

As  $\delta'$  is constructed for all inputs on B, B is marked.

C is unmarked still.

Then, construct the  $\delta'$  function for the new unmarked state C for inputs 'a' and 'b'.

$$\begin{aligned}
 \delta'(C, a) &= \varepsilon - \text{closure}(\delta(C, a)) \\
 &= \varepsilon - \text{closure}(\delta((q_1, q_3), a)) \\
 &= \varepsilon - \text{closure}(\delta(q_1, a) \cup \delta(q_3, a)) \\
 &= \varepsilon - \text{closure}(\{q_2\} \cup \{\Phi\}) \\
 &= \varepsilon - \text{closure}(q_2) = q_2 = D
 \end{aligned}$$

$$\begin{aligned}
 \delta'(C, b) &= \varepsilon - \text{closure}(\delta(C, b)) \\
 &= \varepsilon - \text{closure}(\delta((q_1, q_3), b)) \\
 &= \varepsilon - \text{closure}(\delta(q_1, b) \cup \delta(q_3, b)) \\
 &= \varepsilon - \text{closure}(q_3 \cup \{\Phi\}) \\
 &= \varepsilon - \text{closure}(q_3) = q_1, q_3 = C
 \end{aligned}$$

As  $\delta'$  is constructed for all inputs on C, C is marked.

D is unmarked still.

Then, construct the  $\delta'$  function for the new unmarked state D for inputs 'a' and 'b'.

$$\begin{aligned}
 \delta'(D, a) &= \varepsilon - \text{closure}(\delta(D, a)) \\
 &= \varepsilon - \text{closure}(\delta((q_2), a)) \\
 &= \varepsilon - \text{closure}(\delta(q_2, a)) \\
 &= \varepsilon - \text{closure}(\{\Phi\}) \\
 &= \Phi \\
 \delta'(D, b) &= \varepsilon - \text{closure}(\delta(D, b)) \\
 &= \varepsilon - \text{closure}(\delta((q_2), b)) \\
 &= \varepsilon - \text{closure}(\delta(q_2, b))
 \end{aligned}$$

$$\begin{aligned}
&= \varepsilon - \text{closure}(\{q_3\}) \\
&= \{q_1, q_3\} = C
\end{aligned}$$

States A and C are final states as the states contain  $q_3$ .

The equivalent DFA is given in Fig. 5.21

## 1 picture

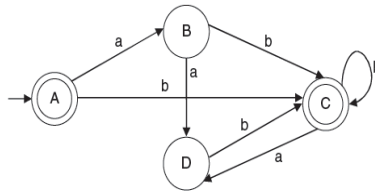


Fig. 5.21

## 5.7 Equivalence of Two Finite Automata

Two FA are said to be equivalent if they generate the same (equivalent) RE. If both the FA are minimized, they reveal the equivalence by being minimized to the same number of states and the same transitional functions. In this section, we shall discuss the process of proving the equivalence between two FA.

Let there be two FA,  $M$  and  $M'$ , where the number of input symbols are the same.

- Make a comparison table with  $n + 1$  columns, where  $n$  is the number of input symbols.
- In the first column, there will be a pair of vertices  $(q, q')$  where  $q \in M$  and  $q' \in M'$ . The first pair of vertices will be the initial states of the two machines  $M$  and  $M'$ . The second column consists of  $(q_a, q'_a)$ , where  $q_a$  is reachable from the initial state of the machine  $M$  for the first input, and  $q'_a$  is reachable from the initial state of the machine  $M'$  for the first input. The other  $n - 2$  columns consist of a pair of vertices from  $M$  and  $M'$  for  $n - 1$  inputs, where  $n = 2, 3, \dots, n - 1$ .
- If any new pair of states appear in any of the  $n - 1$  next state columns, which were not taken in the first column, take that pair in the present state column and construct subsequent column elements like the first row.
- If a pair of states  $(q, q')$  appear in any of the  $n$  columns for a pair of states in the present state column, where  $q$  is the final state of  $M$  and  $q'$  is the non-final state of  $M'$  or vice versa, terminate the construction and conclude that  $M$  and  $M'$  are not equivalent.
- If no new pair of states appear, which were not taken in the first column, stop the construction and declare that  $M$  and  $M'$  are equivalent.

The following examples describe this in detail.

