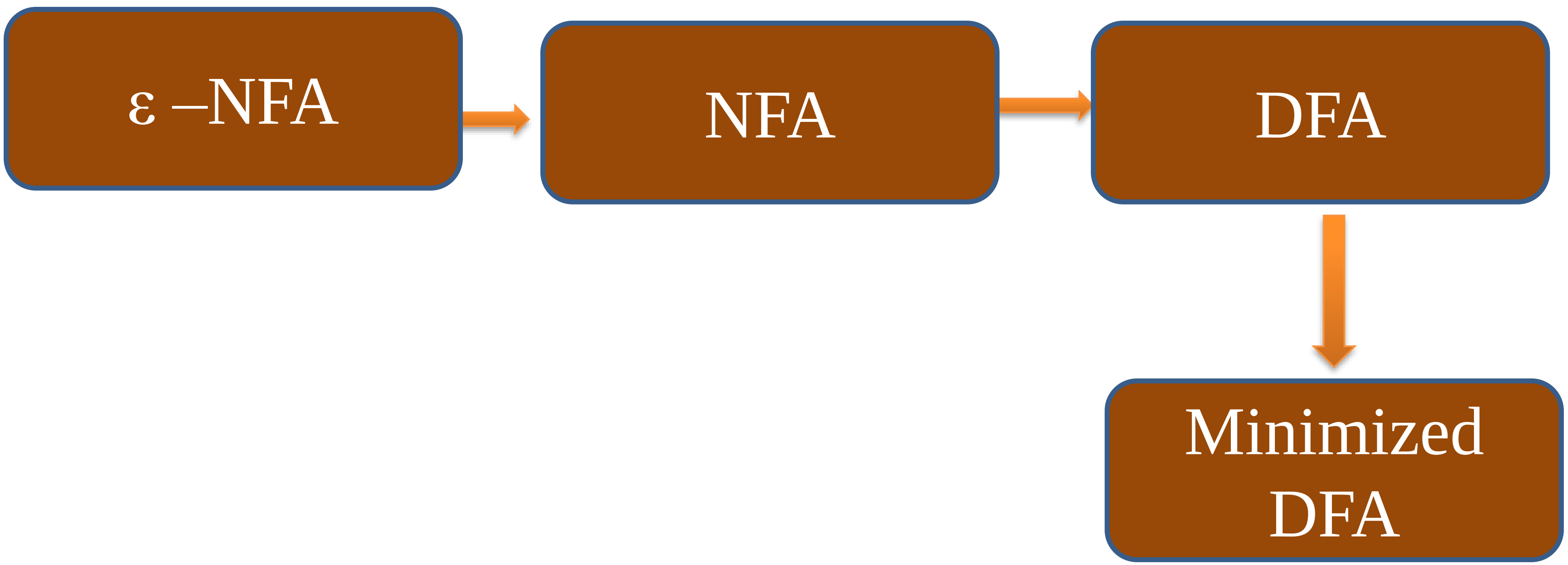
Deterministic Finite Automata Minimization

Outlines

Deterministic Finite Accepters (DFA)

