AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Science and Technology

AIUB

DEPARTMENT OF COMPUTER SCIENCE

CSC3113: THEORY OF COMPUTATION

Lecture: # 6 Week: # 3 Semester: Spring 2022-2023

NON-DETERMINISTIC FINITE AUTOMATON (NFA)

Instructor: Shakila Rahman,

Department of Computer Science, Faculty of Science & Technology.

Shakila.Rahman@aiub.edu

LECTURE OUTLINE

- DFA-NFA Equivalence.
- Nondeterministic Finite Automata (NFA).
  - Practice, solve exercise of NFA.
  - Closure under regular operations.

Learning Objective

- Equivalence of DFA & NFA.
- Understand, learn & practice with example
  - Practice designing NFA.
  - Understanding closure under regular operation for NFA.

LEARNING OUTCOME

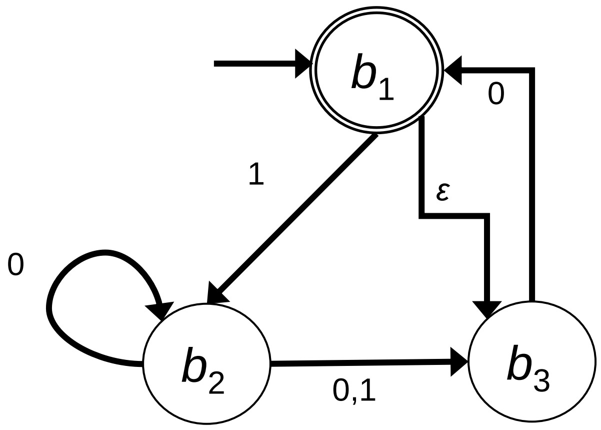
ALL OUTCOME ARE REPRESENTED WITH EXAMPLES

- Equivalence of DFA & NFA.
- Conversion from NFA to DFA.
- Practice & Design of NFA
- Closure under regular operations for NFA.

EQUIVALENCE BETWEEN NFA & DFA

- Every NFA has an equivalent DFA.
- Let $N = (Q, \Sigma, \delta, q_0, F)$ be the NFA recognizing some language A.
- Construct a DFA $M = (Q', \Sigma, \delta', q_0', F')$ which recognizes A.
  - $Q' = \mathcal{P}(Q)$, power set of Q.
    - Every state of M is a set of states of N.
  - Let $E(R) = \{q \mid q$ can be reached from $R \subseteq Q$ by traveling along 0 or more ε arrows, including the members of R themselves$\}$.
  - For $B \in Q'$ and $a \in \Sigma$, $\delta'(B, a) = \{q \in Q' \mid q \in E(\delta(r, a))$ for some $r \in B\}$.
    - Each state B may go to a set of states after reading any symbol a. So, we take the union of all these sets.
  - $q_0' = E(\{q_0\})$.
    - M starts at the state corresponding to the collection containing all the possible states that can be reached from the start state of N along with the ε arrows.
  - $F' = \{D \in Q' \mid D$ contains an accept state of $N\}$.

NFA-DFA Equivalence



Let, the above NFA $N_2=(Q_2, \Sigma, \delta_2, b_1, F_2)$.

$\boldsymbol{Q_2} = \{b_1,b_2,b_3\}$; $\boldsymbol{\Sigma} = \{0,1\}$;

$\boldsymbol{b_1}$ = start state; $\boldsymbol{F_2} = \{b_1\}$.

$\boldsymbol{\delta_2}$ is given as –

	0	1	ε
b_1	ϕ	$\{b_2\}$	$\{b_3\}$
b_2	$\{b_2, b_3\}$	$\{b_3\}$	ϕ
b_3	$\{b_1\}$	ϕ	ϕ

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

$\boldsymbol{Q} = \mathcal{P}(Q_2) = \mathcal{P}(\{b_1,b_2,b_3\})$

$\boldsymbol{Q} = \{ \phi, \{b_1\}, \{b_2\}, \{b_3\}, \{b_1, b_2\}, \{b_1, b_3\},$
$\{b_2, b_3\}, \{b_1, b_2, b_3\} \};$

