CSE-233 : Section A

Summer 2020

# DFA and Non-Determinism

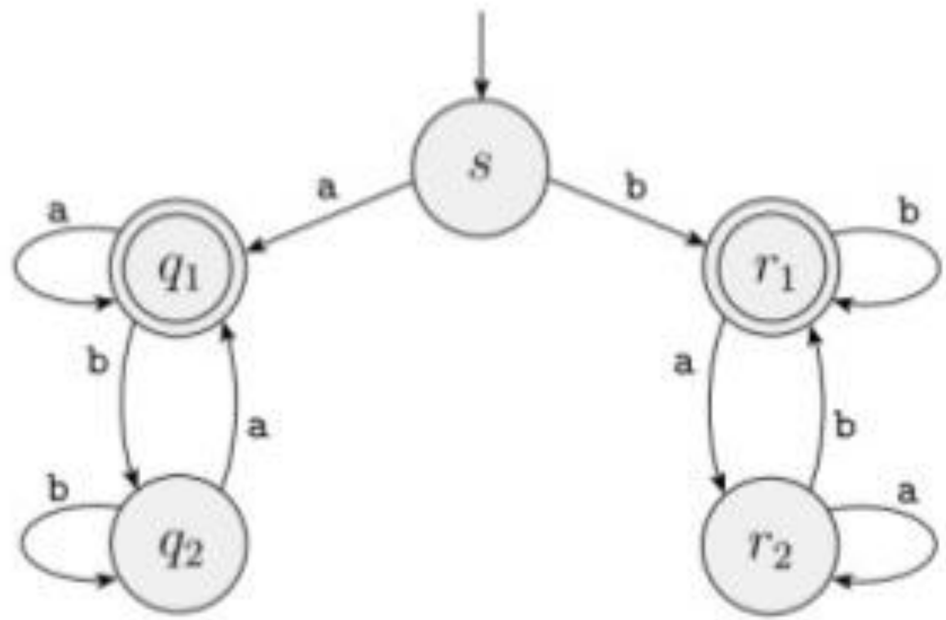
Reference:

Book2 Chapter 1.2

Md. Saidul Hoque Anik
anik@cse.uiu.ac.bd

# Regular Language

A language that is accepted by a finite automata




L = Any string that starts and ends with the same letter

So L is a **regular language**

# Extended Transition function

- If $\delta$ is our transition function, then the extended transition function is denoted by $\hat{\delta}$
- The extended transition function is a function that takes a state $q$ and a string $w$ and returns a state $p$ (the state that the automaton reaches when starting in state $q$ and processing the sequence of inputs $w$)

# Extended Transition function

- $\hat{\delta}(q_0, \epsilon) = q_0$.

| | 0 | 1 |
|---|---|---|
| $* \rightarrow q_0$ | $q_2$ | $q_1$ |
| $q_1$ | $q_3$ | $q_0$ |
| $q_2$ | $q_0$ | $q_3$ |
| $q_3$ | $q_1$ | $q_2$ |

# Extended Transition function

- $\hat{\delta}(q_0, \epsilon) = q_0$.
- $\hat{\delta}(q_0, 1) = \delta\big(\hat{\delta}(q_0, \epsilon), 1\big) = \delta(q_0, 1) = q_1$.

| | 0 | 1 |
|---|---|---|
| $* \rightarrow q_0$ | $q_2$ | $q_1$ |
| $q_1$ | $q_3$ | $q_0$ |
| $q_2$ | $q_0$ | $q_3$ |
| $q_3$ | $q_1$ | $q_2$ |

# Extended Transition function

- $\hat{\delta}(q_0, \epsilon) = q_0$.
- $\hat{\delta}(q_0, 1) = \delta\big(\hat{\delta}(q_0, \epsilon), 1\big) = \delta(q_0, 1) = q_1$.
- $\hat{\delta}(q_0, 11) = \delta\big(\hat{\delta}(q_0, 1), 1\big) = \delta(q_1, 1) = q_0$.

