***Al-Hamdaniya University***

***College of Education***

***Computer Science***

***Stage: 2nd***

# Finite Automata (FA)

An automaton with a finite number of states is called a ***Finite Automaton (FA)*** or ***Finite State Machine (FSM).***

Finite automata are used to recognize patterns. It takes the string of symbol as input and changes its state accordingly. When the desired symbol is found, then the transition occurs. At the time of transition, the automata can either move to the next state or stay in the same state. Finite automata have two states, Accept state or Reject state. When the input string is processed successfully, and the automata reached its final state, then it will accept.

There are two Type of Finite State Machine (FSM):

1- Deterministic Finite Automaton (DFA)

2- Non- deterministic Finite Automaton (NFA)

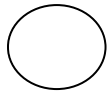
A finite automaton is a collection of three things:

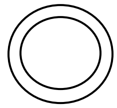
1- A finite set of states, one of which is designated as the initial state, called the ***start state***, and some of which are designated as ***final states***.

2- An alphabet ∑ of possible input letters, from which are formed strings, that are to be read one letter at a time.

3- A finite set of ***transitions*** that tell for each state and for each letter of the input alphabet which state to go to next.

Note:

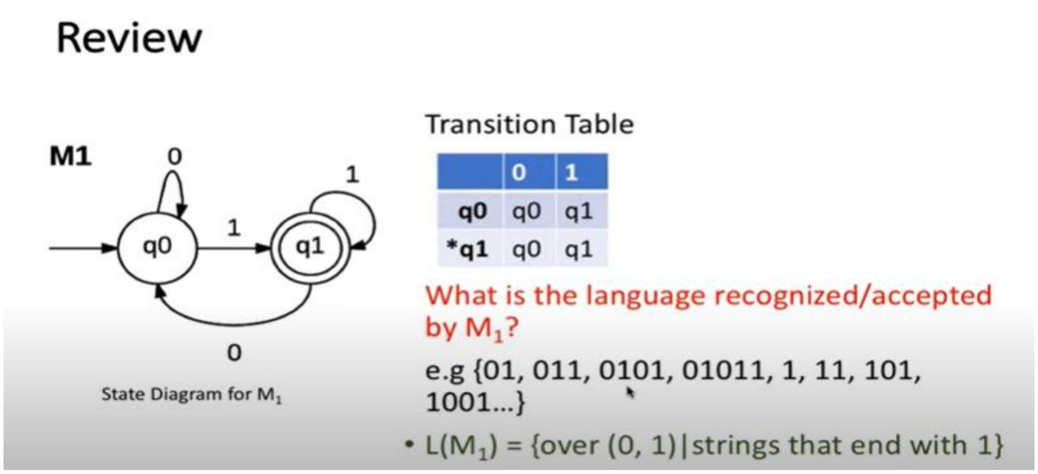
1- Start state denoted 

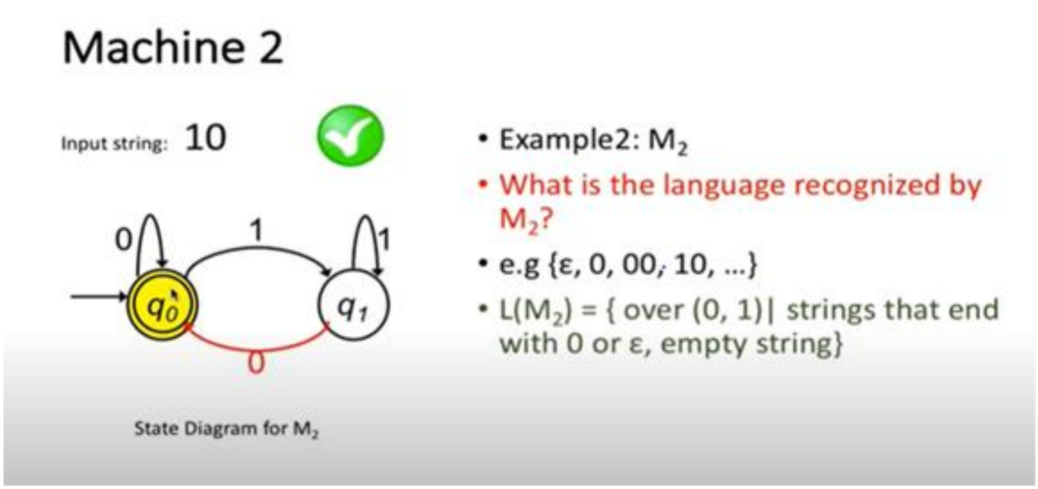
2- Final state denoted 

3-There is one letter input per connection at a time.

