CSC3113: THEORY OF COMPUTATION

Lecture: # 5 Week: # 3 Semester: Spring 2022-2023

NON-DETERMINISTIC FINITE AUTOMATON (NFA)

Instructor: Shakila Rahman, Lecturer,

Department of Computer Science, Faculty of Science & Technology.

Shakila.Rahman@aiub.edu

Lecture Outline

- Nondeterministic Finite Automata (NFA).
  - Running NFA, NFA Tree.
  - Formal Definition of NFA.
  - Practice, solve exercise of NFA.

LEARNING OBJECTIVE

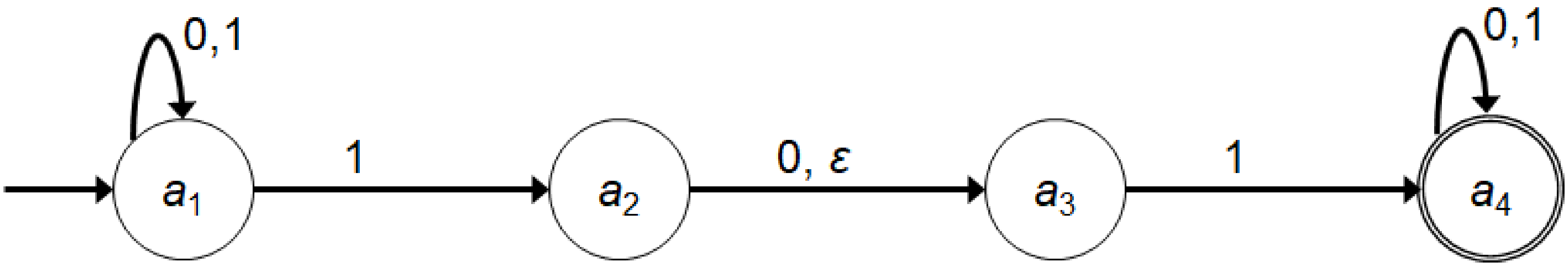
- Understand, learn & practice with example
  - Formal Definition of Nondeterministic Finite Automata (NFA)
  - Practice designing NFA.

LEARNING OUTCOME

ALL OUTCOME ARE REPRESENTED WITH EXAMPLES

- Understand, learn & formal definition of NFA.
- Practice & Design of NFA

Nondeterministic Finite Automata (NFA)



State Diagram and the Properties of NFA

- We already know DFA, so it would be sufficient to look into the differences of properties between the two.
- In NFA a state may have –
  - Zero or more exiting arrows for each alphabet symbol.
  - Zero or more exiting arrows with the label ε.
- So we can see that, not all steps of a computation follows in a unique way from the preceding step. There can be multiple choices to move from one state to another with a symbol. That's the reason it's computation is called nondeterministic.

Running an NFA

- If we encounter a state with multiple ways to proceed –
  - The machine splits into multiple copies of itself and follows all the possibilities in parallel.
  - Each copy of the machine takes one of the possible ways to proceed and continues as before.
  - If there are subsequent choices, the machine splits again.
- If a state with an ε symbol on an exiting arrow is encountered without reading any input, the machine splits into multiple copies,
  - one following each of the exiting ε-labeled arrows and
  - one staying in the current state.
- If the next input symbol doesn't appear on any of the arrows exiting the state occupied by a copy of the machine, that copy of the machine dies, along with the branch of the computation associated with it.
- If any one of these copies of the machine is in an accept state at the end of the input, the NFA accepts the input strings.
- So, nondeterminism may be viewed as a kind of parallel computation wherein several processes can be running concurrently.
- If at least one of these processes accepts then the entire computation accepts.

RUNNING AN NFA

- Another way of viewing a nondeterministic computation is as a tree of possibilities.
  - The root corresponds to the start of the computation.
  - Every branching point in the tree corresponds to a point in the computation at which the machine has multiple choices.
  - The machine accepts if at least one of the computation branches ends in an accept state.