| | 0 | 1 |
|---|---|---|
| $* \to q_0$ | $q_2$ | $q_1$ |
| $q_1$ | $q_3$ | $q_0$ |
| $q_2$ | $q_0$ | $q_3$ |
| $q_3$ | $q_1$ | $q_2$ |

# Extended Transition function

- $\hat{\delta}(q_0, \epsilon) = q_0$.
- $\hat{\delta}(q_0, 1) = \delta\big(\hat{\delta}(q_0, \epsilon), 1\big) = \delta(q_0, 1) = q_1$.
- $\hat{\delta}(q_0, 11) = \delta\big(\hat{\delta}(q_0, 1), 1\big) = \delta(q_1, 1) = q_0$.
- $\hat{\delta}(q_0, 110) = \delta\big(\hat{\delta}(q_0, 11), 0\big) = \delta(q_0, 0) = q_2$.

| | 0 | 1 |
|---|---|---|
| $* \rightarrow q_0$ | $q_2$ | $q_1$ |
| $q_1$ | $q_3$ | $q_0$ |
| $q_2$ | $q_0$ | $q_3$ |
| $q_3$ | $q_1$ | $q_2$ |

# Extended Transition function

| | 0 | 1 |
|---|---|---|
| $* \rightarrow q_0$ | $q_2$ | $q_1$ |
| $q_1$ | $q_3$ | $q_0$ |
| $q_2$ | $q_0$ | $q_3$ |
| $q_3$ | $q_1$ | $q_2$ |

- $\hat{\delta}(q_0, \epsilon) = q_0$.
- $\hat{\delta}(q_0, 1) = \delta\big(\hat{\delta}(q_0, \epsilon), 1\big) = \delta(q_0, 1) = q_1$.
- $\hat{\delta}(q_0, 11) = \delta\big(\hat{\delta}(q_0, 1), 1\big) = \delta(q_1, 1) = q_0$.
- $\hat{\delta}(q_0, 110) = \delta\big(\hat{\delta}(q_0, 11), 0\big) = \delta(q_0, 0) = q_2$.
- $\hat{\delta}(q_0, 1101) = \delta\big(\hat{\delta}(q_0, 110), 1\big) = \delta(q_2, 1) = q_3$.

# Extended Transition function

| | 0 | 1 |
|---|---|---|
| $* \rightarrow q_0$ | $q_2$ | $q_1$ |
| $q_1$ | $q_3$ | $q_0$ |
| $q_2$ | $q_0$ | $q_3$ |
| $q_3$ | $q_1$ | $q_2$ |

- $\hat{\delta}(q_0, \epsilon) = q_0$.
- $\hat{\delta}(q_0, 1) = \delta\big(\hat{\delta}(q_0, \epsilon), 1\big) = \delta(q_0, 1) = q_1$.
- $\hat{\delta}(q_0, 11) = \delta\big(\hat{\delta}(q_0, 1), 1\big) = \delta(q_1, 1) = q_0$.
- $\hat{\delta}(q_0, 110) = \delta\big(\hat{\delta}(q_0, 11), 0\big) = \delta(q_0, 0) = q_2$.
- $\hat{\delta}(q_0, 1101) = \delta\big(\hat{\delta}(q_0, 110), 1\big) = \delta(q_2, 1) = q_3$.
- $\hat{\delta}(q_0, 11010) = \delta\big(\hat{\delta}(q_0, 1101), 0\big) = \delta(q_3, 0) = q_1$.

