2.11. Transformation of NFA to DFA

For every non deterministic finite automata there exist an equivalent deterministic finite automata. The equivalence is defined in terms of language acceptance. Since a NFA is a nothing but a finite automata in which zero, one or more transition on an input symbol is permitted, we can always construct a finite automata which will simulate all the moves of NFA on a particular input symbol in parallel, then get a finite automata in which there will be exactly one transition on every input symbol, hence it will be DFA equivalent to NFA.

TIPS

Since the DFA equivalent of NFA is to simulate the moves in parallel state of a DFA will be combination of one or more states of NFA, hence every state of a DFA will be represented by some subset of set of states of NFA, and therefor the transition of NFA to DFA is normally called a subset construction.

So transformation of NFA to DFA in value, finding subset of set of states of NFA. The conversion of an NFA to DFA involves constructing all subsets of the states of NFA. The subset constructing starts from an NFA $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$. Its goal is the description of DFA $M_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ such that $L(M_1) = L(M_2)$.

The input alphabets of two automata are the same, and start state of M_2 is the set containing only the start state of M_1. The other component of M_2 are constructed as follows :

(i) Q_2 is the set of subsets of Q_1, *i.e.* Q_2 is the power set of Q_1. If Q_1 has n states then Q_2 will have 2^n states. Often not all these states are accessible from the start state of Q_2. In accessible states can be thrown away, so effectively the number of states of M_2 may be much smaller then 2^n.

(ii) F_2 is all sets of M_1's states that include at least one accepting state of F_1.

(iii) For each subset S of Q_1 and for each input symbol a in Σ,

$$\delta_2(S, a) = \cup_{P \text{ in } S}\ \delta_2(P, a)$$

$$S = \{q_1, q_2, ..., q_i\} \in Q_1$$

That is to compute $\delta_2(S, a)$, we look at all the states in P in S, see what states M_1 goes from P on input a and take the union of all those states.

When the NFA M_1 has n states the corresponding DFA M_2 has 2^n states. However we need not construct M_2 for all these 2^n states, but only for those states reachable from starting state. So we start the construction of M_2 from q_0. We continue by considering only states appearing earlier under input columns and constructing M_2 for such states. We halt when no more states appear under the input columns.

Let us implement this concept in the form of set of steps as algorithm through an example.

Example 2.17. Convert the following NFA into DFA.

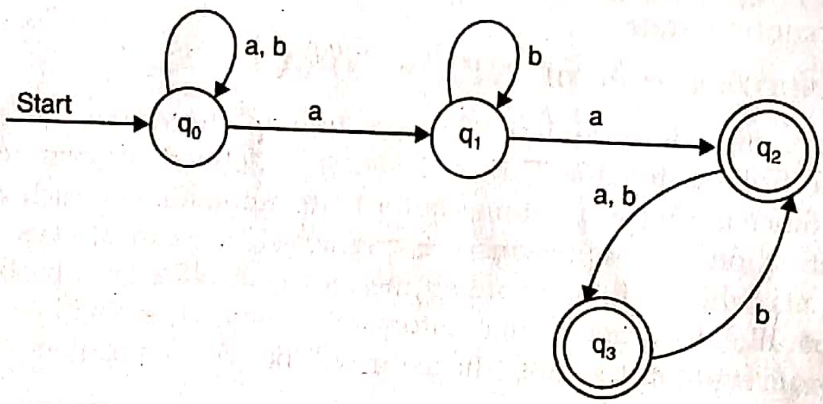

Fig. 2.40.

Solution.