TREE REPRESENTATION

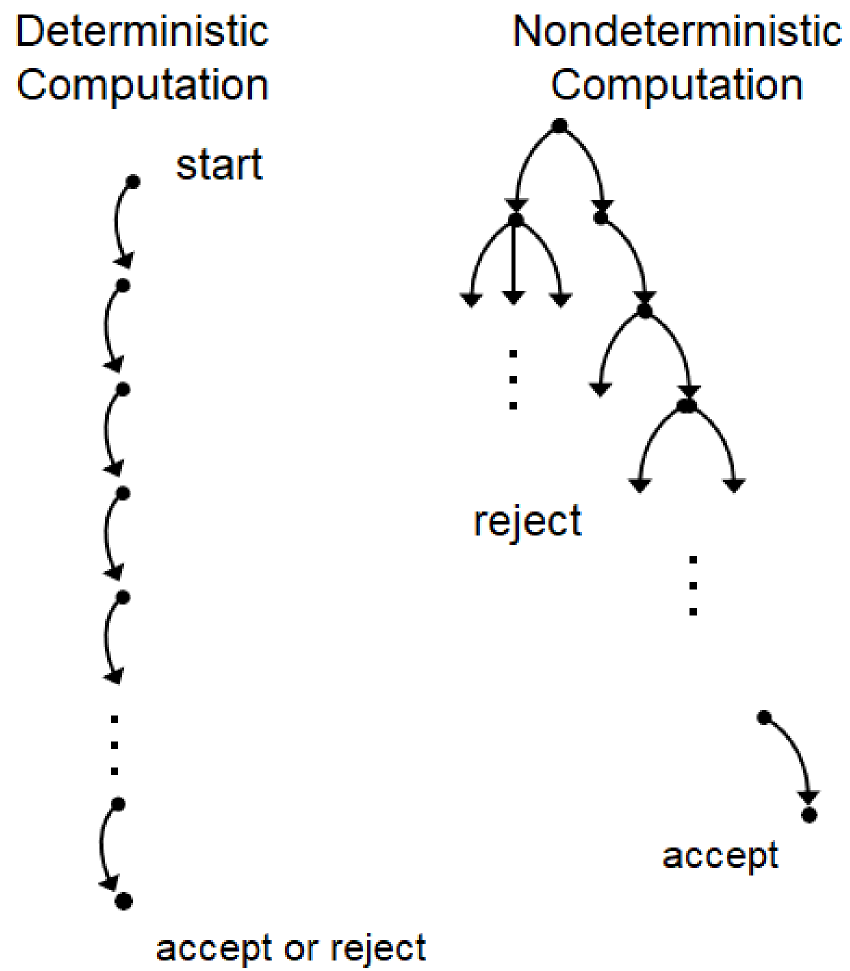
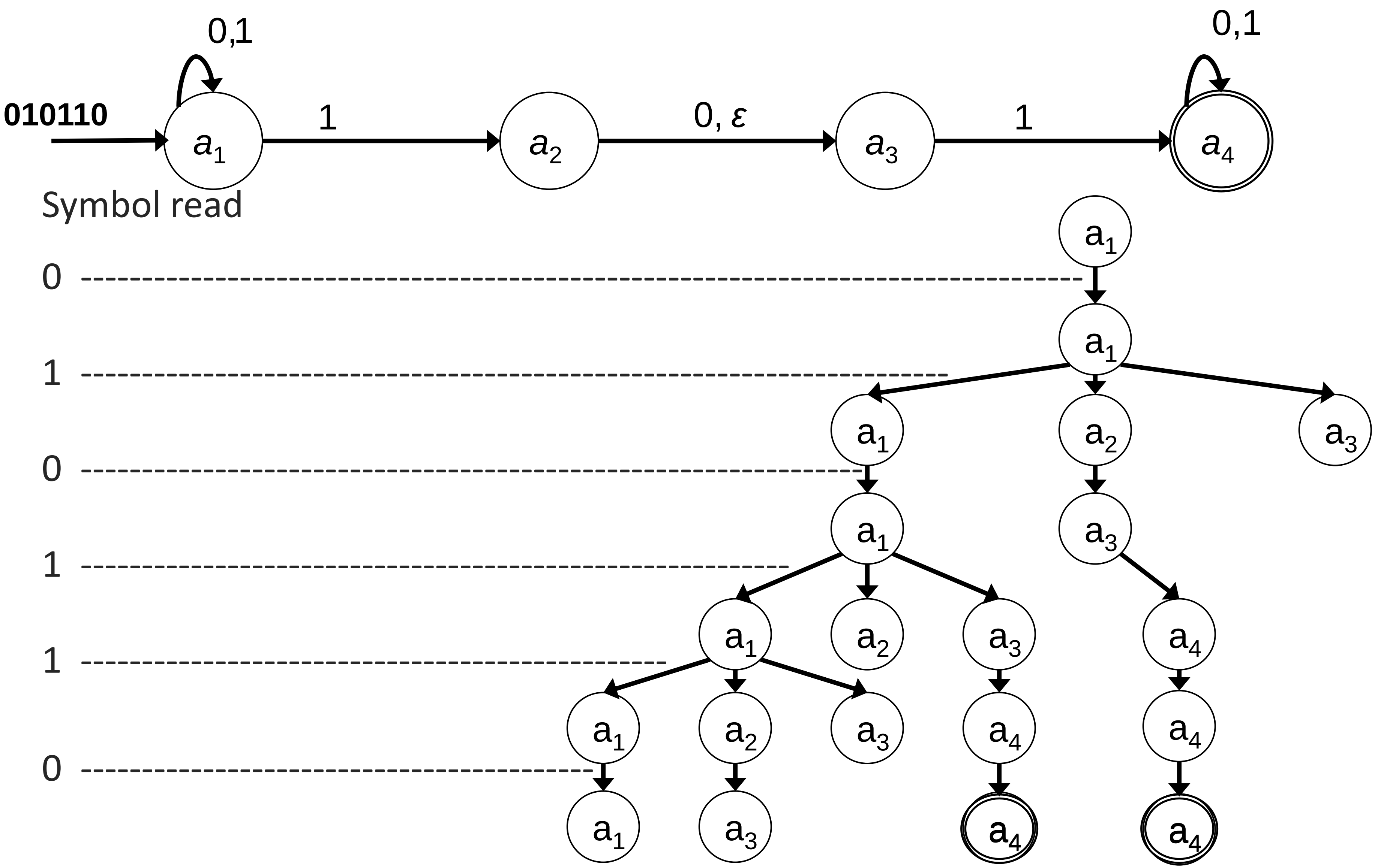