δ is given as –	0	1
ϕ	ϕ	ϕ
$\{b_1\}$	ϕ	$\{b_2\}$
$\{b_2\}$	$\{b_2, b_3\}$	$\{b_3\}$
$\{b_3\}$	$\{b_1, b_3\}$	ϕ
$\{b_1, b_2\}$	$\{b_2, b_3\}$	$\{b_2, b_3\}$
$\{b_1, b_3\}$	$\{b_1, b_3\}$	$\{b_2\}$
$\{b_2, b_3\}$	$\{b_1, b_2, b_3\}$	$\{b_3\}$
$\{b_1, b_2, b_3\}$	$\{b_1, b_2, b_3\}$	$\{b_2, b_3\}$

$\boldsymbol{\Sigma} = \{0, 1\}$.

$\boldsymbol{q_0} = E(\{b_1\}) = \{b_1, b_3\}$ is the start state;

$\boldsymbol{F} = \{\{b_1\}, \{b_1, b_2\}, \{b_1, b_3\}, \{b_1, b_2, b_3\}\}$.

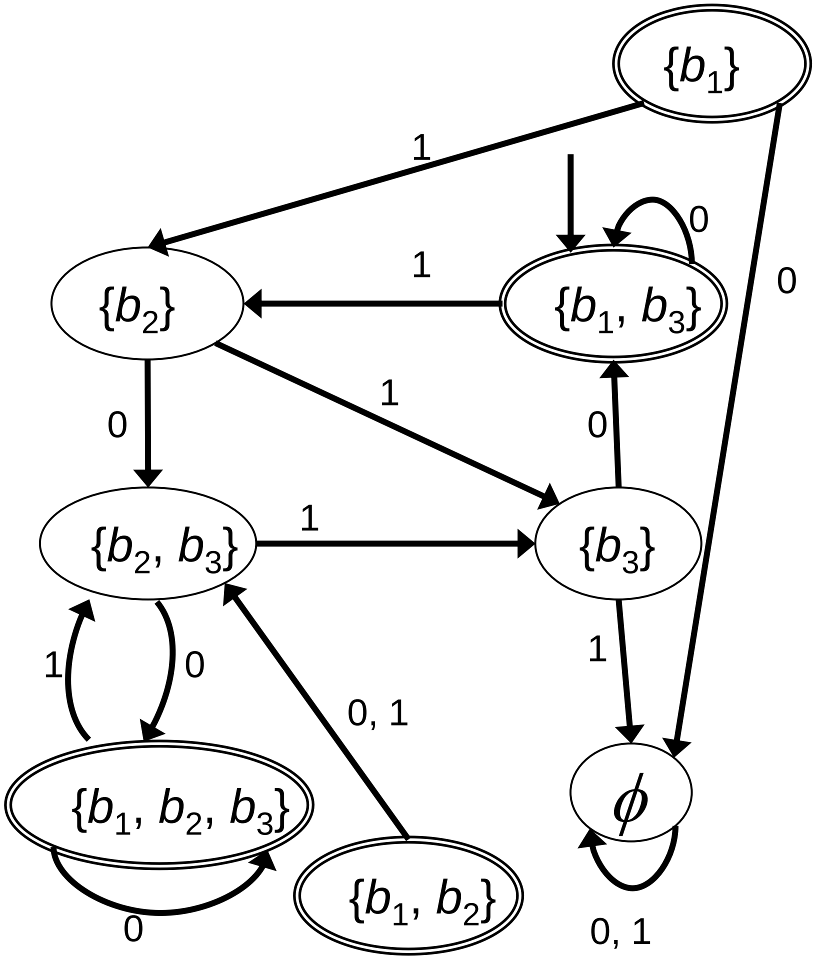
NFA-DFA EQUIVALENCE

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

$Q = \{b_1, b_2, b_3\} = \mathcal{P}(Q)$

$Q = \{\ \phi, \{b_1\}, \{b_2\}, \{b_3\}, \{b_1, b_2\}, \{b_1, b_3\},$
$\{b_2, b_3\}, \{b_1, b_2, b_3\}\ \};$

δ is given as –	0	1
ϕ	ϕ	ϕ
$\{b_1\}$	ϕ	$\{b_2\}$
$\{b_2\}$	$\{b_2, b_3\}$	$\{b_3\}$
$\{b_3\}$	$\{b_1, b_3\}$	ϕ
$\{b_1, b_2\}$	$\{b_2, b_3\}$	$\{b_2, b_3\}$
$\{b_1, b_3\}$	$\{b_1, b_3\}$	$\{b_2\}$
$\{b_2, b_3\}$	$\{b_1, b_2, b_3\}$	$\{b_3\}$
$\{b_1, b_2, b_3\}$	$\{b_1, b_2, b_3\}$	$\{b_2, b_3\}$



Remove the states with no incoming arrows.