- ❑ Minimization of DFA
  - ➢ Equivalence Theorem or Set Partitioning method

Finite Automata



Minimization of DFA

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

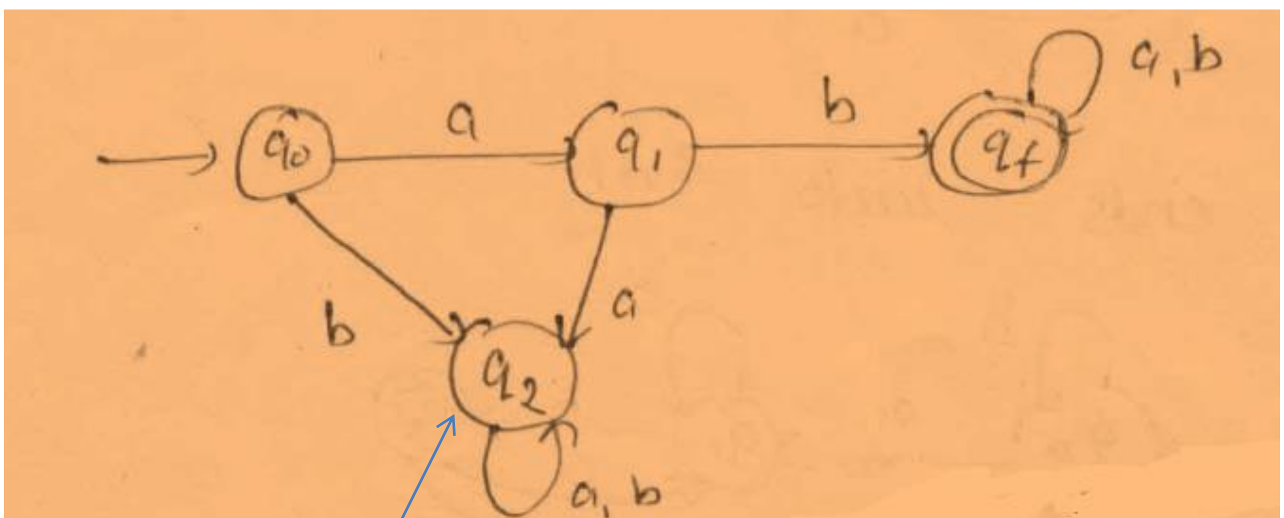
Minimization of DFA

DFA minimization is possible by considering following states:

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

Minimization of DFA

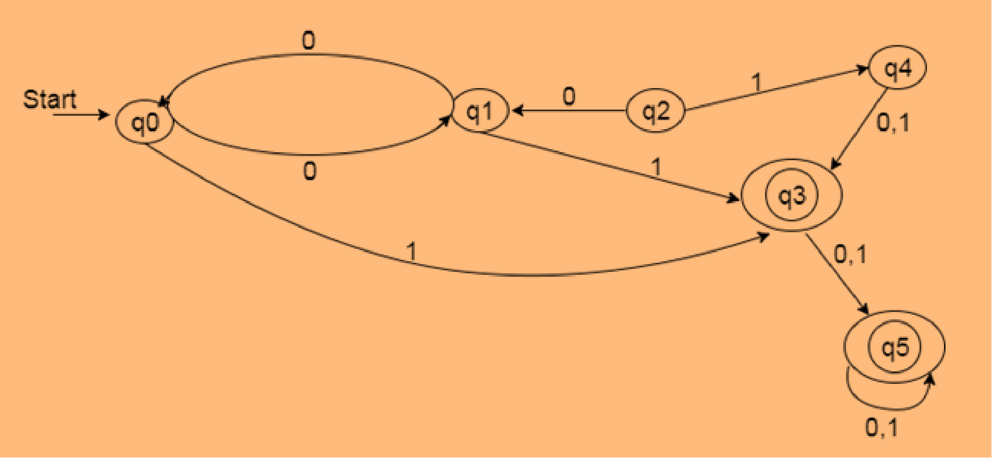
- **Dead states:** A dead state is non accepting state whose transitions for every input symbols terminate on themselves.
- There is no way for dead state to reach a final state.
- Merge all the dead states of DFA.



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

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

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

$\exists$, $w \in \sum^*$ 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.

DFA Minimization using Equivalence Theorem or Set Partitioning method

If q_i and q_j are two states in a DFA, we can combine these two states into $\{q_i, q_j\}$ if they are equal.

Two states are distinguishable, if there is at least one string 'w', such that one of $\delta(q_i, w)$ and $\delta(q_j, w)$ is accepting and another is not accepting.

Hence, a DFA is minimal if and only if all the states are distinguishable.

Steps for DFA Minimization using Equivalence Theorem or Set Partitioning method

Step 1 − All the states **Q** are divided in two partitions − **final states** and **non-final states** and are denoted by $\mathbf{P_0}$. All the states in a partition are 0^{th} equivalent. Take a counter **k** and initialize it with 0.