Figure: Deterministic and nondeterministic computations with an accepting branch

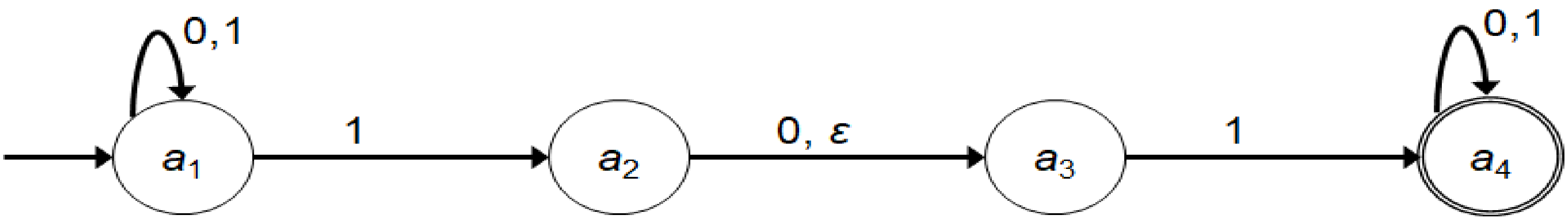
SIMULATION – NFA TREE



Formal Definition of NFA

- NFA is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$
  - Q is a finite set of states.
  - Σ is a finite alphabet.
  - $\delta : Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$.
    - The transition function takes a state and an input symbol or the empty string ($\Sigma_\varepsilon = \Sigma \cup \varepsilon$) and produces the set of possible next states ($\mathcal{P}(Q)$ is the power set of Q).
  - $q_0 \in Q$ is the start state.
  - $F \subseteq Q$ is the set of accepted states.