$\Sigma = \{0, 1\}$.

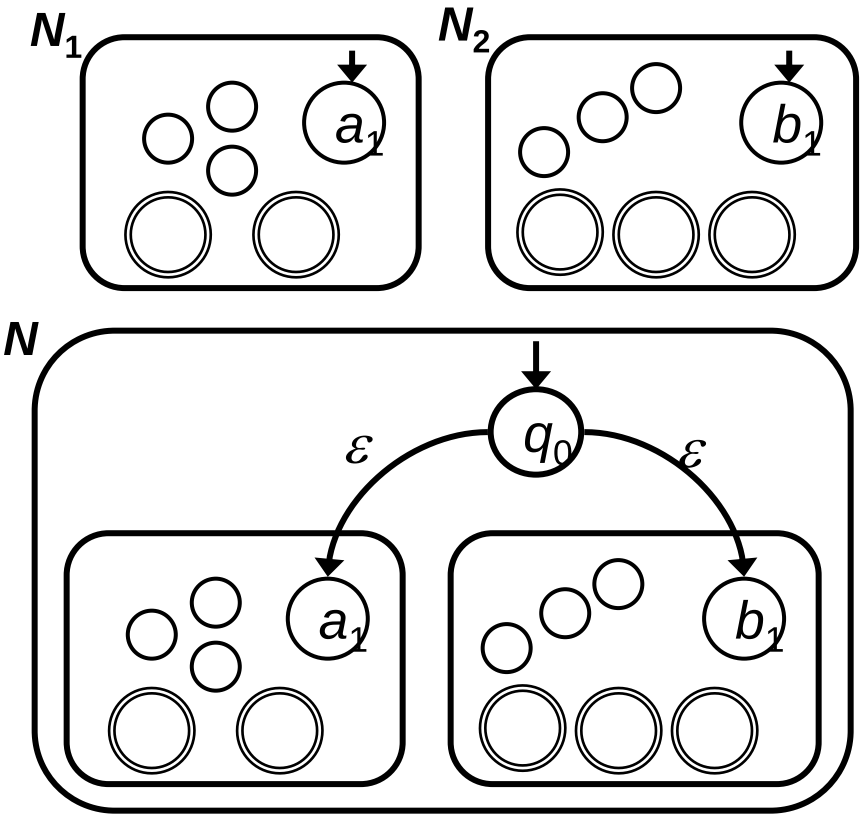
$q_0 = E(\{b_1\}) = \{b_1, b_3\}$ is the start state;

$F = \{\{b_1\}, \{b_1, b_2\}, \{b_1, b_3\}, \{b_1, b_2, b_3\}\}$.

Closure Under Regular Operations

- Let,
  - $N_1 = (Q_1, \Sigma, \delta_1, a_1, F_1)$ recognizes A_1.
  - $N_2 = (Q_2, \Sigma, \delta_2, b_1, F_2)$ recognizes A_2.
- **UNION**: Construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize $A_1 \cup A_2$.
  - $Q = \{q_0\} \cup Q_1 \cup Q_2$.
  - q_0 is the starting state.
  - $F = F_1 \cup F_2$.
  - For any $q \in Q$ and $a \in \Sigma_\varepsilon$,