Step 1. Seek all the transition from starting state q_0 for every symbol in Σ *i.e.* (a, b). If you get a set of states for same input then consider that set as new single state as :

$$\delta(q_0, a) = \{q_0, q_1\}$$

in Fig. 2.40 q_0 transit to both q_0 and q_1 for input a.

$$\delta(q_0, b) = \{q_0\}$$

Step 2. In step 1 we are getting a new state $\{q_0, q_1\}$. Now repeat step 1 for this new state only *i.e.* check all transitions of a and b (that is Σ) from $\{q_0, q_1\}$ as :

$$\delta(\{q_0, q_1\}, a) = \delta(q_0, a) \cup \delta(q_1, a) = \{q_0, q_1\} \cup \{q_2\}$$
$$= \{q_0, q_1, q_2\}$$
$$\delta(\{q_0, q_1\}, b) = \delta(q_0, b) \cup \delta(q_1, b) = \{q_0, q_1\}$$

Step 3. Repeat step 2 till you are getting any new state. All those states which consists at least one accepting state of given NFA as member state will be considered as final states.

$[q_0, q_1, q_2]$ is new state.

$$\delta(\{q_0, q_1, q_2\}, a) = \delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_2, a)$$
$$= \{q_0, q_1\} \cup \{q_2\} \cup \{q_3\}$$
$$= \{q_0, q_1, q_2, q_3\} \quad \text{(New state)}$$
$$\delta(\{q_0, q_1, q_2\}, b) = \delta(q_0, b) \cup (q_1, b)\ \delta(q_2, b)$$
$$= q_0 \cup q_1 \cup q_3 = \{q_0, q_1, q_3\} \quad \text{(New state)}$$
$$\delta(\{q_0, q_1, q_2, q_3\}, a) = \{q_0, q_1, q_2, q_3\} \quad \text{(Old state)}$$
$$\delta(\{q_0, q_1, q_2, q_3\}, b) = \{q_0, q_1, q_3, q_2\} \quad \text{(Old state)}$$
$$\delta(\{q_0, q_1, q_3\}, a) = \{q_0, q_1, q_2, q_3\} \quad \text{(Old state)}$$
$$\delta(\{q_0, q_1, q_3\}, b) = \{q_0, q_1, q_3, q_2\} \quad \text{(Old state)}$$

Let draw transition table

δ/Σ	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
$\{q_0, q_1\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1\}$
$*\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_3\}$
$*\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$
$*\{q_0, q_1, q_3\}$	$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$

Let us say

$$q_0 \rightarrow A$$
$$\{q_0, q_1\} \rightarrow B$$
$$\{q_0, q_1, q_2\} \rightarrow C$$
$$\{q_0, q_1, q_2, q_3\} \rightarrow D$$
$$\{q_0, q_1, q_3\} \rightarrow E$$

A is initial state and C, D and E are final states since they contain q_2 and q_3 as member which are final states of NFA.

So

δ/Σ	a	b
$\rightarrow A$	B	A
B	C	B
$*C$	D	E
$*D$	D	D
$*E$	D	D

Let us draw transition diagram

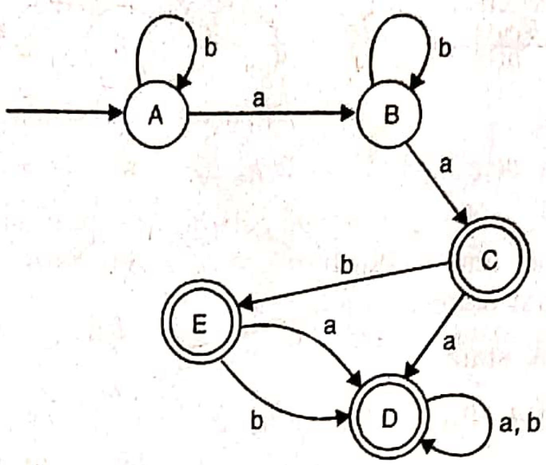

Fig. 2.41. DFA equivalent of given NFA.

It is a DFA.

Example 2.18. Given NFA is

δ/Σ	a	b
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_2\}$
q_1	$\{q_0\}$	$\{q_1\}$
$*q_2$	ϕ	$\{q_0, q_1\}$

Convert it into DFA.

