Minimization of DFA

↓ reducing the no. of states.

- The language generated by a DFA is unique.
- But, ~~there~~ may exist ~~the~~ many DFA's that accept the same language

⇓ DFA's are said to be equivalent.

- For computations or other applications, like string matching, we desire DFA's with ~~lesser~~ fewer no. of states.
- For storage efficiency also, it is desirable to reduce the no. of states, hence minimize the dfa.
- Possible by finding the two diff types of states :→ distinguishable and indistinguishable.

define

For understanding the technique, you may check video by mahesh babu "Minimization of DFA"

Def :→ Two states p and q of a DFA are equivalent (indistinguishable) iff $\delta(p, w)$ and $\delta(q, w)$ are both final states or non-final states for all $w \in \Sigma^*$.

(a) if $\delta(p, w) \in F$ and $\delta(q, w) \in F$ then p and p are equivalent or indistinguishable.

(b) if $\delta(p, w) \notin F$ and $\delta(q, w) \notin F$ then also p and q are equivalent.

(c) If there exists at least one string w such that one of the state $\delta(p, w)$ and $\delta(q, w)$ is in final states, and other is in non-final states, then p and q are not equivalent and are called distinguishable.

Note :→ Two states are either distiaguishable or indistinguishable

Q)

Present state	Next state a	Next state b
→ q0	q1	q0
q1	q0	q2
q2	q3	q1
* q3	q3	q0
q4	q3	q5
q5	q6	q4
q6	q5	q6
q7	q6	q3.

(A) 8 - states
q0 - q7

(A) q0 - IS
q3 → FS.

(1) Divide the states into two groups comprising of Final and non-final states

0 - equivalence

{q0, q1, q2, q4, q5, q6, q7} {q3}

(2) Partition the group of NF states (give input) repeatedly till no more partitioning is possible by separating into equivalence classes: by checking them in previous equivalence.

1 - equivalence

{q0, q1, q5, q6, q7} {q2, q4} {q3}

2 equivalence

{q0, q6, q7} {q1, q5} {q2, q4} {q3}

3- equivalence

$\underset{}{\overset{G_5}{\{q_0, q_6\}}}$ $\overset{G_4}{\{q_7\}}$ $\overset{G_3}{\{q_1, q_5\}}$ $\overset{G_2}{\{q_2, q_4\}}$ $\overset{G_1}{\{q_3\}}$

no further equivalence possible, as 4- equivalence yields the same result.

Transfer the groups onto a TT

Present state	Next state	
	a	b
→ q_0	q_1	q_0
$[q_0, q_6]$	$[q_1, q_5]$	$[q_0, q_6]$
$[q_1, q_5]$	$[q_0, q_6]$	$[q_2, q_4]$
$[q_2, q_4]$	q_3	$[q_1, q_5]$
q_3	q_3	$[q_0, q_6]$
q_7	$[q_0, q_6]$	q_3

only 5 - states exist now

Present state	Next state a	Next state b
→ A	B	G
B	C	D
C	C	D
D	E	F
* E	C	D
F	E	F
G	E	F

0-equivalence

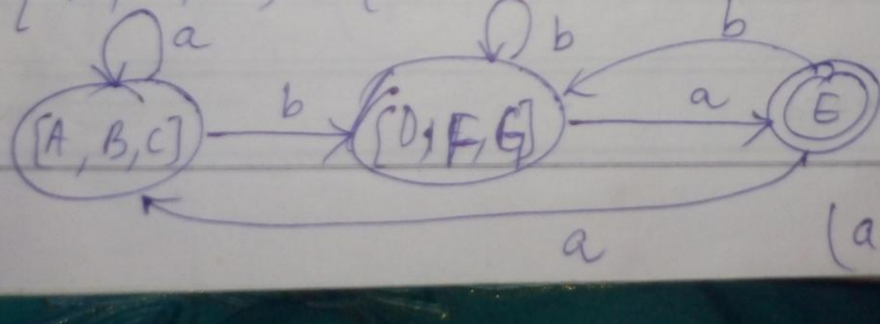
{ A, B, C, D, F, G }   {E}

1- equivalence

{A, B, C}   {D, F, G} {E}

2 - equivalence

G_3 { A, B, C } G_2 { D, F, G } G_1 { E }



3-state

$(a+b)^* ba$

	Present state	Next State a	Next State b
→	q0	q1	q3
	q1	q2	q4
	q2	q1	q4
	q3	q2	q4
*	q4	q4	q4