Step 2 − Increment k by 1. For each partition in P_k, divide the states in P_k into two partitions if they are k-distinguishable.

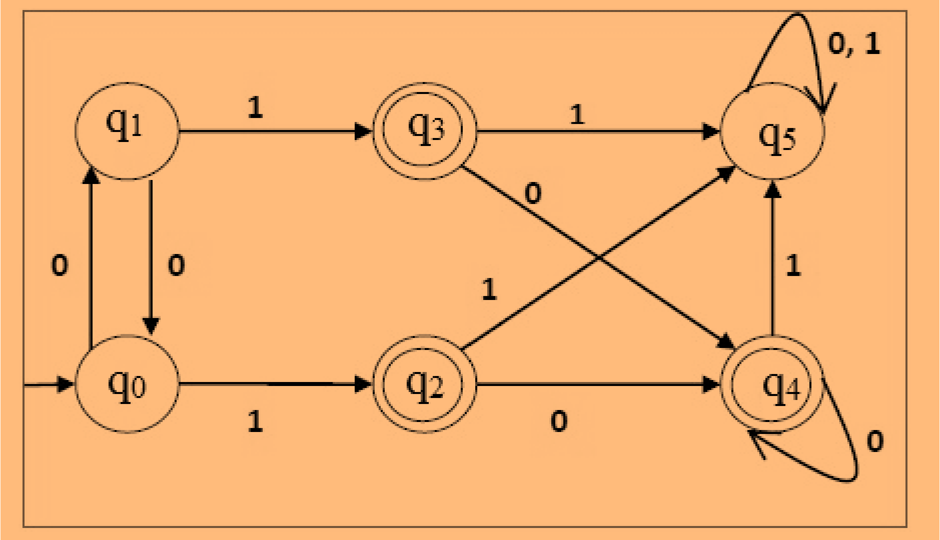
Two states within this partition q_i and q_j are k-distinguishable if there is an input **w** such that $\boldsymbol{\delta(q_i, w)}$ and $\boldsymbol{\delta(q_j, w)}$ are (k-1) distinguishable.

Step 3 − If $P_k \neq P_{k-1}$, repeat Step 2, otherwise go to Step 4.

Step 4 − Combine k^{th} equivalent sets and make them the new states of the reduced DFA.

Note: merge dead states and remove unreachable states before following these steps.

Example: Minimize the DFA using set partitioning method.



Solution

States (q_i)	$\delta(q_i, 0)$	$\delta(q_i, 1)$
→q_0	q1	q2
q_1	q0	q3
*q_2	q4	q5
*q_3	q4	q5
*q_4	q4	q5
q_5	q5	q5

Solution (cont..)

States (q)	$\delta(q_i, 0)$	$\delta(q_i, 1)$
$\rightarrow q_0$	q_1	q_2
q_1	q_0	q_3
$*q_2$	q_4	q_5
$*q_3$	q_4	q_5
$*q_4$	q_4	q_5
q_5	q_5	q_5

Now, apply the equivalence theorem to the DFA:

$P_0 = \{(q2,q3,q4), (q0, q1, q5)\}$

$P_1 = \{(q2,q3,q4),$

Check 1-equivalence of (q2, q3)
$\delta(q2,0)=q4$ and $\delta(q3,0)=q4$
$\delta(q2,1)=q5$ and $\delta(q3,1)=q5$
Check 1-equivalence of (q2, q4)
$\delta(q2,0)=q4$ and $\delta(q4,0)=q4$
$\delta(q2,1)=q5$ and $\delta(q4,1)=q5$
Check 1-equivalence of (q3, q4)
$\delta(q3,0)=q4$ and $\delta(q4,0)=q4$
$\delta(q3,1)=q5$ and $\delta(q4,1)=q5$

Solution (cont..)