Solution. Do the same process as in example 2.16.

$$\delta(q_0, a) = \{q_0, q_1\} \quad \text{(New)}$$
$$\delta(q_0, b) = \{q_2\} \quad \text{(New)}$$
$$\delta(\{q_0, q_1\}, a) = \delta(q_0, a) \cup \delta(q_1, a) = \{q_0, q_1\} \cup \{q_0\}$$
$$= \{q_0, q_1\} \quad \text{(Old)}$$
$$\delta(\{q_0, q_1\}, b) = \delta(q_0, b) \cup \delta(q_1, b) = \{q_1\} \cup \{q_1\}$$
$$= \{q_1, q_2\} \quad \text{(New)}$$
$$\delta\{q_2, a) = \{\phi\} \quad \text{(Empty set)}$$
$$\delta(q_2, b) = \{q_0, q_1\} \quad \text{(Old)}$$
$$\delta(\{q_1, q_2\}, a) = \delta(q_1, a) \cup \delta(q_2, a) = \{q_0\} \cup \{\phi\}$$
$$= \{q_0\} \quad \text{(Old)}$$
$$\delta(\{q_1, q_2\}, b) = \delta(q_1, b) \cup \delta(q_2, b) = \{q_1\} \cup \{q_0, q_1\}$$
$$= \{q_0, q_1\} \quad \text{(Old)}$$

Let say

$$q_0 \rightarrow A \quad \text{(Initial state)}$$
$$q_2 \rightarrow B \quad \text{(Final state)}$$
$$\{q_0, q_1\} \rightarrow C$$
$$\{q_1, q_2\} \rightarrow D \quad \text{(Final state)}$$

Transform table for DFA

δ/Σ	a	b
$\rightarrow A$	C	B
$*B$	$\{\phi\}$	C
C	C	D
$*D$	A	C

Transition diagram for DFA

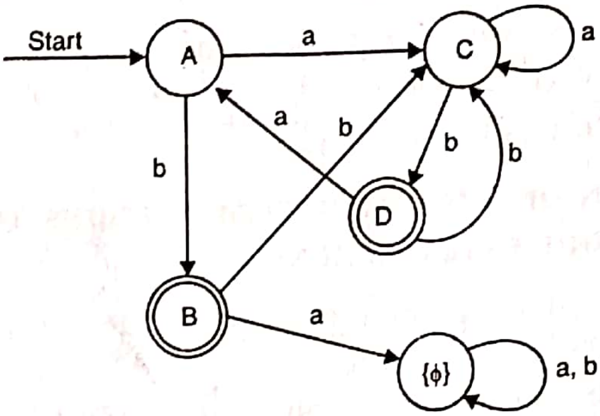

Fig. 2.42.

Here {ϕ} state is nothing but interpreted as if there is a input 'a' on state B then there is no path for it and machine will be hang.

2.12. NFA WITH ∈–TRANSITIONS

Definition If a FA is modified to permit transition without input symbols, along with zero, one or more transition on input symbols, then we get a NFA with ∈-transitions, because the transition made without symbols are called as ∈-transitions.

Consider the NFA given by following diagram.

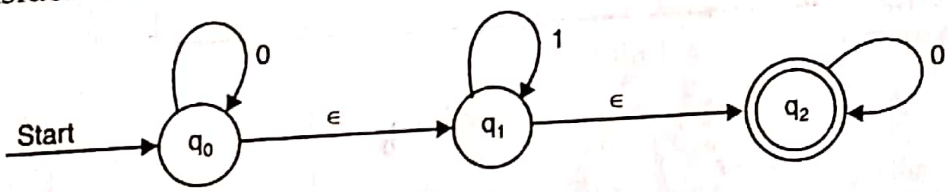

Fig. 2.43.

This is a NFA with ∈-transitions because it is possible to make transition from state q_0 to q_1 without consuming any of the input symbols. Being a finite Automata, NFA with ∈-transitions will also be denoted as five tuple.

$$M = (Q, \Sigma, \delta, q_0, F)$$

where Q, Σ, q_0 and F are having usual meaning, and transition function δ defines a mapping from

$Q \times (\Sigma \cup \in)$ to 2^Q.