NONDETERMINISTIC FINITE AUTOMATA (NFA)



FORMAL DEFINITION

- Let, the above NFA $N_1 = (Q_1, \Sigma, \delta_1, a_1, F_1)$.
  - $Q_1 = \{a_1, a_2, a_3, a_4\}$.
  - $\Sigma = \{0, 1\}$.
  - δ_1

δ_1	0	1	ε
a_1	$\{a_1\}$	$\{a_1, a_2\}$	ϕ
a_2	$\{a_3\}$	ϕ	$\{a_3\}$
a_3	ϕ	$\{a_4\}$	ϕ
a_4	$\{a_4\}$	$\{a_4\}$	ϕ

  - a_1 is the start state.
  - $F_1 = \{a_4\}$.

δ_1 in form $\delta : Q \times \Sigma_\varepsilon \rightarrow \mathcal{P}(Q)$

$\delta_1(a_1, 0) = \{a_1\}$
$\delta_1(a_1, 1) = \{a_1, a_2\}$
$\delta_1(a_1, \varepsilon) = \phi$

$\delta_1(a_2, 0) = \{a_3\}$
$\delta_1(a_2, 1) = \phi$
$\delta_1(a_2, \varepsilon) = \{a_3\}$

$\delta_1(a_3, 0) = \phi$
$\delta_1(a_3, 1) = \{a_4\}$
$\delta_1(a_3, \varepsilon) = \phi$

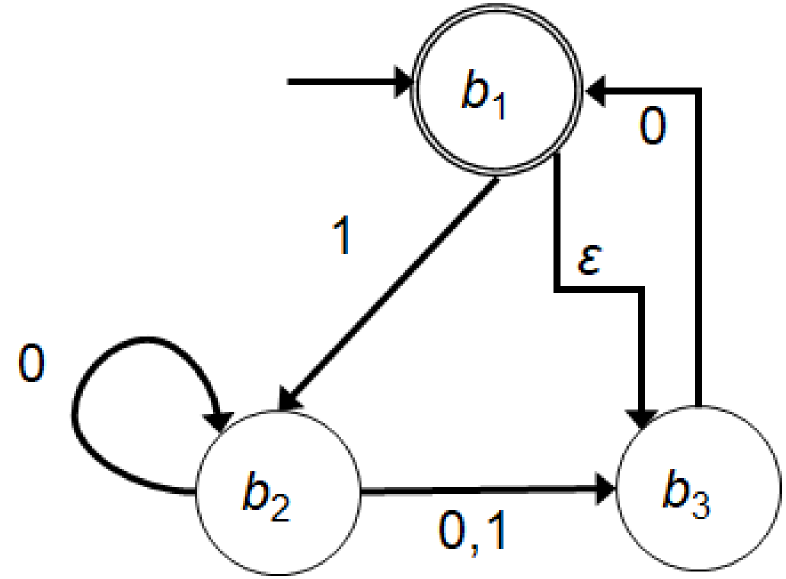
$\delta_1(a_4, 0) = \{a_4\}$
$\delta_1(a_4, 1) = \{a_4\}$
$\delta_1(a_4, \varepsilon) = \phi$

$Q \times \Sigma_\varepsilon = \{(a_1, 0), (a_1, 1), (a_1, \varepsilon), (a_2, 0), (a_2, 1), (a_2, \varepsilon), (a_3, 0), (a_3, 1), (a_3, \varepsilon), (a_4, 0), (a_4, 1), (a_4, \varepsilon)\}$

$\mathcal{P}(Q) = \{\phi, \{a_1\}, \{a_2\}, \{a_3\}, \{a_4\}, \{a_1,a_2\}, \{a_1,a_3\}, \{a_1,a_4\}, \{a_2,a_3\}, \{a_2,a_4\}, \{a_3,a_4\},$
$\{a_1,a_2,a_3\}, \{a_1,a_2,a_4\}, \{a_1,a_3,a_4\}, \{a_2,a_3,a_4\}, \{a_1,a_2,a_3, a_4\}\}$

NONDETERMINISTIC FINITE AUTOMATA (NFA)