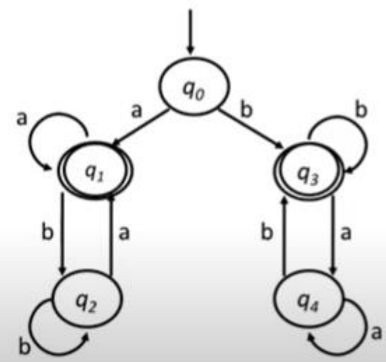
# Deterministic Finite Automaton (DFA)

Ex:



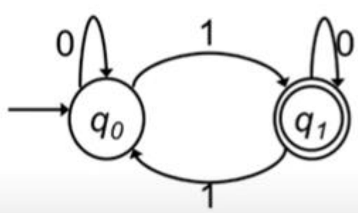
# Machine 3




State Diagram for $M_3$

- Example3: $M_3$
- What is the language recognized by $M_3$?
- Some accepted strings = {a, aa, aba, .., b, bb, bbab, …}
- $M_3$ accepts strings that start and end with a, or starts and end with b.
- $L(M_3)$ = {over (a, b)| strings that start and end with the same symbol}

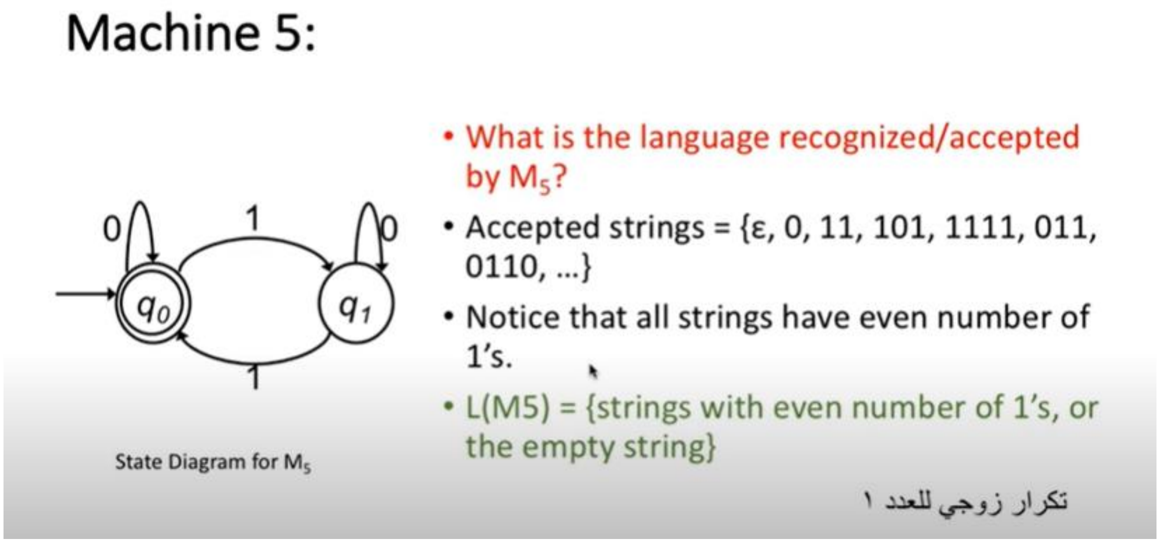
# Machine 4:



