3.9.2 Conversion of an NFA with ε Moves to DFA without ε Move

Let us assume that in the NFA we want to remove the ε move which exists between the states S_1 and S_2.

This can be removed in the following way

3.9.2 Conversion of an NFA with ε Moves to DFA without ε Move

Let us assume that in the NFA we want to remove the ε move which exists between the states S_1 and S_2.

This can be removed in the following way

- Find all the edges (transitions) those start from the state S_2.
- Duplicate all these transitions starting from the state S_1, keeping the edge label the same.
- If S_1 is the initial state, also make S_2 as the initial state.
- If S_2 is the final state, also make S_1 as the final state.

Consider the following examples.

Example 3.8 Convert the following NFA with null move in Fig. 3.13 to an equivalent DFA without ε move.

Example 3.8 Convert the following NFA with null move in Fig. 3.13 to an equivalent DFA without ε move.

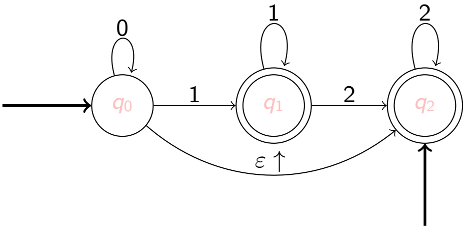

Fig. 3.13

Solution: Three null moves exist in the previous transitional diagram. The transitions are

1. from q_0 to q_1
2. from q_1 to q_2
3. from q_0 to q_2

These null transitions are removed step by step.

Step I:

1. Between the states q_0 and q1, there is a null move. If we want to remove that null transition, we have to find all the edges starting from q_1.The edges are q_1 to q_1 for input 1 and q_1 to q_2 for input ε.
2. All these transitions starting from the state q_0 are duplicated, keeping the edge label the same. q_0 is the initial state, so make q_1 also an initial state. The modified transitional diagram is given in Fig. 3.14.

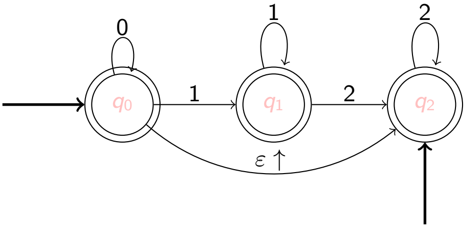

Fig. 3.14

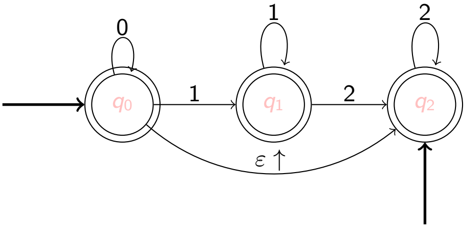

Fig. 3.14

Step II:

Between q_1 and q_2, there is a null transaction. To remove that null transition, find all the edges starting from q_2. The edge is q_2 to q_2 for input 2.

62 | Introduction to Automata Theory, Formal Languages and Computation

2. Duplicate the transition starting from the state q_1, keeping the edge label the same. q_1 is the initial state, so make q_2 also an initial state. q_2 is the final state, so make q_1 also a final state. The modified transitional diagram is given in Fig. 3.15.

62 | Introduction to Automata Theory, Formal Languages and Computation

2. Duplicate the transition starting from the state q_1, keeping the edge label the same. q_1 is the initial state, so make q_2 also an initial state. q_2 is the final state, so make q_1 also a final state. The modified transitional diagram is given in Fig. 3.15.

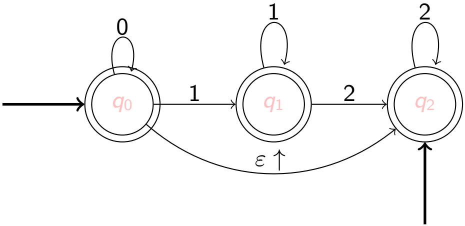

Fig. 3.15

Step III:

1. Between q_0 and q_2, there is a null transaction. To remove that null transition, find all the edges starting from q_2. The edge is q_2 to q_2 for input 2.
2. Duplicate the transition starting from the state q_0, keeping the edge label the same. q_0 is the initial state, so make q_2 also an initial state. q_2 is the final state, so make q_0 also a final state. The modified transitional diagram is given in Fig. 3.16.