EXAMPLE



- Let, the above NFA $N_2 = (Q_2, \Sigma, \delta_2, b_1, F_2)$.
  - $Q_2 = \{b_1, b_2, b_3\}$.
  - $\Sigma = \{0, 1\}$.
  - δ_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

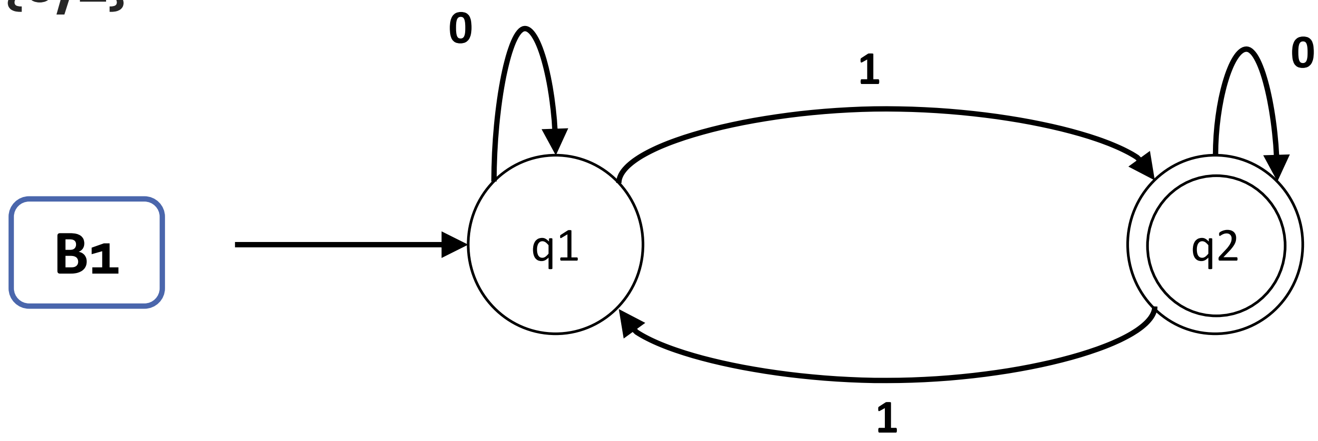
  - b_1 is the start state.
  - $F_2 = \{b_1\}$.

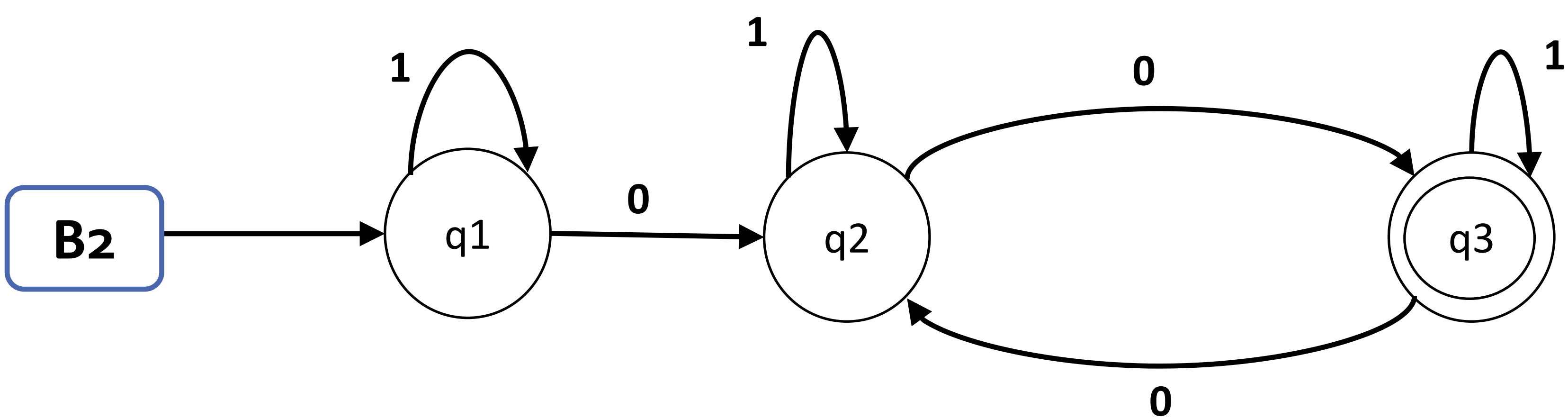
PRACTICE NFA

B1 ={w : w is a binary string containing an odd number of 1s}.