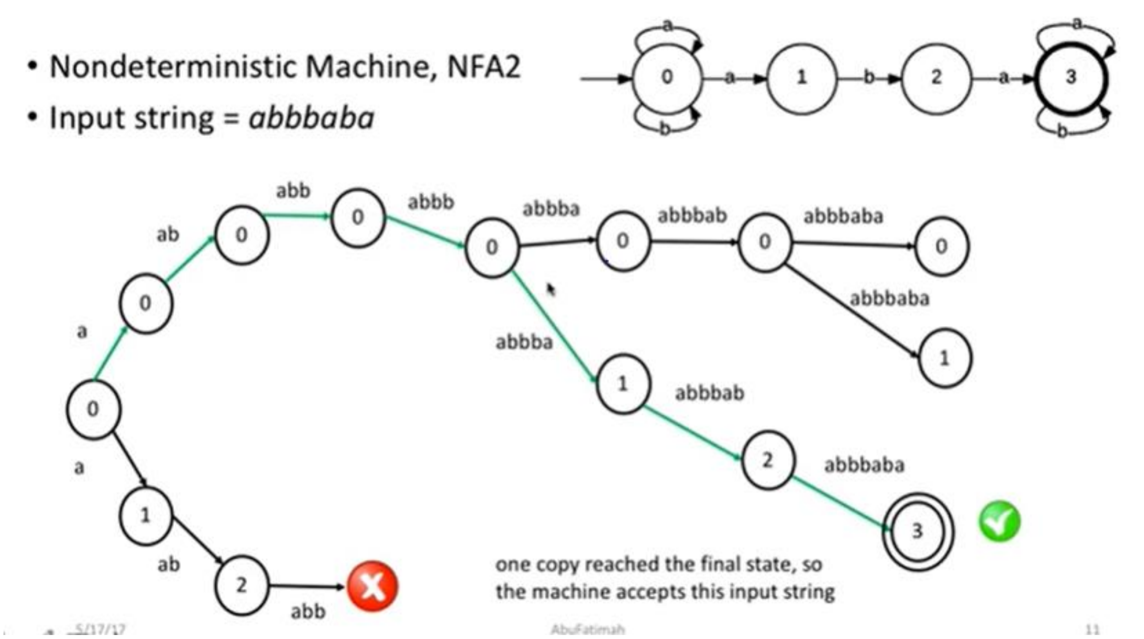

State Diagram for $M_4$

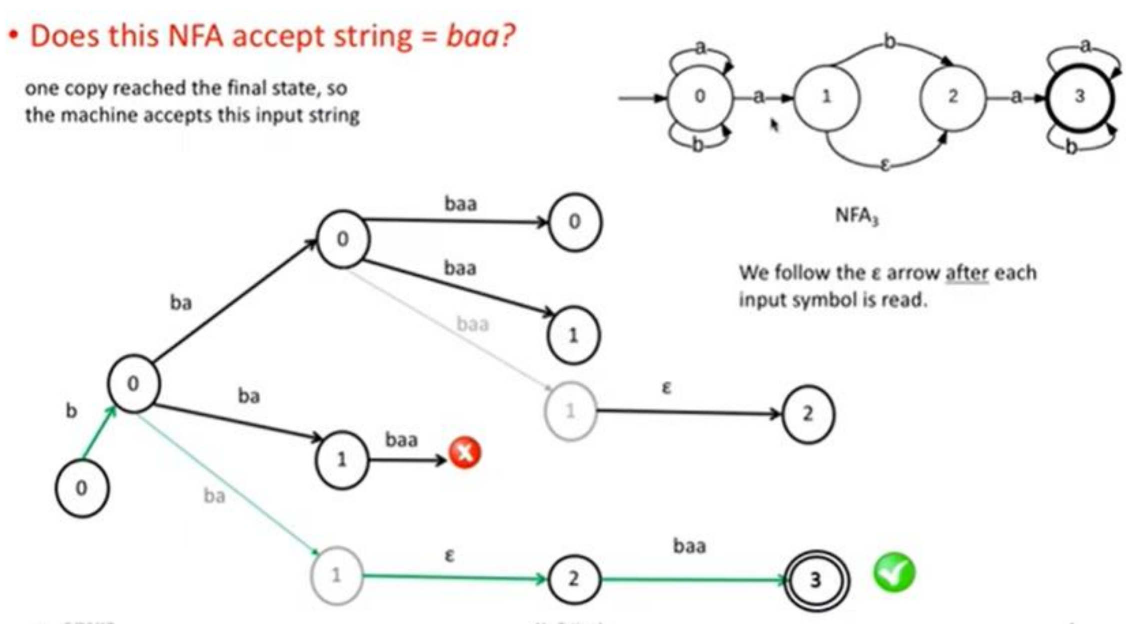
- What is the language recognized/accepted by $M_4$?
- Accepted strings = {1, 10, 111, 1011, 11111, 01, 010, …}
- Notice that all strings have odd number of 1’s.
- $L(M_4)$ = {strings with odd number of 1’s}