2.13. ACCEPTANCE OF A STRING BY NFA WITH ∈-TRANSITIONS

A string w in Σ^* will be accepted by NFA with ∈-transition, if there exist at least one path corresponding w, which starts in an initial state, and ends in one of the final states. Since this path may formed by ∈-transition as well non ∈-transitions to find out wheather w is accepted or not by the NFA with ∈ moves, we require to defined a function ∈-closure (q), where q is a state of the automata.

The function ∈-closure (q) is defined as follows :

∈-closure (q) = set of all those states of the automata (NFA with ∈-transitions) which can be reached from q on a path labeled by ∈ *i.e.* without consuming any input symbol.

Consider the NFA discussed in Fig. 2.43 then

$$\in\text{-closure }(q_0) = \{q_0, q_1, q_2\}$$
$$\in\text{-closure }(q_1) = \{q_1, q_2\}$$
$$\in\text{-closure }(q_2) = \{q_2\}$$

2.14. Conversion of NFA with $\in$-Transitions to NFA without $\in$-Transitions

There is a conversion algorithm from NFA with $\in$-transition s to a NFA without $\in$-transitions !

We will learn this algorithm by solving an example.

Example 2.19. Consider the NFA with $\in$-transition $M = (Q, \Sigma, \delta, q_0, F)$

$$Q = \{q_0, q_1, q_2\}$$
$$\Sigma = \{a, b, c\} \text{ and } \in \text{ moves.}$$
$$\text{Initial state} = \{q_0\}$$
$$F = \{q_2\}$$

Transition Table

δ/Σ	a	b	c	$\in$
$\rightarrow q_0$	$\{q_0\}$	$\{\phi\}$	$\{\phi\}$	$\{q_1\}$
q_1	$\{\phi\}$	$\{q_2\}$	$\{\phi\}$	$\{q_2\}$
$* q_2$	$\{\phi\}$	$\{\phi\}$	$\{q_2\}$	$\{\phi\}$

and transition diagram is following

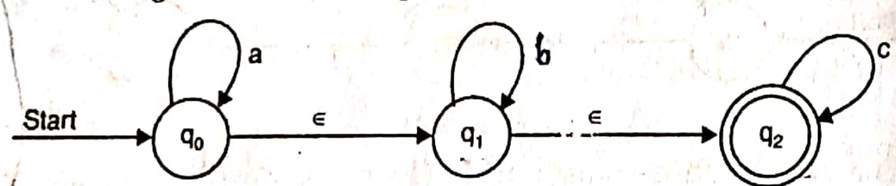

Fig. 2.44. NFA with $\in$-transition.

Solution.

Step 1. Find the states of NFA without $\in$- transition including initial state and final states as follows :

(*i*) The language accepted by the above NFA with $\in$-transform is the set of strings over $\{a, b, c\}$ including the null string and all strings with any number of a's followed by any number of b's and follows by any number of c's.

(*ii*) There will be the same number of states but the names can be constructed by writing the state name as the set of states in the $\in$-closure.

(*iii*) Initial state will be $\in$-closure of initial state of NFA with $\in$-transition in Fig. 2.44 as

$$\in\text{-closure }(q_0) = \{q_0, q_1, q_2\} \quad \text{(new initial state for NFA, without } \in \text{ transition)}$$

(*iv*) Rest of the states are

$$\in\text{-closure }(q_1) = \{q_1, q_2\} \quad \text{(New state)}$$
$$\in\text{-closure }(q_2) = \{q_2\} \quad \text{(New state)}$$

(v) The final states of NFA without ϵ-transition are all those new states which contains final state of NFA with ϵ-transition as member.

So $\{q_0, q_1, q_2\}$, $\{q_1, q_2\}$ and $\{q_2\}$ all are final states.