B2 ={w : w is a binary string containing an even number of 0s}.

∑ = {0,1}



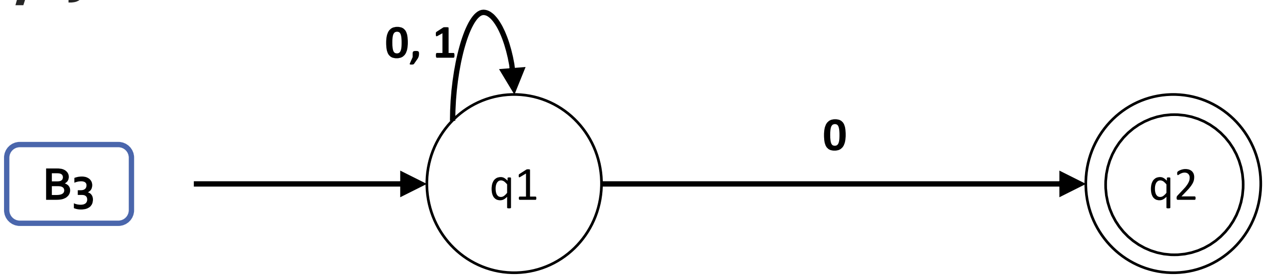


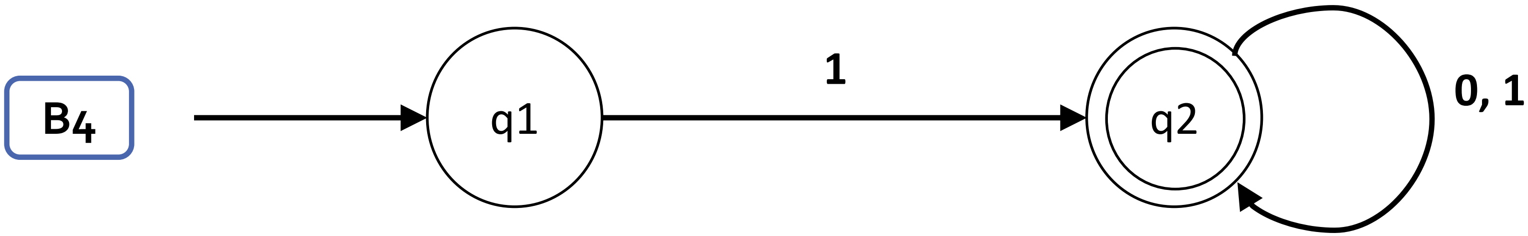
B3 ={w| w ends with a 0}.

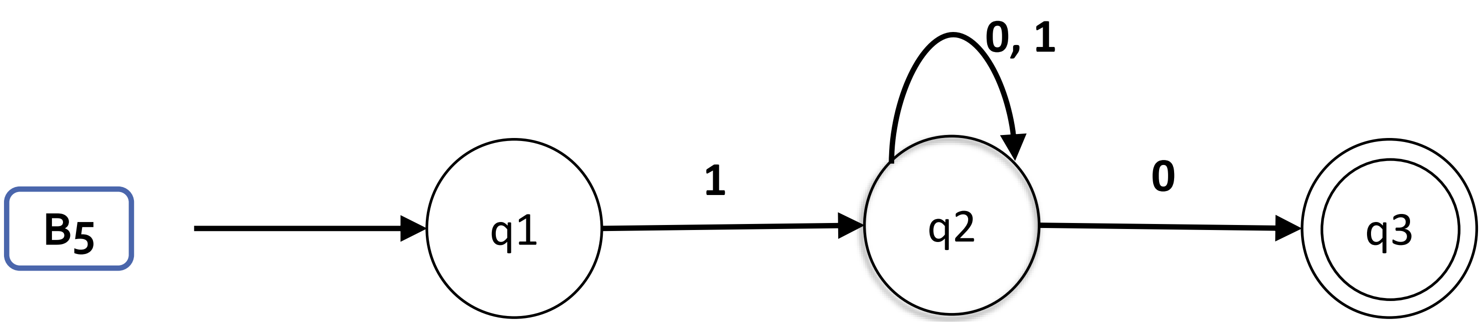
B4 ={w| w begins with a 1}.

