# NDFA: Non-Deterministic Finite State Automata

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

where $Q$, $\Sigma$, $q_0$ & $F$ are same as defined in DFA.

$\delta : Q \times \{\Sigma \cup \{\lambda\}\} \rightarrow 2^Q$.

## Difference b/w DFA & NDFA

In DFA, outcome is a state, an element of $Q$.

In NDFA, the outcome is a subset of $Q$.

⇒ A string $w \in \Sigma^*$ is accepted by NDFA $M$ if $\delta(q_0, w)$ contains some final states.

## Equivalence b/w DFA & NDFA

1. DFA can simulate behaviour of NDFA by increasing no. of states.
2. Any NDFA is more general machine without being more powerful.

**Theorem:** For every NDFA, there exists a DFA which simulates the behaviour of NDFA. If $L$ is the set accepted by NDFA then there exists a DFA which also ~~exists~~ accepts $L$.

**Q.** Construct DFA equivalent to $M = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_0\})$.

| $\delta$ state/M | 0 | 1 |
|---|---|---|
| $\rightarrow q_0$ | $q_0$ | $q_1$ |
| $q_1$ | $q_1$ | $q_0, q_1$ |

$\delta'([q_1, q_2, \ldots q_k], a) = \bigcup_{i=1}^{k} \delta(q_i, a)$

**Sol<sup></sup>:**

For DFA $M_1$ i) state are subsets of $\{q_0, q_1\}$

i.e. $\phi$, $[q_0]$, $[q_0, q_1]$, $[q_1]$

ii) $[q_0]$ is initial state

iii) $[q_0]$ & $[q_0, q_1]$ both are final states.

(iv) $\delta_1$:

| state/$\Sigma$ | 0 | 1 |
|---|---|---|
| $\phi$ | $\phi$ | $\phi$ |
| $\rightarrow [q_0]$ | $[q_0]$ | $[q_1]$ |
| $[q_1]$ | $[q_1]$ | $[q_0, q_1]$ |
| $[q_0, q_1]$ | $[q_0, q_1]$ | $[q_0, q_1]$ |

→ When M has n state, then corresponding finite automata has $2^n$ state. But consider only those which are reachable from $q_0$ (initial state).

$$\delta'([q_1 \ldots q_k], a) = \bigcup_{i=1}^{k} \delta(q_i, a)$$

**Q.** $M = (\{q_0, q_1, q_2\}, \{a, b\}, \delta, q_0, \{q_2\})$

$\delta$

| state/$\Sigma$ | a | b |
|---|---|---|
| $\rightarrow q_0$ | $q_0, q_1$ | $q_2$ |
| $q_1$ | $q_0$ | $q_1$ |
| $q_2$ | $\phi$ | $q_0, q_1$ |

**Q.** $M = (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_3\})$.

$\delta$.

| | a | b |
|---|---|---|
| $\rightarrow q_0$ | $q_0, q_1$ | $q_0$ |
| $q_1$ | $q_2$ | $q_1$ |
| $q_2$ | $q_3$ | $q_3$ |
| $q_3$ | | $q_2$ |

## Melay & Moore Machines (finite automata with o/p)

**Melay Machine**

$Z(t) = f(q(t), x(t))$

↑ current state ↑ Present i/p

**Moore Machine**

$Z(t) = f(q(t))$

independent of current i/p.

**Def<sup></sup>:** Moore M/c : six tuple $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$

$Q$ : finite set of states

$\Delta$ : o/p alphabets

$\Sigma$ : i/p alphabets

$\delta : Q \times \Sigma \rightarrow Q$

$\lambda$ : o/p function mapping $Q$ into $\Delta$

$q_0$ : initial state.

**Def:** Melay M/c six tuple $(Q, \Sigma, \Delta, \delta, \lambda, q_0)$.

$\lambda : \Sigma \times Q \rightarrow \Delta$

**Ex.** Moore M/c

| Present state | Next state $\delta$ $a=0$ | $a=1$ | o/p $\lambda$ |
|---|---|---|---|
| $\rightarrow q_0$ | $q_3$ | $q_1$ | 0 |
| $q_1$ | $q_1$ | $q_2$ | 1 |
| $q_2$ | $q_2$ | $q_3$ | 0 |
| $q_3$ | $q_3$ | $q_0$ | 0 |

let i/p string 0111

o/p will be

$q_0 \rightarrow q_3 \rightarrow q_0 \rightarrow q_1 \rightarrow q_2$

00010.