So if NFA without ϵ-transition is

$$M' = (Q', \Sigma, \delta', q_0', F')$$
$$Q' = (\{q_0, q_1, q_2\}, \{q_1, q_2\}, \{q_2\})$$
$$q_0' = \{q_0, q_1, q_2\}$$
$$F' = (\{q_0, q_1, q_2\}, \{q_1, q_2\}, \{q_2\})$$

Step 2. Now we have to decide δ' to find out the transitions as follows :

Very carefully consider each old machine transition in each row, you can ignore ϕ enteries and ϵ-transitions column. In the old machine $\delta(q_0, a) = q_0$ thus in new machine *i.e.* in M', $\delta'(\{q_0, q_1, q_2\}, a) = \{q_0, q_1, q_2\}$ this is just because the new machine accepts the same language as the old machine and must at least have the same transitions for the new state names. We can allow interpret it as follows :

$$\begin{aligned} \delta'(\{q_0, q_1, q_2\}, a) &= \epsilon\text{-closure } (\delta(q_0, q_1, q_2), a) \\ &= \epsilon\text{-closure } (\delta(q_0, a) \cup \delta(q_1, a) \cup \delta(q_2, a)) \\ &= \epsilon\text{-closure } (q_0 \cup \phi \cup \phi) \\ &= \epsilon\text{-closure } (q_0) = \{q_0, q_1, q_2\} \end{aligned}$$

Similarly
$$\begin{aligned} \delta'(\{q_0, q_1, q_2\}, b) &= \epsilon\text{-closure } (\delta(q_0, q_1, q_2), b) \\ &= \epsilon\text{-closure } (\delta(q_0, b) \cup \delta(q_1, b) \cup \delta(q_2, b)) \\ &= \epsilon\text{-closure } (\phi \cup q_1 \cup \phi) \\ &= \epsilon\text{-closure } (q_1) = \{q_1, q_2\}. \end{aligned}$$

$$\begin{aligned} \delta'(\{q_0, q_1, q_2\}, c) &= \epsilon\text{-closure } (\delta(q_0, q_1, q_2), c) \\ &= \epsilon\text{-closure } (\delta(q_0, c) \cup \delta(q_1, c) \cup \delta(q_1, c) \\ &= \epsilon\text{-closure } (\phi \cup \phi \cup q_2) \\ &= \epsilon\text{-closure } (q_2) = \{q_2\}. \end{aligned}$$

In the same fashion we can write

$$\begin{aligned} \delta'(\{q_1, q_2\}, a) &= \phi \\ \delta'(\{q_1, q_2\}, b) &= \{q_1, q_2\} \\ \delta'(\{q_1, q_2\}, c) &= \{q_2\} \\ \delta'(q_2, a) &= \phi \\ \delta'(q_2, b) &= \phi \\ \delta'(q_2, c) &= \{q_2\} \end{aligned}$$

= ε-closure (δ(q1, q2), a)
= ε-closure (δ(q1, a) ∪ δ(q2, a))
= ε-closure (φ ∪ φ)
= φ

So transition table for NFA without ϵ-transition will be as follows :

δ'/Σ	a	b	c
$\rightarrow * \{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_1, q_2\}$	$\{q_2\}$
$* \{q_1, q_2\}$	ϕ	$\{q_1, q_2\}$	$\{q_2\}$
$* \{q_2\}$	ϕ	ϕ	$\{q_2\}$

Let us say $\{q_0, q_1, q_2\}$ as q_x

$\{q_1, q_2\}$ as q_y

and $\{q_2\}$ as q_z

So transition diagram is as follows :

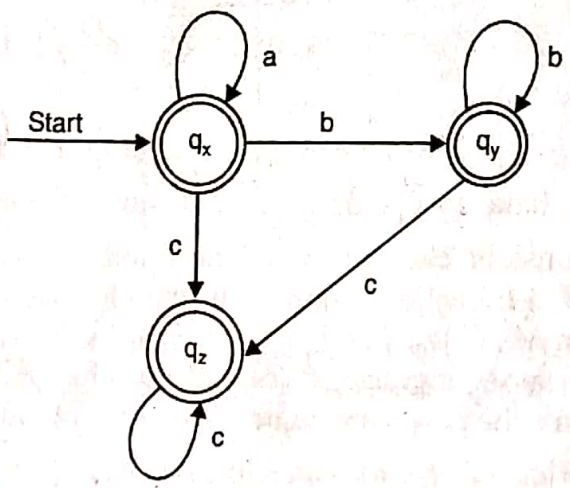

Fig. 2.45. NFA without $\in$-transitions.