# Extended Transition function

| | 0 | 1 |
|---|---|---|
| $* \to q_0$ | $q_2$ | $q_1$ |
| $q_1$ | $q_3$ | $q_0$ |
| $q_2$ | $q_0$ | $q_3$ |
| $q_3$ | $q_1$ | $q_2$ |

- $\hat{\delta}(q_0, \epsilon) = q_0$.
- $\hat{\delta}(q_0, 1) = \delta\big(\hat{\delta}(q_0, \epsilon), 1\big) = \delta(q_0, 1) = q_1$.
- $\hat{\delta}(q_0, 11) = \delta\big(\hat{\delta}(q_0, 1), 1\big) = \delta(q_1, 1) = q_0$.
- $\hat{\delta}(q_0, 110) = \delta\big(\hat{\delta}(q_0, 11), 0\big) = \delta(q_0, 0) = q_2$.
- $\hat{\delta}(q_0, 1101) = \delta\big(\hat{\delta}(q_0, 110), 1\big) = \delta(q_2, 1) = q_3$.
- $\hat{\delta}(q_0, 11010) = \delta\big(\hat{\delta}(q_0, 1101), 0\big) = \delta(q_3, 0) = q_1$.
- $\hat{\delta}(q_0, 110101) = \delta\big(\hat{\delta}(q_0, 11010), 1\big) = \delta(q_1, 1) = q_0$.

# More Design Examples

Draw a DFA for the language accepting strings-

1. Having exactly 2 a’s over alphabet {a, b}
2. At least 1 b over alphabet {a, b}
3. At most 1 b over alphabet {a, b}
4. Length of at least 2 over alphabet {a, b}
5. Length of at most 2 over alphabet {a, b}

# Non-Deterministic Finite Automata

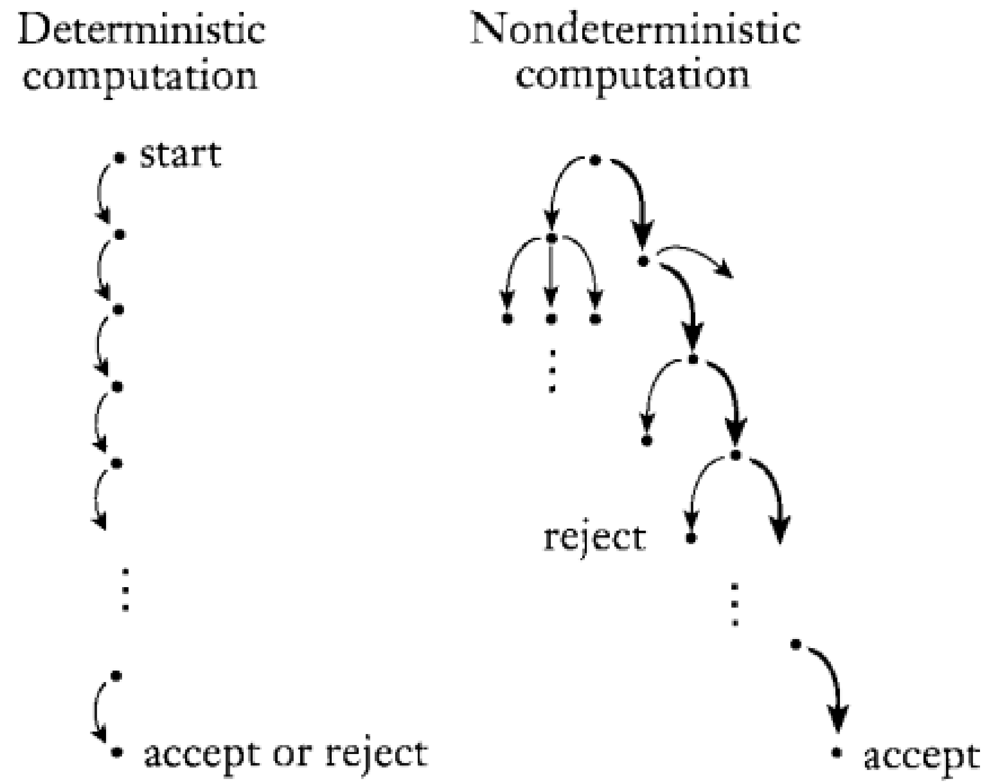



FIGURE 1.28
Deterministic and nondeterministic computations with an accepting branch