Melay M/c

| Present state | Next state $a=0$ state | o/p | $a=1$ state | o/p |
|---|---|---|---|---|
| $\rightarrow q_1$ | $q_3$ | 0 | $q_2$ | 0 |
| $q_2$ | $q_1$ | 1 | $q_4$ | 0 |
| $q_3$ | $q_2$ | 1 | $q_1$ | 1 |
| $q_4$ | $q_4$ | 1 | $q_3$ | 0 |

let i/p string 0011

o/p

$q_1 \rightarrow q_3 \rightarrow q_2 \rightarrow q_4 \rightarrow q_3$

0100.

Imp: For Moore M/c, if i/p string is of length n then o/p string will have length $\underline{n+1}$.

For Melay M/C, if i/p string is of length n, then o/p string will also have length $\underline{n}$.

## Transforming Melay M/c into Moore M/c

Ex: construct Moore M/c from given Melay M/C

Melay M/c

| PS. | NS a=0 state | o/p | a=1 state | o/p |
|---|---|---|---|---|
| → $q_1$ | $q_3$ | 0 | $q_2$ | 0 |
| $q_2$ | $q_1$ | 1 | $q_4$ | 0 |
| $q_3$ | $q_2$ | 1 | $q_1$ | 1 |
| $q_4$ | $q_4$ | 1 | $q_3$ | 0. |

* split $q_i$ into several different states, the # of such state being equal to no. of different o/p associated with $q_i$.

| PS | NS a=0 | | a=1 | |
|---|---|---|---|---|
| → $q_1$ | $q_3$ | 0 | $q_{20}$ | 0 |
| $q_{20}$ | $q_1$ | 1 | $q_{40}$ | 0 |
| $q_{21}$ | $q_1$ | 1 | $q_{40}$ | 0 |
| $q_3$ | $q_{21}$ | 1 | $q_1$ | 1 |
| $q_{40}$ | $q_{41}$ | 1 | $q_3$ | 0 |
| $q_{41}$ | $q_{41}$ | 1 | $q_3$ | 0 |

→

| PS | NS a=0 | a=1 | o/p |
|---|---|---|---|
| → $q_1$ | $q_3$ | $q_{20}$ | 1 |
| $q_{20}$ | $q_1$ | $q_{40}$ | 0 |
| $q_{21}$ | $q_1$ | $q_{40}$ | 1 |
| $q_3$ | $q_{21}$ | $q_1$ | 0 |
| $q_{40}$ | $q_{41}$ | $q_3$ | 0 |
| $q_{41}$ | $q_{41}$ | $q_3$ | 1 |

Here, $q_1$ has o/p 1, means that with i/p Λ we get o/p 1, if the m/c starts at $q_1$. Thus this Moore M/c accepts Λ zero length string ~~which is~~ with o/p 1. This is not accepted by Melay M/c. To overcome this, ~~add~~ either ignore this case or add a new state $q_0$ with o/p 0 as initial state with same transition as $q_1$.

## Transforming Moore M/c to Melay M/c

**Ex. 1** Moore M/c

| PS | NS a=0 | NS a=1 | o/p |
|---|---|---|---|
| → $q_0$ | $q_3$ | $q_1$ | 0 |
| $q_1$ | $q_1$ | $q_2$ | 1 |
| $q_2$ | $q_2$ | $q_3$ | 0 |
| $q_3$ | $q_3$ | $q_0$ | 0 |

⇒ Melay M/c

| PS | NS a=0 | | NS a=1 | |
|---|---|---|---|---|
| → $q_0$ | $q_3$ | 0 | $q_1$ | 1 |
| $q_1$ | $q_1$ | 1 | $q_2$ | 0 |
| $q_2$ | $q_2$ | 0 | $q_3$ | 0 |
| $q_3$ | $q_3$ | 0 | $q_0$ | 0 |