Let us see one more example.

Example 2.20. Consider the NFA given by following diagram

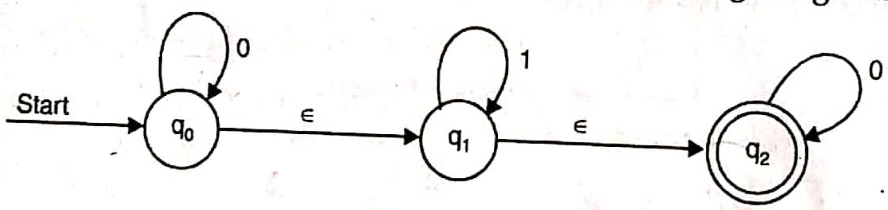

Fig. 2.46.

Find the equivalent NFA without $\in$-transitions.

Solution. Let NFA without $\in$-transition be

$$M' = \{Q', \Sigma, q_0', \delta', F'\}$$

So as we see in example 2.19.

$$Q' = (\in\text{-closure } (q_0), \in\text{-closure } (q_1), \in\text{-closure } (q_2))$$
$$= (\{q_0, q_1, q_2\}, \{q_1, q_2\}, \{q_2\}) \cong (q_a, q_b, q_c)$$
$$\Sigma = \{0,1\}$$
$$F' = (q_a, q_b, q_c) \text{ as } q_a \to \{q_0, q_1, q_2\}$$
$$q_b \to \{q_1, q_2\}$$
$$q_c \to \{q_2\}$$

and

$$\delta'(q_a, 0) = q_a$$
$$\delta'(q_a, 1) = q_b$$
$$\delta'(q_b, 0) = q_c$$
$$\delta'(q_b, 1) = q_b$$
$$\delta'(q_c, 0) = q_c$$
$$\delta'(q_c, 1) = \phi$$

So transition table is as follows :

δ'/Σ	0	1
$\rightarrow{}^{*}q_a$	q_a	q_b
${}^{*}q_b$	q_c	q_b
${}^{*}q_c$	q_c	ϕ

and transition diagram will be as follows :

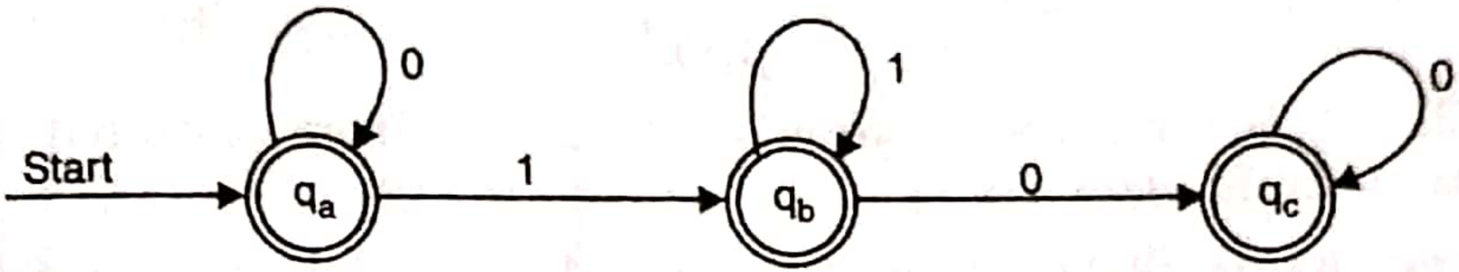

Fig. 2.47. NFA without $\in$-transitions.