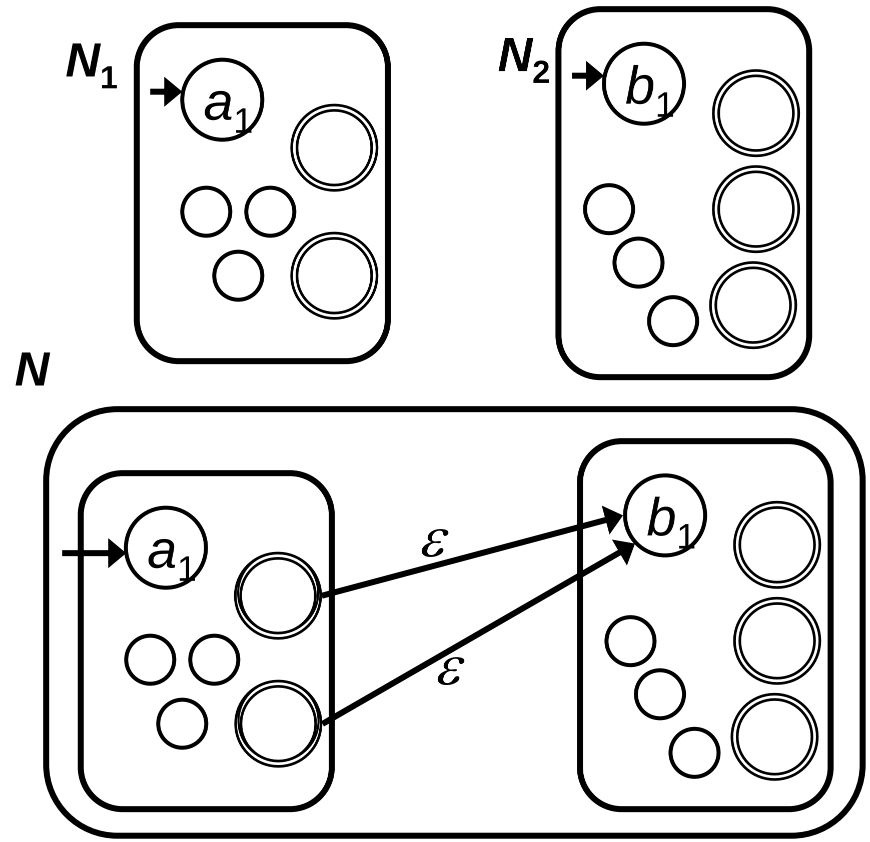
$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \\ \delta_2(q,a) & q \in Q_2 \\ \{a_1, b_1\} & q = q_0 \text{ and } a = \varepsilon \\ \phi & q = q_0 \text{ and } a \neq \varepsilon \end{cases}$$



CLOSURE UNDER REGULAR OPERATIONS

- Let,
  - $N_1 = (Q_1, \Sigma, \delta_1, a_1, F_1)$ recognizes A_1.
  - $N_2 = (Q_2, \Sigma, \delta_2, b_1, F_2)$ recognizes A_2.
- **CONCATENATION**: Construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize $A_1 \circ A_2$.
  - $Q = Q_1 \cup Q_2$.
  - $q_0 = a_1$.
  - $F = F_2$.
  - For any $q \in Q$ and $a \in \Sigma_\varepsilon$,

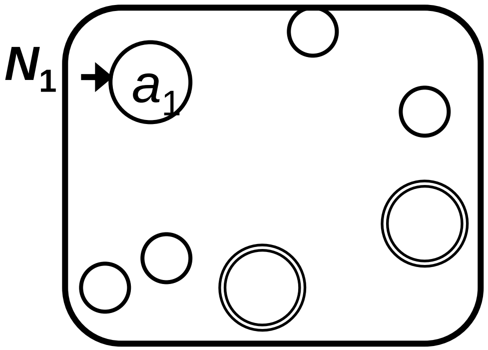
$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \text{ and } a \neq \varepsilon \\ \delta_1(q,a) \cup \{b_1\} & q \in F_1 \text{ and } a = \varepsilon \\ \delta_2(q,a) & q \in Q_2 \end{cases}$$