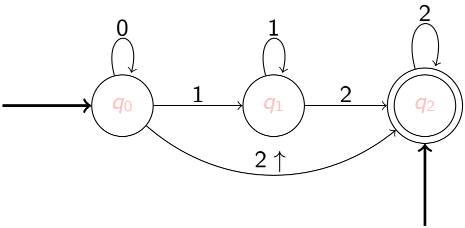

Fig. 3.16

This is the equivalent NFA without ε move obtained from the NFA with null transaction.

Example 3.9 Convert the following NFA with null move to an equivalent NFA without ε move as given in Fig. 3.17.

Example 3.9 Convert the following NFA with null move to an equivalent NFA without ε move as given in Fig. 3.17.

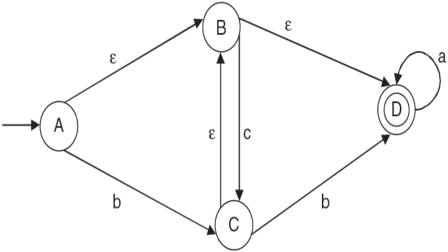

Fig. 3.17

Example 3.9 Convert the following NFA with null move to an equivalent NFA without ε move as given in Fig. 3.17.

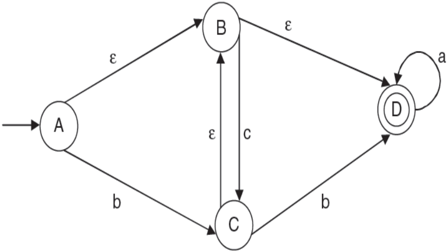

Fig. 3.17

Solution: In the given NFA with ε move, there are three ε transitions: from A to B, from B to D, and from B to C. The three ε transitions are removed step by step.

Step I: There is a ε transition from A to B. To remove that ε transition, we have to find all the edges starting from B. The edges are B to C for input c and B to D for input ε.

Duplicate the transition starting from the state A, keeping the edge label the same. A is the initial state, so make B also an initial state. The modified transaction diagram is shown in Fig. 3.18.

Step I: There is a ε transition from A to B. To remove that ε transition, we have to find all the edges starting from B. The edges are B to C for input c and B to D for input ε.

Duplicate the transition starting from the state A, keeping the edge label the same. A is the initial state, so make B also an initial state. The modified transaction diagram is shown in Fig. 3.18.

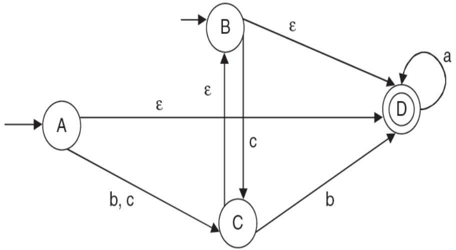

Fig. 3.18

Step II: Again there are three ε transactions in the NFA. These are from A to D, from B to D, and from C to B. Let us remove the ε transaction from B to D. The edge starting from D is D to D for input a. Duplicate the transition starting from the state B, keeping the edge label the same. As B is the initial state, D will also be the initial state. As D is the final state, B will also be the final state. The modified transaction diagram will be as shown in Fig. 3.19.

Step II: Again there are three ε transactions in the NFA. These are from A to D, from B to D, and from C to B. Let us remove the ε transaction from B to D. The edge starting from D is D to D for input a. Duplicate the transition starting from the state B, keeping the edge label the same. As B is the initial state, D will also be the initial state. As D is the final state, B will also be the final state. The modified transaction diagram will be as shown in Fig. 3.19.

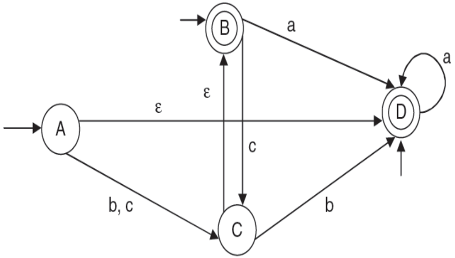

Fig. 3.19

Step III: Now we are going to remove the ε transaction from A to D. The edge starting from D is D to D for input a. Duplicate the transition starting from the state A, keeping the edge label the same. As A is the initial state, D will also be the initial state. As D is the final state, A will also be the final state. The modified transaction diagram will be as shown in Fig. 3.20.