States (q)	$\delta(q_i, 0)$	$\delta(q_i, 1)$
$\rightarrow q_0$	q1	q2
q_1	q0	q3
$*q_2$	q4	q5
$*q_3$	q4	q5
$*q_4$	q4	q5
q_5	q5	q5

Now, apply the equivalence theorem to the DFA:

$P_0 = \{(q2,q3,q4), (q0, q1, q5)\}$

$P_1 = \{(q2,q3,q4), (q0, q1),(q5)\}$

Check 1-equivalence of (q0, q1)

$\delta(q0,0)=q1$ and $\delta(q1,0)=q0$

$\delta(q0,1)=q2$ and $\delta(q1,1)=q3$

Check 1-equivalence of (q0, q5)

$\delta(q0,0)=q1$ and $\delta(q5,0)=q5$

$\delta(q0,1)=q2$ and $\delta(q5,1)=q5$

Solution (cont..)

States (q)	$\delta(q_i, 0)$	$\delta(q_i, 1)$
$\rightarrow q_0$	q1	q2
q_1	q0	q3
$*q_2$	q4	q5
$*q_3$	q4	q5
$*q_4$	q4	q5
q_5	q5	q5

Now, apply the equivalence theorem to the DFA:

P_0 = {(q2,q3,q4), (q0, q1, q5)}

P_1 = {(q2,q3,q4), (q0, q1),(q5)}

Check 1-equivalence of (q0, q1)
δ(q0,0)=q1 and δ(q1,0)=q0
δ(q0,1)=q2 and δ(q1,1)=q3

Check 1-equivalence of (q0, q5)
δ(q0,0)=q1 and δ(q5,0)=q5
δ(q0,1)=q2 and δ(q5,1)=q5

Solution (cont..)

States (q)	$\delta(q_i, 0)$	$\delta(q_i, 1)$
→ q_0	q1	q2
q_1	q0	q3
$*q_2$	q4	q5
$*q_3$	q4	q5
$*q_4$	q4	q5
q_5	q5	q5

Now, apply the equivalence theorem to the DFA:

P_0 = {(q2,q3,q4), (q0, q1, q5)}

P_1 = {(q2,q3,q4), (q0, q1),(q5)}

Check 1-equivalence of (q1, q5)
δ(q1,0)=q0 and δ(q5,0)=q5
δ(q1,1)=q3 and δ(q5,1)=q5

Solution (cont..)

States (q)	$\delta(q_i, 0)$	$\delta(q_i, 1)$
$\rightarrow q_0$	q_1	q_2
q_1	q_0	q_3
$*q_2$	q_4	q_5
$*q_3$	q_4	q_5
$*q_4$	q_4	q_5
q_5	q_5	q_5

Now, apply the equivalence theorem to the DFA:

P_0 = {(q2,q3,q4), (q0, q1, q5)}

P_1 = {(q2,q3,q4), (q0, q1),(q5)}

P_2 = {(q2,q3,q4)

Hence, $P_1 = P_2$.

Check 2-equivalence of (q2, q3)
δ(q2,0)=q4 and δ(q3,0)=q4
δ(q2,1)=q5 and δ(q3,1)=q5
Check 2-equivalence of (q2, q4)
δ(q2,0)=q4 and δ(q4,0)=q4
δ(q2,1)=q5 and δ(q4,1)=q5
Check 2-equivalence of (q3, q4)
δ(q3,0)=q4 and δ(q4,0)=q4
δ(q3,1)=q5 and δ(q4,1)=q5

Solution (cont..)

States (q)	$\delta(q_i, 0)$	$\delta(q_i, 1)$
$\rightarrow q_0$	q1	q2
q_1	q0	q3
$*q_2$	q4	q5
$*q_3$	q4	q5
$*q_4$	q4	q5
q_5	q5	q5

Now, apply the equivalence theorem to the DFA:

P_0 = {(q2,q3,q4), (q0, q1, q5)}

P_1 = {(q2,q3,q4), (q0, q1),(q5)}

P_2 = {(q2,q3,q4), (q0, q1),(q5)}

Hence, $P_1 = P_2$.