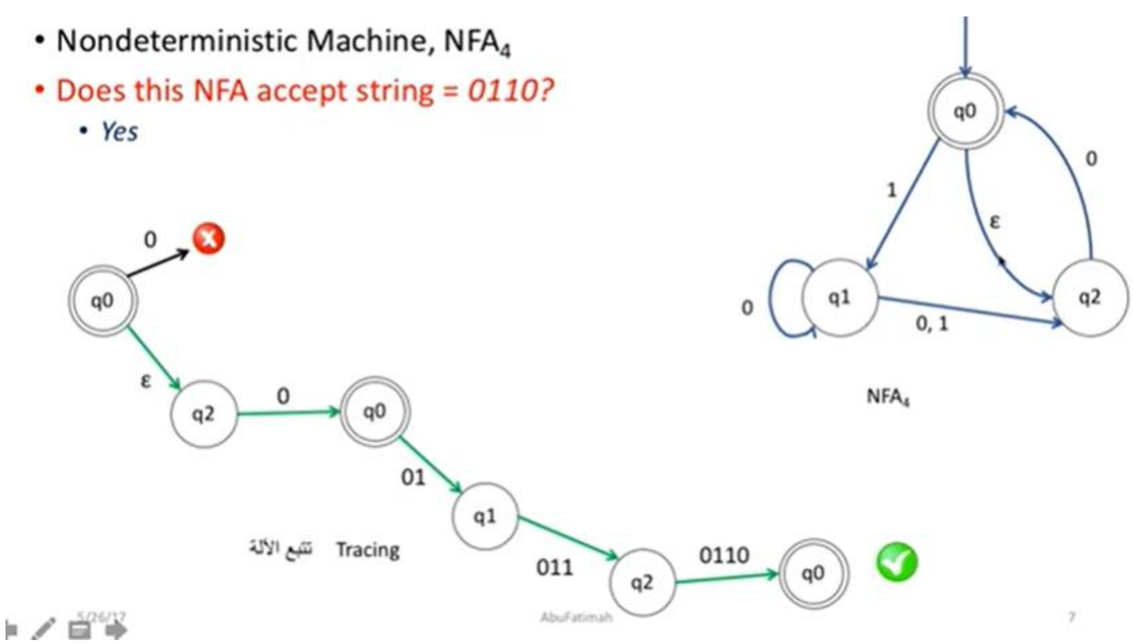
تكرار فردي للعدد ١

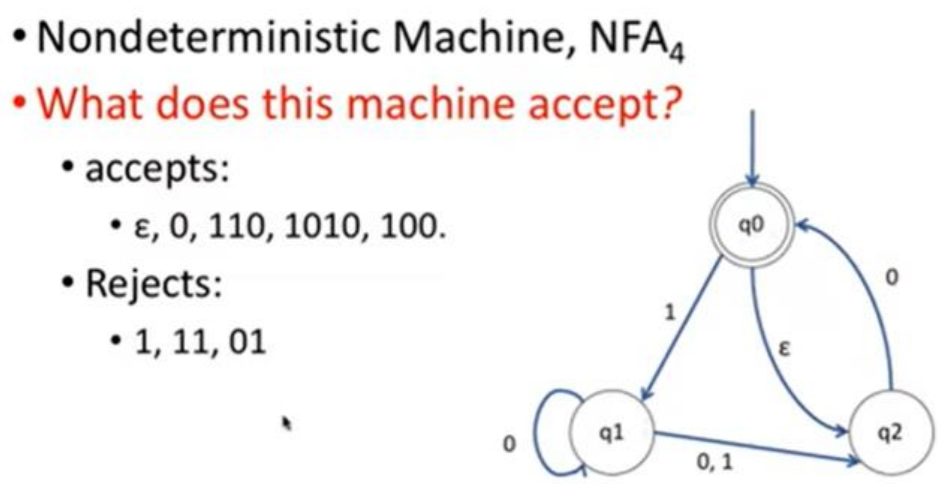



# Non- deterministic Finite Automaton (NFA)




• Does this NFA accept string = baa?
one copy reached the final state, so the machine accepts this input string
NFA3
We follow the ε arrow after each input symbol is read.
ba
baa
ε



• Nondeterministic Machine, NFA4
• Does this NFA accept string = 0110?
• Yes
NFA4
تتبع الآلة Tracing
01
011
0110
AbuFatimah



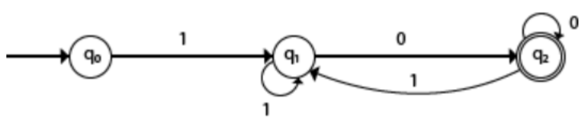

## Designing Deterministic Finite Automata ( DFA)

### Example 1:

Design a FA with ∑ = {0, 1} accepts those string which starts with 1 and ends with 0.

**Solution:**

The FA will have a start state q0 from which only the edge with input 1 will go to the next state.




In state q1, if we read 1, we will be in state q1, but if we read 0 at state q1, we will reach to state q2 which is the final state. In state