Step III: Now we are going to remove the ε transaction from A to D. The edge starting from D is D to D for input a. Duplicate the transition starting from the state A, keeping the edge label the same. As A is the initial state, D will also be the initial state. As D is the final state, A will also be the final state. The modified transaction diagram will be as shown in Fig. 3.20.

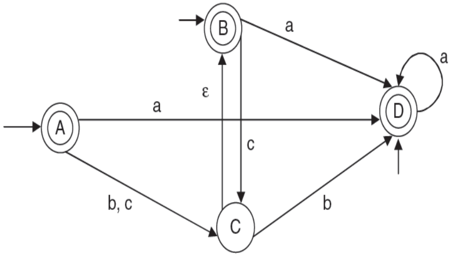

Fig. 3.20

64 | Introduction to Automata Theory, Formal Languages and Computation

Step IV: There is only one ε transaction from C to B. To remove this, find the edges starting from B.
There are two: from B to D for input a and from B to C for input c. Start these edges from C. B is the final state, so make C as the final state. The modified transitional diagram will be as shown in Fig. 3.21.

Step IV: There is only one ε transaction from C to B. To remove this, find the edges starting from B.
There are two: from B to D for input a and from B to C for input c. Start these edges from C. B is the final state, so make C as the final state. The modified transitional diagram will be as shown in Fig. 3.21.

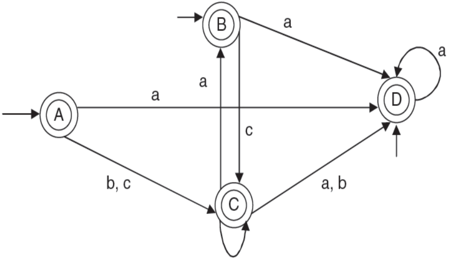

Fig. 3.21

3.10 Equivalence of DFA and NFA

A DFA can simulate the behaviour of an NFA by increasing its number of states. The DFA does not contain any null move and any transitional functions of a state and input reaching to more than one states. In this section, we shall prove the following: 'If a language L is accepted by an NDFA, then there exists a DFA that accepts L'.

Proof : Let M be an NDFA denoted by $\{Q, \Sigma, \delta, q_0, F\}$ which accepts L.
We construct a DFA M' $=\{Q', \Sigma, \delta', q_0', F'\}$,where Q' contains the subsets of $Q, i.e., Q' = 2^Q$.

Q' may be denoted by $[q_1, q_2, ...q_n]$ as a single state where $q_1, q_2, ...q_n \in Q$

Initial state $q_0' = [q_0]$

Final state $F' =$ set of all states in Q' containing at least one final state of M.

Transitional function δ' is defined as

$$\delta'([q_1, q_2, ...q_n], a) = [P_1, P_2, ...P_k] \text{ where } q_1, q_2, ...q_n \in Q, \text{ a} \in \Sigma \text{ and}$$
$$\delta(\{q_1, q_2, ...q_n\}, a) = \delta(q_1, a) \cup \delta(q_2, a) \cup ..... \cup \delta(q_n, a)$$
$$= P_1 \cup P_2 \cup ..... \cup P_k$$

This is the case for a single input 'a'. Now, we shall prove it for some input string x. x may be of length 0 to length n. We prove this by induction.

Let $|x| = 0, i.e., x$ is a null string.

$$\delta'(q_0', x) = [q_0] as q_0' = [q_0] \text{ and } \delta(q_0, x) = \delta(q_0, \varepsilon) = q_0.$$

Thus, the induction has a base condition.

Let us assume that it is true for each string of length n. Now we need to prove that the result is true

for any string of length $(n+1)$.

Let $\quad S = xa.$

So, $\quad |S| = |xa| = |x| + |a| = n + 1.$

$$\delta'(q_0', S) = \delta'(q_0', xa) = \delta'(\delta'(q_0', x)a).$$