## 2 picture

**Example 5.18** Find whether the two DFAs given in Fig. 5.22 are equivalent or not.

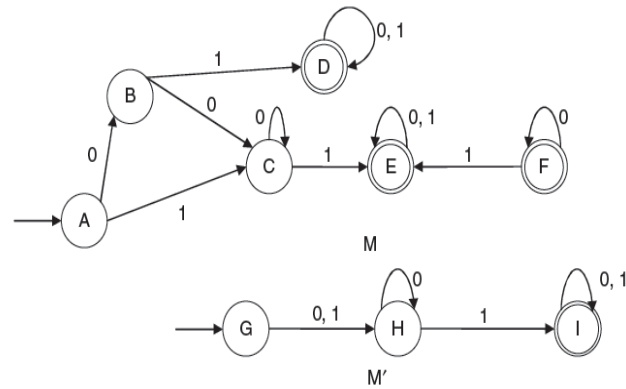


Fig. 5.22

**Solution:** In the previous two DFAs, the number of input is two 0 and 1. Therefore, the comparison table is of three columns. The beginning state pair is (A, G). From there, for input 0, we are getting (B, H) and for input 1 we are getting (C, H). Both of them are new combination of states. Among them, first take (B, H) in the present state column. The next state pairs for input 0 and 1 are constructed. By this process, the comparison continues.

After the fifth step, further construction is stopped as no new state combinations have appeared. In the whole table, no such type of combination of states appear where one state is the final state of one machine and the other is the non-final state of another machine.

Therefore, the two DFAs are equivalent.

<i>PresentState</i> ( $q, q'$ )	<i>NextState</i>	
	<i>ForI/P0</i> ( $q_0, q'_0$ )	<i>ForI/P1</i> ( $q_1, q'_1$ )
(A, G)	(B, H)	(C, H)
(B, H)	(C, H)	(D, I)
(C, H)	(C, H)	(E, I)
(E, I)	(E, I)	(E, I)
(D, I)	(D, I)	(D, I)

### 3 picture

**Example 5.19** Find whether the two DFAs given in Fig. 5.23 are equivalent or not.

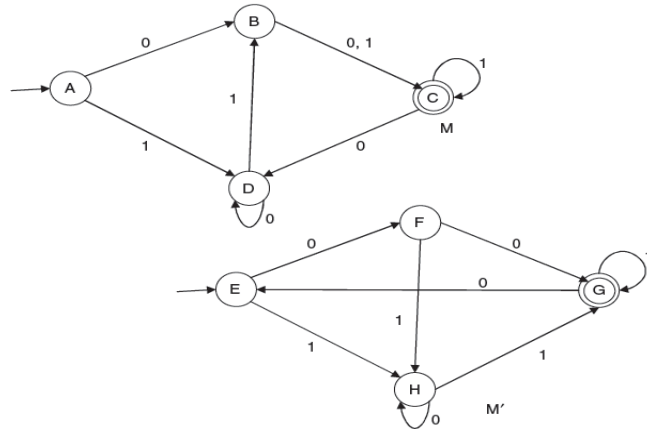


Fig. 5.23

**Solution:** In the previous two DFAs, the number of input is two 0 and 1. Therefore, the comparison table is of three columns. The beginning state pair is (A, E). From there, for input 0, we get (B, F) and for input 1 we get (D, H). Both of them are new combination of states. Among them, take (B, F) in the present state column. The next state pairs for input 0 is (C, G) and for input 1 is (C, H). (C, , H) is a combination of states where C is the final state of M and H is the non-final state of  $M'$ . Further construction stops here declaring that the two DFAs are not equivalent.

<i>PresentState</i> ( $q, q'$ )	<i>NextState</i>	
	<i>ForI/P0</i> ( $q_0, q'_0$ )	<i>ForI/P1</i> ( $q_1, q'_1$ )
(A, E)	(B, F)	(D, H)
(B, F)	(C, G)	(C, H)

## 5.8 Equivalence of Two Regular Expressions

For every RE, there is an accepting FA. If the FA constructed from both of the REs are the same, then we can say that two REs are equivalent.

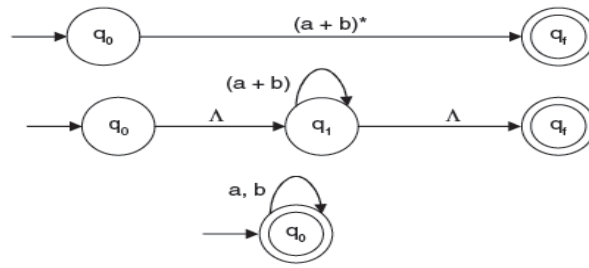
## 4 picture

**Example 5.20** Prove that the following REs are equivalent.

$$L_1 = (a + b)^* L_2 = a * (b * a)^*$$

**Solution:** The FA for  $L_1$  is constructed in Fig. 5.24.

## 5 picture



**Fig. 5.24**

The FA for  $L_2$  is constructed in Fig. 5.25.

## 6 picture

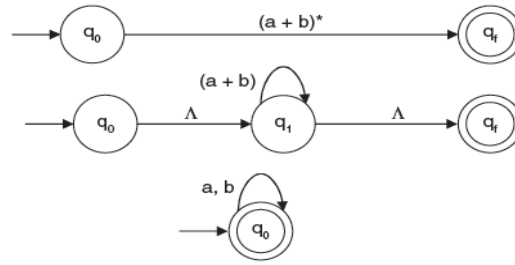


Fig. 5.24