Check 2-equivalence of (q0, q1)
δ(q0,0)=q1 and δ(q1,0)=q0
δ(q0,1)=q2 and δ(q1,1)=q3

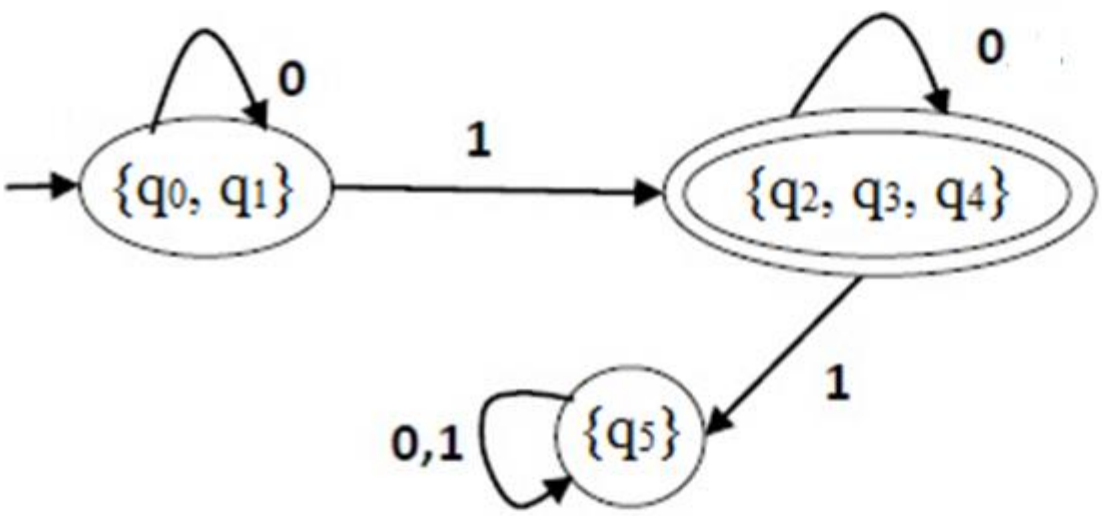
Transition table of Minimized DFA

States (q)	δ(q_i, 0)	δ(q_i, 1)
→ q_0	q1	q2
q_1	q0	q3
*q_2	q4	q5
*q_3	q4	q5
*q_4	q4	q5
q_5	q5	q5

States (q)	δ(q_i, 0)	δ(q_i, 1)
→ {q_0, q_1}	{q_0, q_1}	*{q_2 , q_3, q_4}
*{q_2 , q_3, q_4}	*{q_2 , q_3, q_4}	q_5
q_5	q_5	q_5

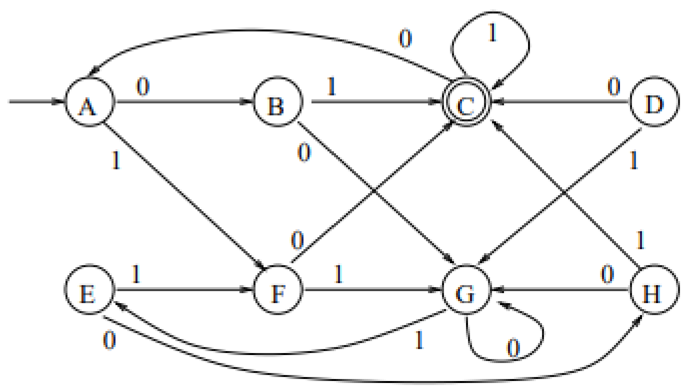
Transition Diagram of Minimized DFA

States (q)	δ(q_i, 0)	δ(q_i, 1)
→ {q_0, q_1}	{q_0, q_1}	*{q_2 , q_3, q_4}
*{q_2 , q_3, q_4}	*{q_2 , q_3, q_4}	q_5
q_5	q_5	q_5



Practice problem

1. Minimize the DFA using set partitioning method



2. Minimize the DFA using set partitioning method whose transition table is:

State	Input	
	a	*b*
$\rightarrow q_0$	q_0	q_3
q_1	q_2	q_5
q_2	q_3	q_4
q_3	q_0	q_5
q_4	q_0	q_6
q_5	q_1	q_4
q_6	q_1	q_3

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