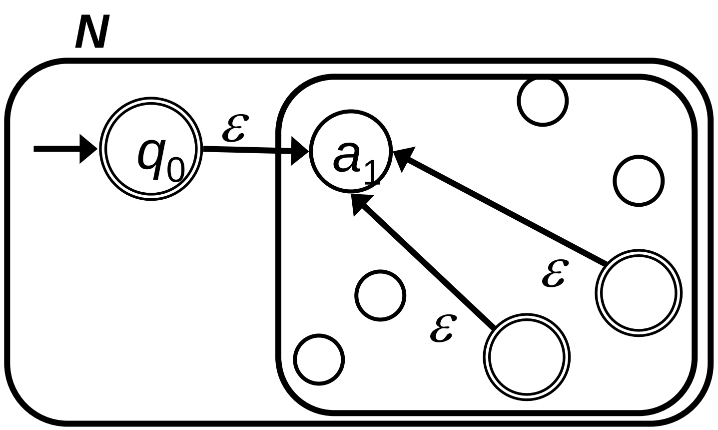


CLOSURE UNDER REGULAR OPERATIONS

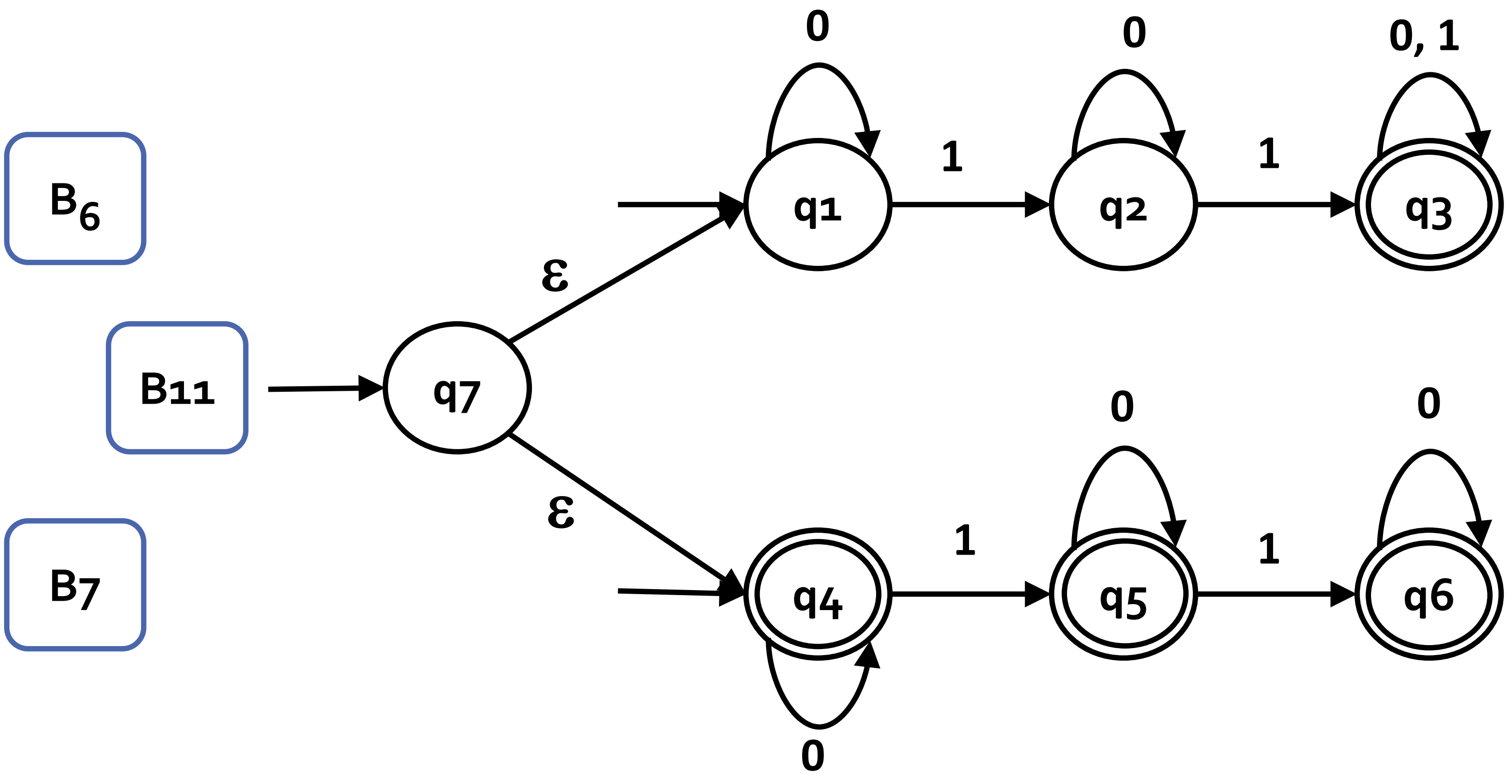
- Let,
  - $N_1 = (Q_1, \Sigma, \delta_1, a_1, F_1)$ recognizes A_1.
- **STAR**: Construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize $A_1{}^*$.
  - $Q = \{q_0\} \cup Q_1$.
  - q_0 is the new start state.
  - $F = \{q_0\} \cup F_1$.
  - For any $q \in Q$ and $a \in \Sigma_\varepsilon$,