# NFA Example

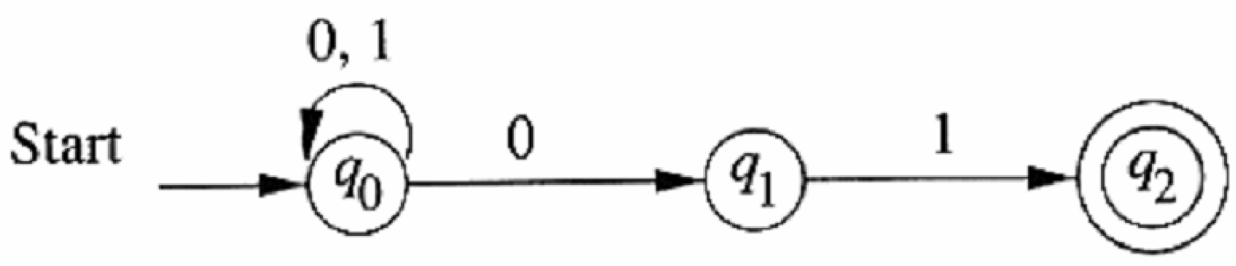



Figure 2.9: An NFA accepting all strings that end in 01

- Each state can have zero, one, or more transitions out labeled by the same symbol
  - Eg, for a single input 1, we can guess that the next state can either be $q_0$ or $q_1$