**Ex. 2**

| PS | NS a=0 | NS a=1 | o/p |
|---|---|---|---|
| → $q_1$ | $q_1$ | $q_2$ | 0 |
| $q_2$ | $q_1$ | $q_3$ | 0 |
| $q_3$ | $q_1$ | $q_3$ | 1 |

⇒ Melay M/c

| PS | NS a=0 | | NS a=1 | |
|---|---|---|---|---|
| → $q_1$ | $q_1$ | 0 | $q_2$ | 0 |
| $q_2$ | $q_1$ | 0 | $q_3$ | 1 |
| $q_3$ | $q_1$ | 0 | $q_3$ | 1 |

Since in Melay M/c $q_2$ & $q_3$ are same state then we can delete either of them. So revised M/c is

| PS | NS a=0 | | NS a=1 | |
|---|---|---|---|---|
| → $q_1$ | $q_1$ | 0 | $q_2$ | 0 |
| $q_2$ | $q_1$ | 0 | $q_2$ | 1 |

# Minimization of Finite Automata

**Def 1**: Two states $q_1$ & $q_2$ are equivalent if both $\delta(q_1, x)$ and $\delta(q_2, x)$ are final states or both of them are non final states for all $x \in \Sigma^*$. They also called as Indistinguishable states.

**Def 2**: Two states are k-equivalent ($k \geq 0$) if both $\delta(q_1, x)$ and $\delta(q_2, x)$ are final or both non-final states for all strings x of length k or less.

→ In particular, any two final states are 0-equivalent and any two non-final states are also 0-equivalent.

**Property 1**: The relⁿ defined above are equivalence relⁿ.

**P 2**: These induces partitions of Q. These partitions can be denoted by $\Pi$ and $\Pi_k$ respectively. Elements of $\Pi_k$ are k-equivalence classes.

**P3:** If $q_1$ & $q_2$ are $(k+1)$ equivalent, then they are $k$-equivalent.

**P4:** If $q_1$ & $q_2$ are $k$-equivalent $\forall k \geq 0$, then they are equivalent.

**P5:** $\Pi_n = \Pi_{n+1}$ for some $n$ ($\Pi_n$ denotes the set of equivalence classes under $n$-equivalence)

**Result:** Two states $q_1$ & $q_2$ are $(k+1)$ equivalent if a) they are $k$-equivalent b) $\delta(q_1, a)$ and $\delta(q_2, a)$ are also $k$-equivalent for every $a \in \Sigma$.

**Construction:**

**Step 1:** construction of $\Pi_0$: by def$^n$ of 0 equivalence

$$\Pi_0 = \{Q_1^0, Q_2^0\}.$$

where $Q_1^0$ is set of final states & $Q_2^0 = Q - Q_1^0$.

**Step 2:** Constructions of $\Pi_{k+1}$ from $\Pi_k$:

**Step 3:** $\boxed{\Pi_{n+1} = \Pi_n \quad \text{stop}}$

eg-1

| | 0 | 1 |
|---|---|---|
| $\rightarrow q_0$ | $q_1$ | $q_5$ |
| $q_1$ | $q_6$ | $q_2$ |
| $q_2$ | $q_0$ | $q_2$ |
| $q_3$ | $q_2$ | $q_6$ |
| $q_4$ | $q_7$ | $q_5$ |
| $q_5$ | $q_2$ | $q_6$ |
| $q_6$ | $q_6$ | $q_4$ |
| $q_7$ | $q_6$ | $q_2$ |

$$\Pi_0 = \{\{q_2\}, \{q_0, q_1, q_3, q_4, q_5, q_6, q_7\}\}$$

$$\Pi_1 = \{\{q_2\}, \{q_0, q_4, q_6\}\{q_1, q_7\}, \{q_3, q_5\}\}$$

$$\Pi_2 = \{\{q_2\}, \{q_0, q_4\}, \{q_6\}, \{q_1, q_7\}, \{q_3, q_5\}\}$$

$$\Pi_3 = \Pi_2$$