B5 ={w| w begins with a 1 and ends with a 0}.

∑ = {0,1}

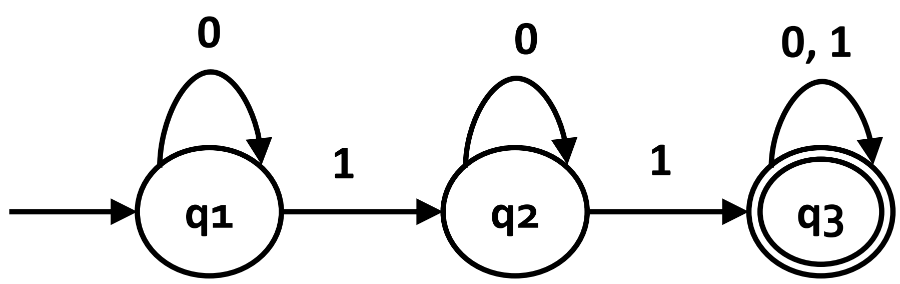




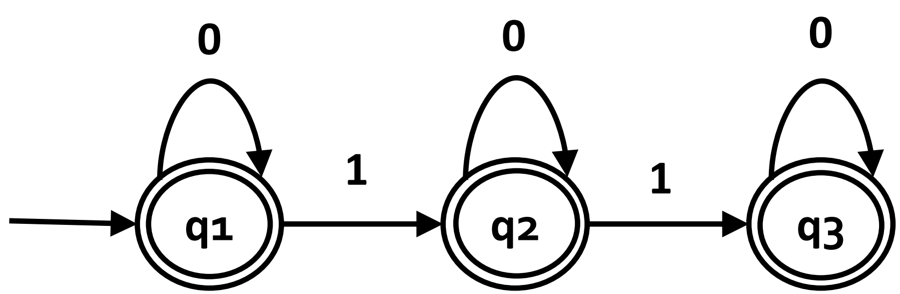


B_6 ={w| w has at least two 1s}.
B_7 ={w| w has at most two 1s}.
$\sum$ = {0,1}

B_6



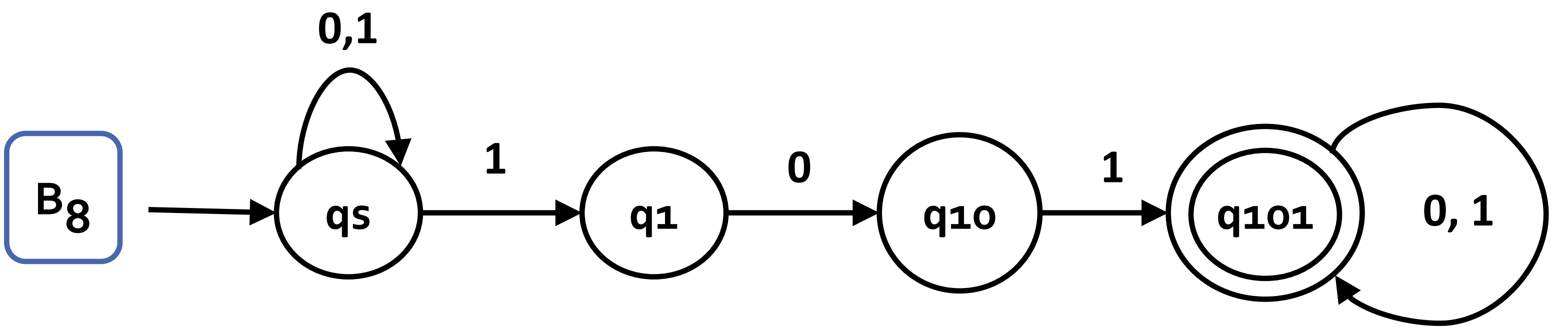
B_7

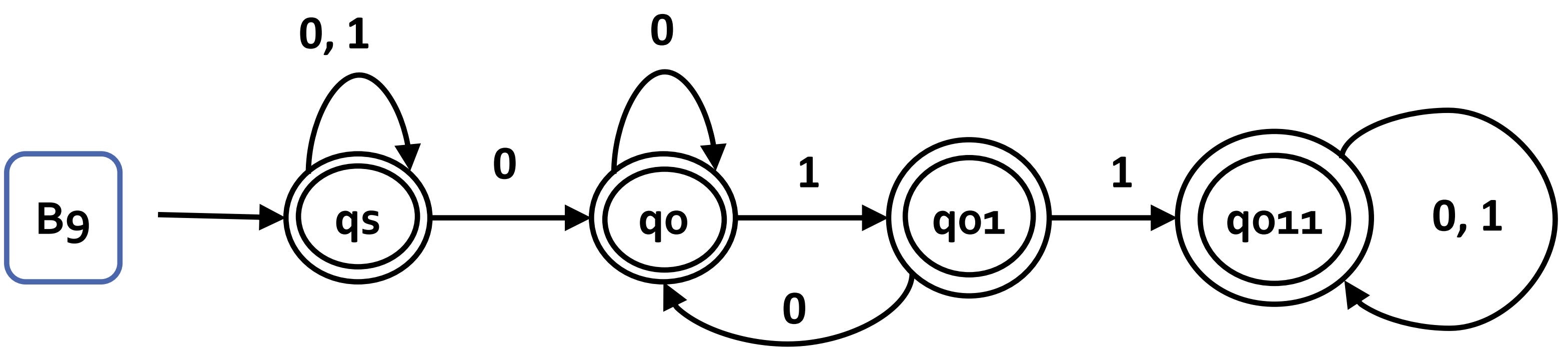