- **What will be the tree for input: 00101?**

# NFA Example

**For input: 00101**

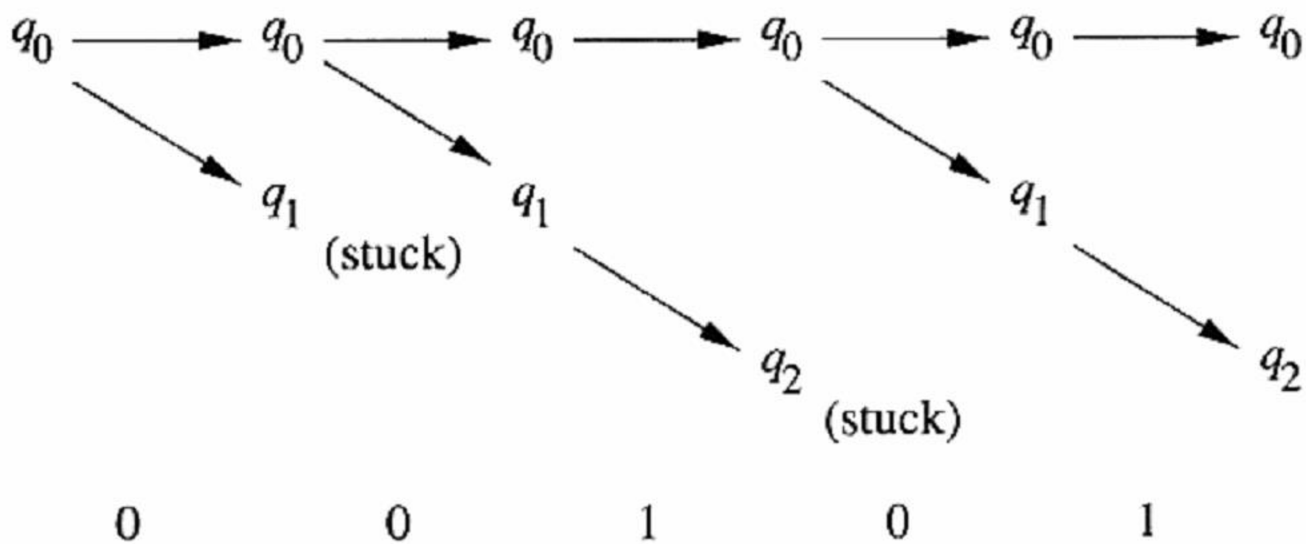


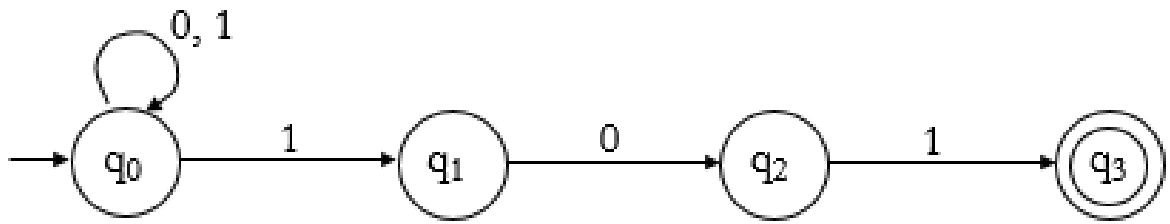

Figure 2.10: The states an NFA is in during the processing of input sequence 00101

# Formal Definition of NFA

A ***nondeterministic finite automaton*** is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

1. $Q$ is a finite set of states,
2. $\Sigma$ is a finite alphabet,
3. $\delta \colon Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ is the transition function,
4. $q_0 \in Q$ is the start state, and
5. $F \subseteq Q$ is the set of accept states.

# Example




Alphabet = {0, 1}

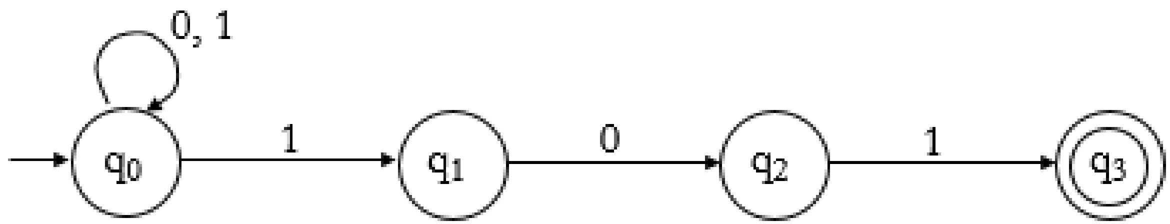
start state $Q$ = {$q_0$, $q_1$, $q_2$, $q_3$}

initial state $q_0$

accepting states $F$ = {$q_3$}

Transition Function = ?

# Example




Alphabet = {0, 1}

start state $Q = \{q_0, q_1, q_2, q_3\}$

initial state $q_0$

accepting states $F = \{q_3\}$

Transition Function:

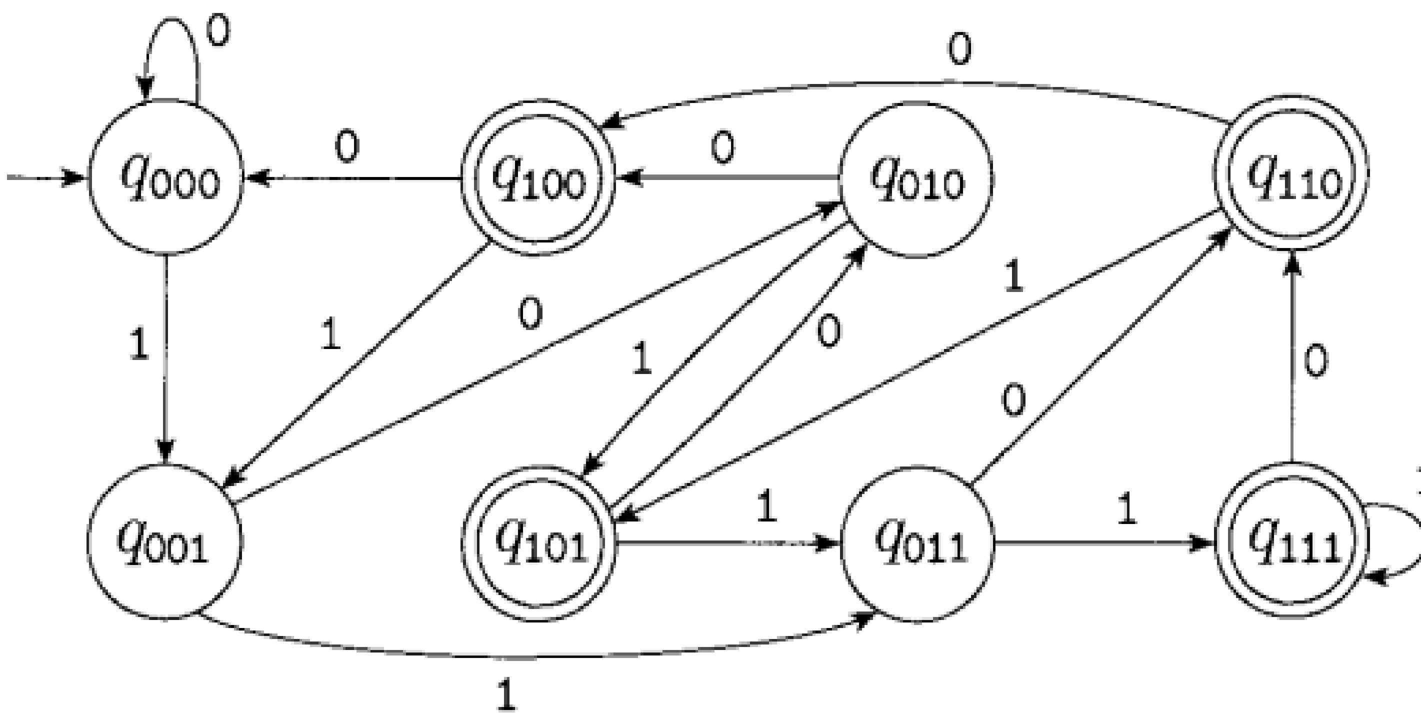
| | inputs | |
|---|---|---|
| states | 0 | 1 |
| $q_0$ | $\{q_0\}$ | $\{q_0, q_1\}$ |
| $q_1$ | $\{q_2\}$ | $\varnothing$ |
| $q_2$ | $\varnothing$ | $\{q_3\}$ |
| $q_3$ | $\varnothing$ | $\varnothing$ |

# NFA vs. DFA

Design a Machine that detected all strings over {0, 1} containing 1 in the third position from the end

# NFA vs. DFA

Design a Machine that detected all strings over {0, 1} containing 1 in the third position from the end



# NFA vs. DFA

Design a Machine that detected all strings over {0, 1} containing 1 in the third position from the end

Let $A$ be the language consisting of all strings over $\{0,1\}$ containing a 1 in the third position from the end (e.g., 000100 is in $A$ but 0011 is not). The following four-state NFA $N_2$ recognizes $A$.

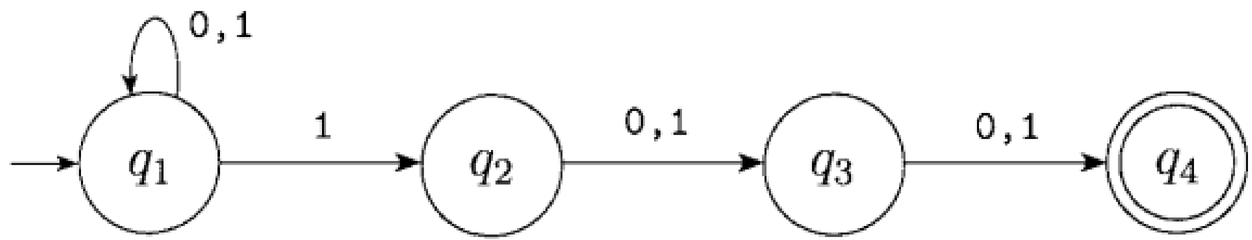



**FIGURE 1.31**
The NFA $N_2$ recognizing $A$