q2, if we read either 0 or 1, we will go to q2 state or q1 state respectively. Note that if the input ends with 0, it will be in the final state.

### Example 2:

Design a FA with ∑ = {0, 1} accepts the only input 101.

**Solution:**

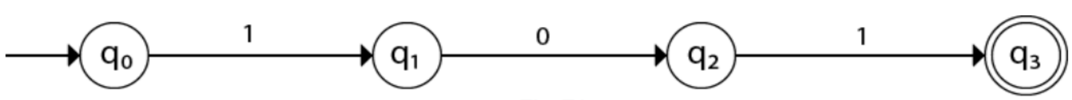



**Fig: FA**

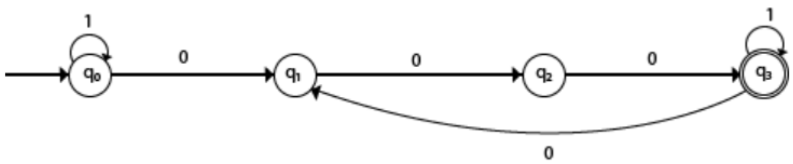
In the given solution, we can see that only input 101 will be accepted. Hence, for input 101, there is no other path shown for other input.

## Example 4:

Design FA with ∑ = {0, 1} accepts the set of all strings with three consecutive 0's.

**Solution:**

The strings that will be generated for this particular languages are 000, 0001, 1000, 10001, .... in which 0 always appears in a clump of 3. The transition graph is as follows:

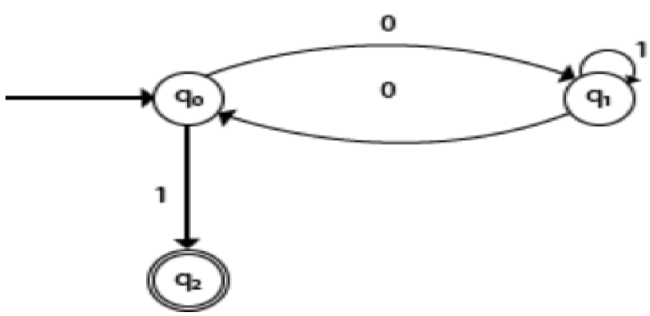



## Example 5

Design a FA with ∑ = {0, 1} accepts the strings with an even number of 0's followed by single 1.