B_8 ={w| w has substring 101}.

B_9 ={w| w has substring 011}.

Σ = {0,1}

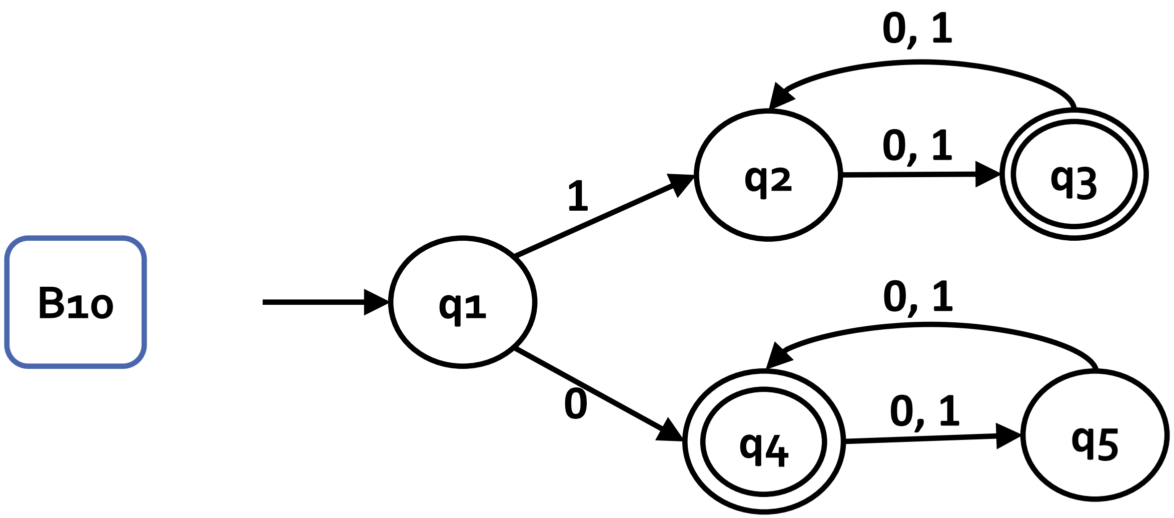




What happens for the language, **A2 ={w| w does not have substring 011}**?

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

∑ = {0,1}



REFERENCES

- Introduction to Theory of Computation, Sipser, (3rd ed), NFA.