0 - equivalence

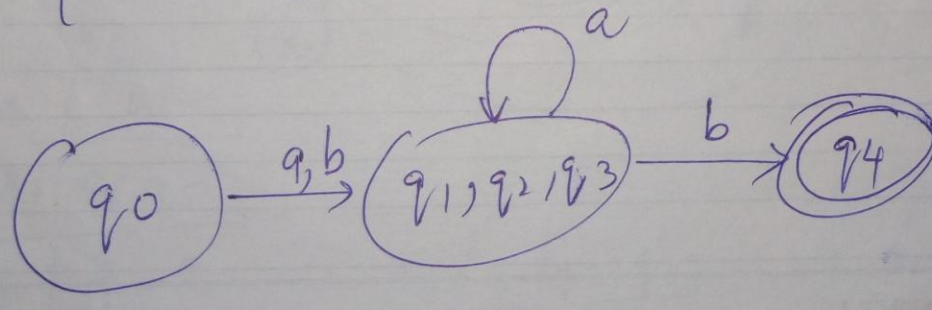
{q0, q1, q2, q3} {q4}

1- equivalence

{q0} {q1, q2, q3} ~~{q2, q3}~~ {q4}

2- equivalance

{q0} {q1, q2, q3} {q4}



(7)

	a	b
→q0	q1	q5
q1	q6	q2
*q2	q0	q2
q3	q2	q6
q4	q7	q5
q5	q2	q6
q6	q6	q5
q7	q6	q2

0-equivalence

{q0, q1, q3, q4, q5, q6, q7} {q2}

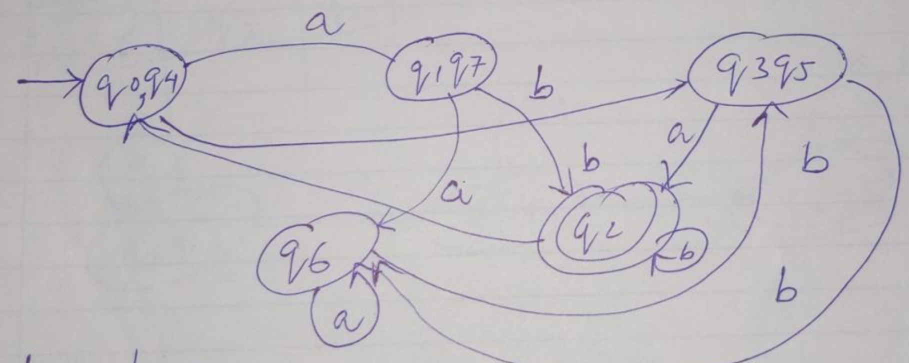
1-equivalence

{q0, q4, q6} {q1, q3, q5, q7} {q2}

2-equivalence

{q0, q4} {q6} {q1, q7} {q3, q5} {q2}

3-equivalence ↓ same.



	a	b
→q0q4	q1,q7	q3, q5
q1q7	q6	q2 *
q3q5	q2	q6
q2	q0	q2

a

q6	a	q3, q5

	a	b
→ q_0	q_1	q_3
q_1	q_1	q_2
⊛ q_2	—	q_2
q_3	q_4	q_3
q_4	q_4	q_5
* q_5	q_2	q_5

0-equivalence

$\{q_0, q_1, q_3, q_4\}$ $\{q_2, q_5\}$

1-equivalence

$\{q_0, q_3\}$ $\{q_1, q_4\}$ $\{q_2, q_5\}$

2-equivalence

same

	a	b
q_0, q_3	q_1, q_4	q_3
q_1, q_4	q_1, q_4	q_2, q_5
q_2, q_5	q_2	q_2, q_5