**Solution:**

The DFA can be shown by a transition diagram as:



# Designing Non-Deterministic Finite Automata ( NFA)

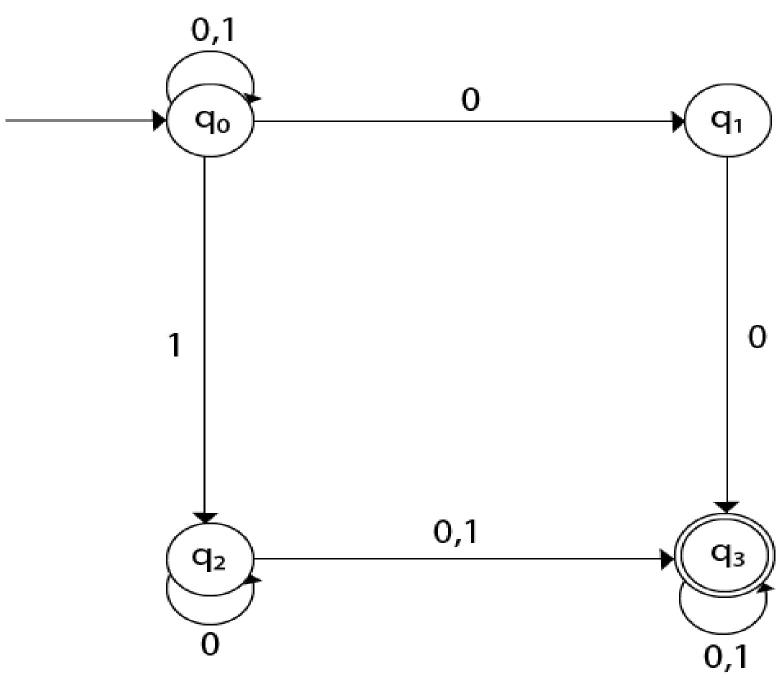
## Example 1:

Design a NFA for the transition table as given below:

| Present State | 0 | 1 |
|---|---|---|
| →q0 | q0, q1 | q0, q2 |
| q1 | q3 | ε |
| q2 | q2, q3 | q3 |
| →q3 | q3 | q3 |

**Solution:**

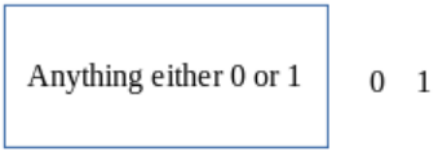
The transition diagram can be drawn by using the mapping function as given in the table.



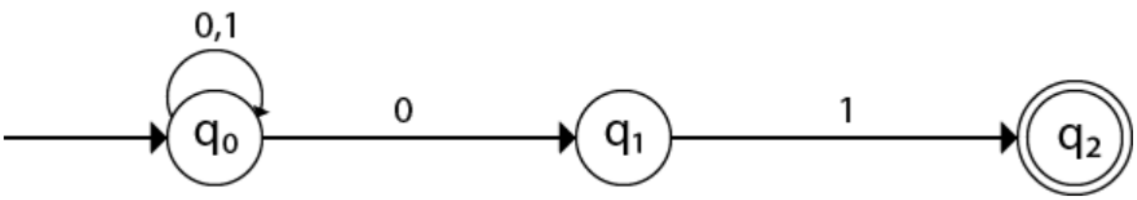
## Example 2:

Design an NFA with ∑ = {0, 1} accepts all string ending with 01.

**Solution:**



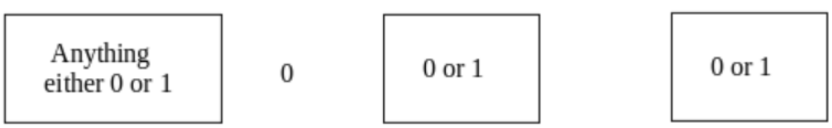

Hence, NFA would be:




## Example 3

Design an NFA with ∑ = {0, 1} accepts all string in which the third symbol from the right end is always 0.

**Solution:**




Thus we get the third symbol from the right end as '0' always. The NFA can be: