NFA to DFA.

Two finite automata N & D are said to be equivalent, if

$$L(N) = L(D)$$

where D represents deterministic finite automata & N represents NFA.

That is, N and D accept the same language. This means,

- any language accepted by D can also be accepted by some N, and
- any language accepted by N can also be accepted by some D.

In worst case, however the smallest D can have 2^n states while the smallest N for the same language has only n states.

IMP.
※ Theorem. : For every NFA, there exists a DFA, that accepts the same language.

or

A language L is accepted by some NFA, if and only if, it is accepted by some DFA.

ANS: This theorem, has two parts to prove:

a. if L is accepted by DFA, D then L is accepted by some NFA N.

b. if L is accepted by NFA, N then L is accepted by some DFA D.

Proof:

Part A: If L is accepted by D, then L is accepted by some N.

Definition (D): A DFA, D is defined by the 5-tuple.

$$D = (Q', \Sigma, \delta', q_0', F').$$

where Q' = Finite set of states

Σ - Finite set of symbols, input alphabet

δ' - Transition function $\delta' : Q' \times \Sigma \rightarrow Q'$

q_0' - Initial state.

F' - set of final states $F' \subseteq Q'$.

Definiton 'N' : An NFA N is defined by the 5-tuple

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

where Q - Finite set of states

Σ - Finite set of symbols, input alphabet

δ - Transition function $\delta : Q \times \Sigma \rightarrow 2^Q$.

q_0 - Initial state.

F - set of final states $F \subseteq Q$.

From the above definitions, it follows that every DFA is also an NFA, which implies that if $w \in L(D)$, then $w \in L(N)$.

Part b : If L is accepted by N, then L is accepted by some D. In order to prove this, let N & D be the NFA and DFA such that, states of D are $Q' = 2^Q$ i.e. all the states of D are subsets of the set of states of N.

The initial state of D is the initial state of N i.e. $q_0' = \{q_0\}$. The final state of D will be any state of D that contains a final state of N. The first next state of D is determined from the initial state followed by successive states.

Here δ' is defined as follows:

$$\delta'(\{q_1, q_2, \cdots q_i\}, a) = \{p_1, p_2, \cdots, p_j\}.$$

if and only if

$$\delta(\{q_1, q_2 \cdots, q_i\}, a) = \{p_1, p_2, \cdots p_j\}.$$

on applying δ to each of $q_1, \cdots q_i$ and taking the union we get new set of states $p_1, \cdots p_i$.

Now we prove that for some Input string 'x',

$\delta'(q_0', x) = \{q_1, q_2, \cdots q_i\}$ iff, $\delta(q_0, x) = \{q_1, q_2 \cdots q_i\}$

We prove this by induction.

Base case: The result is true for $|x| = 0$, if $x = \epsilon$, because

$$\delta'(q_0', x) = \{q_0'\} \text{ and } \delta(q_0, x) = \{q_0\}.$$

Here $\delta'(q_0, x) = \{q_0\}$ iff $\delta(q_0, x) = \{q_0\}$.

Let us assume that the result is true for each string of length n. Now, we shall show that this result is true for any string of length $(n+1)$.

Let $w = xa$ with $|w| = (n+1)$ and $|x| = n$ and $a \in \Sigma$. Thus by induction,

$$\delta'(q_0', x) = \{p_1, p_2, \cdots, p_j\}.$$

$$\text{iff } \delta(q_0, x) = \{p_1, p_2, \cdots, p_j\} \quad \text{-------- (1)}$$

where $\{p_1, p_2 \cdots, p_j\}$ are states of N.

By definition of δ',

$$\delta'(\{p_1, p_2 \cdots, p_j\}, a) = \{r_1, \cdots, r_k\}.$$

$$\text{iff } \delta(\{p_1, p_2 \cdots, p_j\}, a) = \{r_1, \cdots r_k\} \quad \text{---- (2)}$$

Thus,

$$\begin{aligned} \delta'(q_0', xa) &= \delta'(\delta'(q_0', x), a) \\ &= \delta'(\{p_1, \cdots, p_j\}, a) && [\text{using (1)}] \\ &= \{r_1, r_2, \cdots r_k\} && [\text{using (2)}] \end{aligned}$$

Hence the result is true for $|w| = n+1$, when the result is true for a string of length n. Now $\delta'(q_0', x) \in F'$, exactly when $\delta(q_0, x) \in F$.

$$\Rightarrow L(N) = L(D).$$

Thus for every NFA there exists an equivalent DFA, which accepts the same