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in Q_1 \text{ and } q \notin F_1 \\ \delta_1(q,a) & q \in F_1 \text{ and } a \neq \varepsilon \\ \delta_1(q,a) \cup \{a_1\} & q \in F_1 \text{ and } a = \varepsilon \\ \{a_1\} & q = q_0 \text{ and } a = \varepsilon \\ \phi & q = q_0 \text{ and } a \neq \varepsilon \end{cases}$$

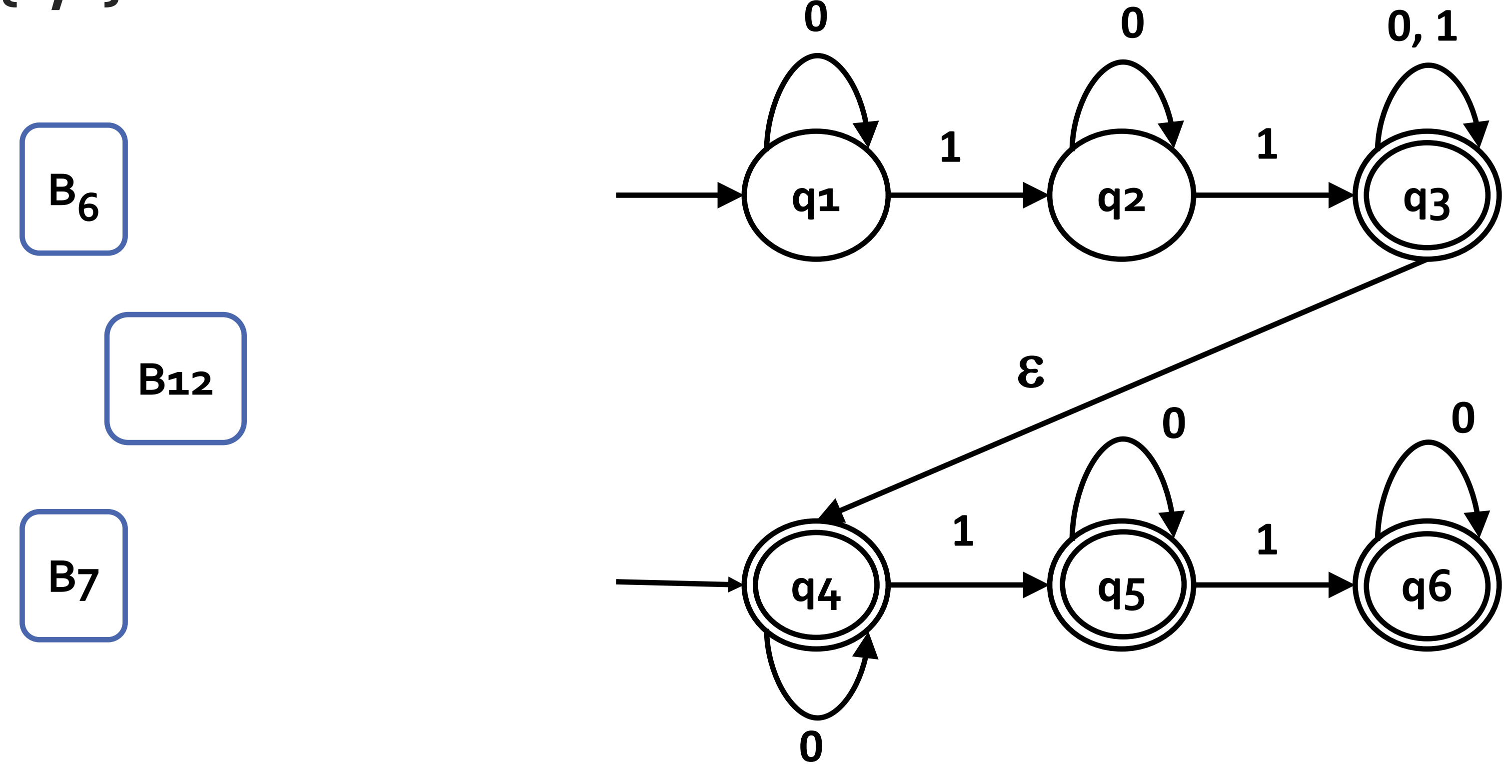




B_6 ={w| w has at least two 1s}.
B_7 ={w| w has at most two 1s}.
B_{11} ={w| w has at least two 1s or w has at most two 1s}.
Σ = {0,1}

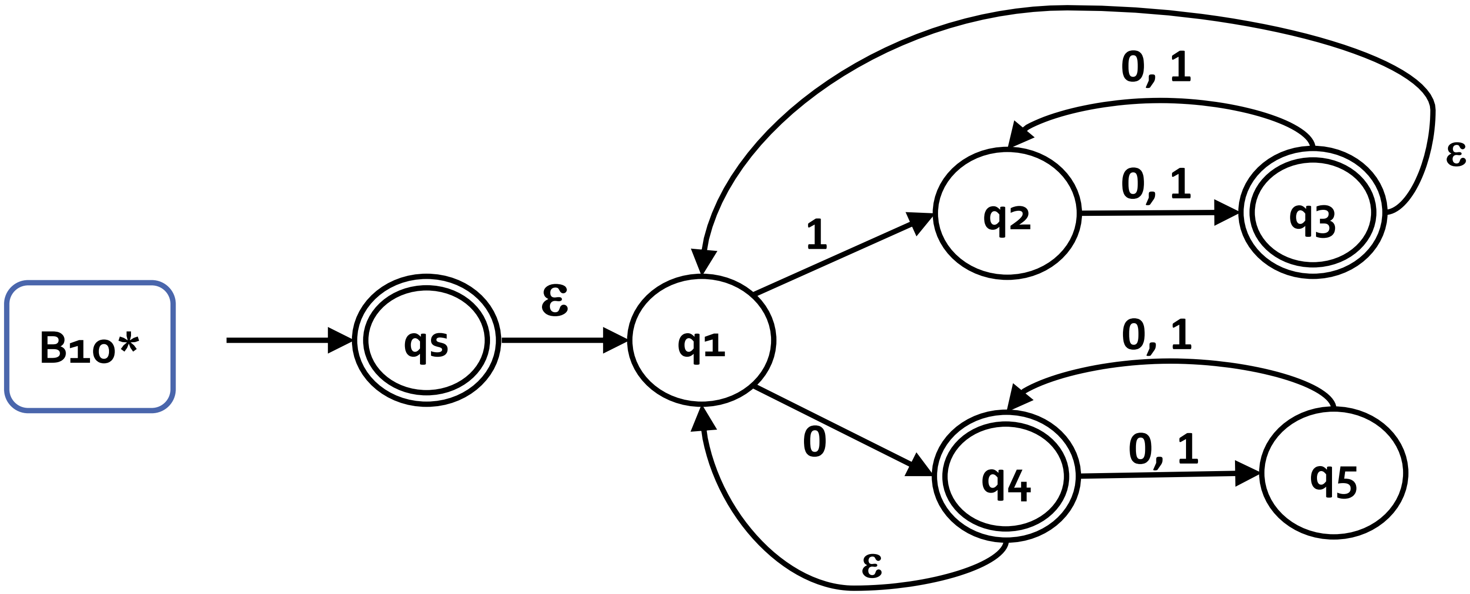


B_6 ={w| w has at least two 1s}.
B7 ={w| w has at most two 1s}.
B12 ={w| w has at least two 1s followed by w has at most two 1s}.
∑ = {0,1}



B10 ={w| w starts with a 1 and has even length or w starts with a 0 and has odd length}.

Find B10* for ∑ = {0,1}



References

All exercises for Finite Automata & Regular Language

- Same as previous Lecture…
- Elements of the Theory of Computation, Papadimitriou (2nd ed), All exercises.
- Introduction to Automata Theory, Languages, and Computations, Hopcroft, All exercises.